

Quantum Hamiltonian simulation of linearised Euler equations in complex geometries

Vladyslav Bohun,¹ Andrij Kuzmak,^{1,2} and Maciej Koch-Janusz^{1,3}

¹Haiqu, Inc., 95 Third Street, San Francisco, CA 94103, USA

²Department for Theoretical Physics, Ivan Franko National University of Lviv, Lviv, Ukraine

³Department of Physics, University of Zürich, 8057 Zürich, Switzerland

Quantum computing promises exponential improvements in solving large systems of partial differential equations (PDE), which forms a bottleneck in high-resolution computational fluid dynamics (CFD) simulations, in, among others, aerospace applications and weather forecasting. One approach is via mapping classical CFD problems to a quantum Hamiltonian evolution, for which recently an explicit quantum circuit construction has been shown in simple cases, allowing proof-of-concept execution on quantum processors. Here we extended this method to more complex and practically relevant cases. We first demonstrate how boundary conditions corresponding to arbitrary complex-shaped obstacles can be introduced in the quantum representations of elementary difference operators used to implement the PDE. We provide explicit and efficient circuit constructions, and show they neither increase the Trotter error, nor asymptotic gate complexity with respect to the free space equation. Using these methods we then derive quantum circuits for simulating the linearized Euler equations in a presence of a background fluid flow and obstacles. We illustrate our results by simulating the obtained quantum circuits for a number of boundary conditions, and compare the errors of the quantum solution to classical finite difference methods.

I. INTRODUCTION

Computational Fluid Dynamics (CFD) methods are key to understanding and modeling natural phenomena of fundamental importance in both science and engineering, from weather and climate forecasting, through biological flows, to aircraft and automobile design. The computational complexity of solving the associated partial differential equation (PDE) systems on required resolutions and sufficiently long time-scales poses, however, a formidable challenge to even state-of-art High-Performance Computing systems.

The progress in quantum computing hardware [1–3] and foundational algorithms [4–10] has inspired a flurry of theoretical research into solving PDEs on a quantum computer [11–22] over the past decade, motivated by the exponential speedups achievable. The proposed quantum approaches span a large space, and differ in their domain applicability (*e.g.* steady state or evolution problems), theoretical guarantees, and suitability for execution on devices prior to achieving full fault-tolerance (*e.g.* HHL [5, 7] *vs.* variational approaches [23–29]). We focus on time-dependent partial differential equations.

One of the most promising approaches to solving such time-dependent problems is to recast them as a quantum time-evolution described by a Schrödinger equation, with an auxiliary Hamiltonian defined by the PDE [30–35]. The evolution is then implemented on a quantum computer as a quantum circuits, using *e.g.* Lie-Trotter-Suzuki method. Whether the problem is easily expressible as Hamiltonian operator is heavily dependent on the PDE; in general case it may require introducing auxiliary qubits and methods such as block-encoding and linear combinations of unitaries [36–42]. While such general constructions may require prohibitively large resources, recently a number of proposals have appeared for implementations in specific cases, in particular for the wave-

equation [34, 43–45], some of which are more suited to near term devices.

Here we tackle a more general problem. As most practically relevant CFD simulations require complicated boundary conditions describing *e.g.* objects such as wings, or constrictions in the flow, we first show how in the mapping to a Hamiltonian evolution arbitrary obstacles can be efficiently introduced in the quantum representations of the PDE. We build explicit quantum circuits allowing their implementation using the Trotter approach, and prove that our constructions neither increases the Trotter error of the simulation nor the asymptotic gate complexity of the simulation circuits. Second, using the above results, we develop explicit quantum circuits for the important problem of modeling sound propagation and scattering of obstacles in the presence of a constant background fluid flow, which is described by the linearized Euler equations (LEE). We verify our constructions by simulating the obtained quantum circuits for different boundary conditions, and comparing the results to those of classical finite difference methods (FDM).

The paper is organized as follows: in Sec. II we review how finite difference operators can be represented as quantum circuits, and how consequently PDEs can be represented as a Schrödinger equation which may be implemented on a quantum computer. In Sec. III we describe how obstacles can be efficiently implemented in the difference operators. In Sec. IV we derived the quantum circuit for modeling sound propagation and scattering described by the linearized Euler equations. We prove statements about the complexity of the resulting algorithm in Sec. V. In Sec. VI we present the results of numerical simulations for a number of obstacles and compare them to classical FDM approach. We conclude and provide outlook in Sec. VII. Appendices provide the details of the circuit constructions and of the mathematical proofs, as well as further numerical examples.

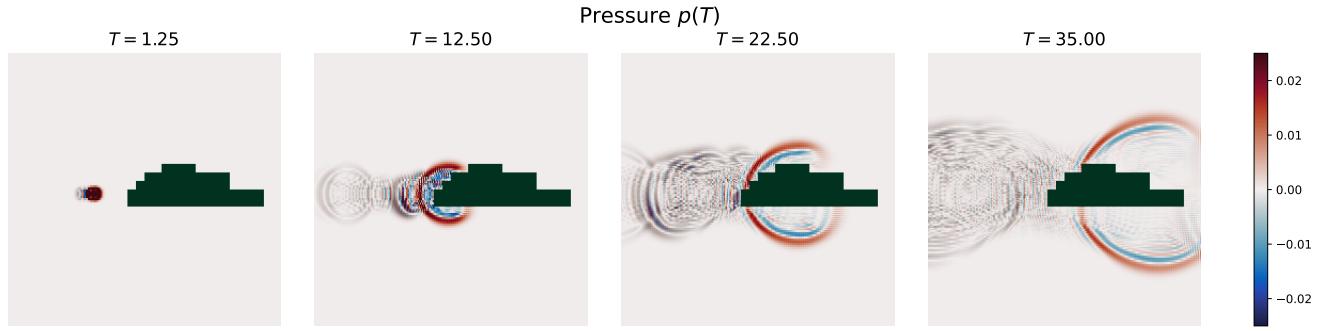


Figure 1: A quantum time-evolution simulation of the 2D linearized Euler equations (LEE) describing acoustic wave propagation in the presence of a constant background fluid flow, and an impenetrable airfoil-like obstacle. Here we plot the deviation of pressure from the equilibrium. The results were obtained by simulating the quantum circuits derived in this work, using $n = 9$ qubits and a Trotter step of $\tau = 0.05$.

II. DIFFERENTIAL EQUATIONS AS TROTTERIZED EVOLUTIONS

We begin by reviewing how some PDEs can be mapped to a quantum Hamiltonian evolution problem on a digital quantum computer using explicit quantum circuit implementation of discrete difference operators. We largely follow the exposition and conventions of Ref. [34].

We consider a system of linear PDEs

$$\frac{df}{dt} = Af, \quad (1)$$

where $f = (f_1, \dots, f_s)$ is a vector function, (x_1, \dots, x_d) are the spatial coordinates, and A is $s \times s$ matrix containing the physical parameters of the system and spatial derivatives $\partial/\partial x_i, \partial^2/\partial x_i \partial x_j, \dots$

When $H := iA$ is a Hermitian operator, that is, $H^\dagger = H$, Eq. 1 can be rewritten as Schrödinger equation

$$\frac{df}{dt} = -iHf, \quad (2)$$

where H plays a role of the Hamiltonian. Solving the equation can then, in theory, be obtained by simply time-evolving an initial condition $f(t=0)$ encoded in the amplitudes of a quantum state with the evolution operator $\exp(-iHt)$.

Performing such a procedure on a digital quantum computer is in practice more subtle. First, to encode the equation in a quantum state the PDE Eq. 1 needs to be discretized. This entails substituting all partial derivatives by suitably chosen difference operators, yielding a difference equation governed by operator D , a discretized version of A . It is important to emphasize that it is D which has to give rise to a hermitian H , and whether this is possible depends on the equation itself (including parameter values), the boundary conditions, and the discretization scheme chosen. Incorporating complex boundary conditions while preserving hermiticity, in particular, is of key practical importance in simulations and is our main focus.

Second, after mapping, the re-cast Hamiltonian evolution needs to be simulated on a digital QPU to a prescribed accuracy, which requires deriving quantum circuits implementing this goal. Whatever the simulation method chosen, the resulting circuit complexity in terms of the elementary gates of the QPU architecture must be shown not to invalidate the putative quantum advantage of the whole approach. We focus on the Trotter evolution and show explicitly that circuits implementing complex conditions, such as obstacles in the fluid flow, can be implemented with only polynomial gate complexity.

A. Difference operators

Let us review the mapping procedure in more detail focusing on, without loss of generality, the example of a scalar function in 1D (*i.e.* $s = d = 1$ in Eq. 1).

We first discretize the 1D support of $f \equiv f_1$ using $N = 2^n$ equidistant points x^k , labeled by integers k , with a lattice constant l . We encode the values of f at these points in the amplitudes of an n -qubit quantum state $|u\rangle$:

$$|u\rangle = \sum_{k=0}^{N-1} u_k |k\rangle, \quad \text{with } u_k := \frac{f(x^k)}{\sqrt{\sum_i |f(x^i)|^2}}, \quad (3)$$

where $|k\rangle \equiv |k^0 k^1 \dots k^n\rangle$ are the computational basis states of the n -qubit system and $k^0 k^1 \dots k^n$ is the binary expansion of k . Note that the amplitudes of $|u\rangle$ are normalized.

The first order derivative $\partial/\partial x$ can be replaced with a discrete forward, backward, or central difference operator D^+ , D^- and D^\pm . The action of these operators on the vector of discretized function values is given by:

$$(D^+ u)_k = \frac{u_{k+1} - u_k}{l}, \quad (D^- u)_k = \frac{u_k - u_{k-1}}{l}, \quad (D^\pm u)_k = \frac{u_{k+1} - u_{k-1}}{2l}. \quad (4)$$

To write down an explicit representation of the difference operators in the n -qubit system in terms of quantum gates we will first need the following shift operators:

$$\begin{aligned} S^- &= \sum_{k=1}^{N-1} |k-1\rangle\langle k| = \sum_{j=1}^n I^{\otimes(n-j)} \otimes \sigma_{01} \otimes \sigma_{10}^{\otimes(j-1)}, \\ S^+ &= \sum_{k=1}^{N-1} |k\rangle\langle k-1| = \sum_{j=1}^n I^{\otimes(n-j)} \otimes \sigma_{10} \otimes \sigma_{01}^{\otimes(j-1)}, \end{aligned} \quad (5)$$

where I is a 1-qubit identity and the 1-qubit operators $\sigma_{\alpha\beta}$ are given by the following 2×2 matrices:

$$\begin{aligned} \sigma_{01} &= |0\rangle\langle 1| = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, & \sigma_{10} &= |1\rangle\langle 0| = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, \\ \sigma_{00} &= |0\rangle\langle 0| = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, & \sigma_{11} &= |1\rangle\langle 1| = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}. \end{aligned} \quad (6)$$

Using the above auxiliary operators difference operators can be written down. Note that their representation depends on the boundary conditions imposed. With Dirichlet conditions at the ends of a simply connected interval they assume the following form:

$$\begin{aligned} D_D^+ &= \frac{1}{l} (S^- - I^{\otimes n}), \\ D_D^- &= \frac{1}{l} (I^{\otimes n} - S^+), \\ D_D^\pm &= \frac{1}{2l} (S^- - S^+). \end{aligned} \quad (7)$$

They are easily seen, from their explicit $N \times N$ matrix representation, to reproduce the action in Eqs. 4:

$$\begin{aligned} D_D^+ &= \frac{1}{l} \begin{pmatrix} -1 & 1 & 0 & \dots \\ 0 & -1 & 1 & \dots \\ 0 & 0 & -1 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}, \\ D_D^- &= \frac{1}{l} \begin{pmatrix} 1 & 0 & 0 & \dots \\ -1 & 1 & 0 & \dots \\ 0 & -1 & 1 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}, \\ D_D^\pm &= \frac{1}{2l} \begin{pmatrix} 0 & 1 & 0 & \dots \\ -1 & 0 & 1 & \dots \\ 0 & -1 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}. \end{aligned} \quad (8)$$

In what follows we shall focus on the Dirichlet conditions (dropping the “D” subscript), and will derive the explicit representation of the difference operators in the more complicated and practically relevant case of impermeable obstacles within the function domain.

The generalization to a d -dimensional domain is simple: each dimension is discretized independently. While different amount of qubits or grid distance can be used for each direction, for simplicity we consider the case when

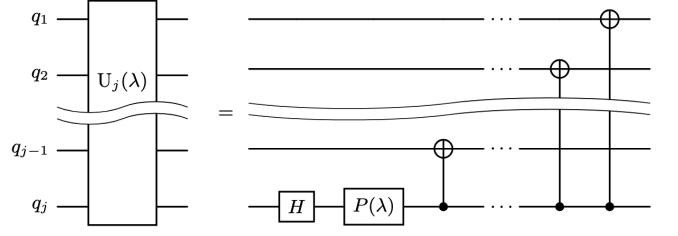


Figure 2: The quantum circuit for the $U_j(\lambda)$ operator (see Eq. 13) acting on the q_1, \dots, q_j register.

n qubits are used per spatial coordinate with the same grid l . Then, the total state on $d \cdot n$ qubits reads

$$|u\rangle = \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{N-1} \dots \sum_{k_d=0}^{N-1} u_{k_1, k_2, \dots, k_d} |k_1, k_2, \dots, k_d\rangle \quad (9)$$

with binary expansions $k_i \equiv k_i^0 k_i^1 \dots k_i^n$. The difference operator acting in the α direction is defined as

$$D_\alpha = I^{\otimes(\alpha-1)n} \otimes D \otimes I^{\otimes(d-\alpha)n}, \quad (10)$$

where D can contain any boundary conditions or differentiation scheme. This form follows from the choice of the encoding $|u\rangle$.

Finally, if the function f is a k -component one with $k > 1$, then the individual components are encoded using at least m ancillary qubits, where $m = \lceil \log_2 k \rceil$. For instance, the two components of a function $f = (f_1, f_2)$ could be encoded in the states $|0\rangle \otimes |u^{(1)}\rangle$ and $|1\rangle \otimes |u^{(2)}\rangle$, and the complete state normalized. In this work we focus on this encoding. Thus a k -component function of d dimensions with 2^n discretisation points in each of them requires a total of $dn + m$ qubits.

B. Induced evolution operators as quantum circuits

To actually implement the PDE on a quantum processor we need an explicit quantum circuit realization of the quantum evolution induced by the difference operator $\exp(-i(iD^\pm)\gamma)$. A naive approach would be to directly express the shifts in terms of which the difference operator can be written (see Eq. 5) through Pauli strings – this would, however, result in an exponentially growing number of terms. Instead, Ref. [34] derived a polynomially sized representation by diagonalizing the Hamiltonian terms in the modified Bell basis composed of states of the form $(|0\rangle|1\rangle^{\otimes j-1} \pm e^{-i\lambda}|1\rangle|0\rangle^{\otimes j-1})/\sqrt{2}$: applying the first order Trotter decomposition in can be shown that the evolution operator can be written as

$$\begin{aligned} e^{D^\pm\gamma} &= e^{-i(iD^\pm)\gamma} \\ &= e^{-i\frac{i}{2l} \sum_{j=1}^n I^{\otimes(n-j)} \otimes (\sigma_{01} \otimes \sigma_{10}^{\otimes(j-1)} - \sigma_{10} \otimes \sigma_{01}^{\otimes(j-1)})} \end{aligned} \quad (11)$$

$$\approx \prod_{j=1}^n U_j(-\pi/2) MCRZ_j^{1, \dots, j-1} \left(\frac{\gamma}{l} \right) U_j(-\pi/2)^\dagger. \quad (12)$$

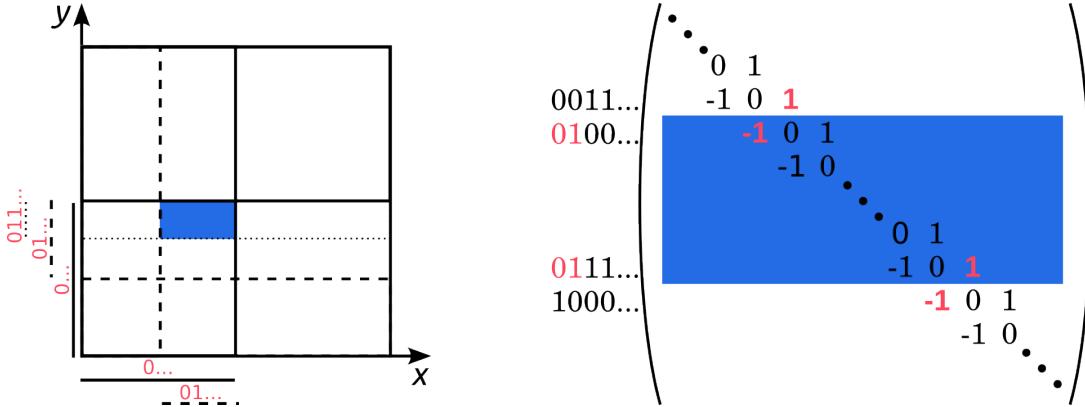


Figure 3: (left) A rectangular object placed inside a binary cell of the discretized domain. The cell is defined by the bit-string prefix 01 in x -direction, and by 011 in y -direction [see the main text]. (right) The matrix representation of the $2D^\pm$ operator (here for the x -direction of the obstacle shown in the left panel), where rows which correspond to the points with the x -coordinate inside the obstacle are shaded in blue.

Here, in the second line, before Trotterization, we used the definitions in Eqs. 5 and 7; in the third line $MCRZ_j^{1,\dots,j-1}(\gamma/l)$ is an $\exp(-iZ_j\gamma/(2l))$ rotation of the j -th qubit (multi)-controlled on the $1,\dots,(j-1)$ -th qubits, with Z the Pauli-Z matrix. The operator $U_j(\lambda)$ mapping to the modified Bell basis is given by (see Fig. 2):

$$U_j(\lambda) = \left(\prod_{m=1}^{j-1} CNOT_m^j \right) P_j(\lambda) H_j, \quad (13)$$

where $CNOT_m^j$ has the j -th qubit as the control and the m -th as the target, and H_j , $P_j(\lambda)$ are the Hadamard and phase shift operators on the j -th qubit:

$$H_j = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad P_j(\lambda) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\lambda} \end{pmatrix}. \quad (14)$$

If necessary, this operator can be implemented in logarithmic depth following the optimized implementations of the GHZ state [46, 47].

A detailed derivation of Eq. 12 and the upcoming CFD operators is presented in Appendices A–C. Note, however, that the individual j factors in the product in Eq. 12 are generated by the corresponding j terms in the sum in Eq. 11, which, respectively, implement 2^{n-j} pairs of ± 1 above/below the diagonal. This intuition will prove useful later.

III. IMPLEMENTING FINITE DIFFERENCE OPERATORS WITH OBSTACLES

Having reviewed the implementation of the basic difference operators we now extend these results and demonstrate how complicated boundary conditions, in particular impermeable obstacles of arbitrary shapes, can be *efficiently* introduced. This is of key practical importance in CFD simulations, when we are interested in, for

instance, sound propagation around an object such as an airfoil or a car chassis moving in an airstream.

For pedagogical and clarity reasons we will now focus on two spatial dimensions (2D) denoted by x and y ; the extension to higher dimensions is trivial. We denote the qubits encoding the x and y coordinates by $q_{x,i}$ and $q_{y,i}$, respectively, where $1 \leq i \leq n$. We will assume Dirichlet boundary conditions at the edges of the domain, in particular we set the value of the field to zero inside the area defining the obstacle. We will also put the derivative of the field to zero inside the obstacle.

As introduced in Sec. II A we use a binary discretization, where each point's coordinates $\alpha = x, y$ are labeled by bitstrings $b_\alpha \equiv b_{\alpha,1} \dots b_{\alpha,n}$ with $b_{\alpha,i} \in \{0,1\}$. We will sometimes find it convenient to think of the bitstring as a binary expansion of an integer $0 \leq b_\alpha \leq 2^n - 1$ which labels the consecutive points on the discretized coordinate axis. In such a grid we define a *binary cell* labeled by a pair of strings $(b_{x,1} \dots b_{x,k_x}, b_{y,1} \dots b_{y,k_y})$ for $1 \leq k_x, k_y \leq n$ to consist of all points (b_x, b_y) whose coordinates have these cell strings as prefixes. The cell prefixes specify both its position and the area, which is easily seen to be $1/2^{k_x} \times 1/2^{k_y}$. An example of a binary cell, defined by the strings (01, 011), is shown in Fig. 3.

Let us begin by first implementing an elementary rectangular obstacle described by a single binary cell. In a classical numerical update scheme implementing a D^\pm difference operator would entail setting all the rows corresponding to the obstacle coordinates in its matrix representation to zero, which from an algorithmic standpoint avoids any updates of the field values at these points. The matrix representation of D^\pm for the x coordinate is shown in Fig. 3b, with the matrix elements to be put to zero in the shaded rectangle.

The quantum evolution implementation, however, is of a different nature, since all of the amplitudes are updated simultaneously. Rather than trying to directly implement an evolution “skipping” all of the selected up-

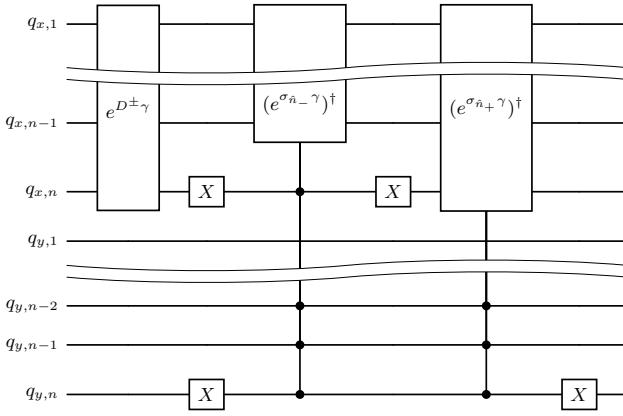


Figure 4: (**Left**) The quantum circuit implementing the difference operator (derivative) on the x axis with an impenetrable object inside the 2D environment (with Dirichlet boundary conditions on its edges) defined by the bitstrings $b_{x,1}b_{x,2} = 01$ on the x coordinate and by $b_{y,1}b_{y,2}b_{y,3} = 011$ on the y coordinate. Here the D^\pm blocks implement the obstacle-free evolution and the prefix-controlled operators (see the main text) remove the matrix elements corresponding to the obstacles. (**Right**) The corresponding circuit for the difference operator on the y axis. Note that the “little-endian” convention is used, that is the left-most bits in the bit-strings correspond to the qubits with the largest index.

dates, which would result in a highly complex circuit, our approach will be different. First, we implement the evolution under the full D^\pm operator (see Eq. 12), and subsequently we cancel back some of the updates with an evolution under an auxiliary operator. Naively one may think that for this purpose a quantum circuit needs to be constructed, which removes all of the elements put to zero in the classical scheme, but in practice it suffices to construct an evolution which only sets to zero four entries in D^\pm (marked red in Fig. 3b), which greatly simplifies the circuit construction. Indeed, due to the choice of boundary conditions and field values inside the obstacle, putting the two marked matrix entries outside the shaded area to zero does not affect the numerical scheme, since the field values of interest are zero.

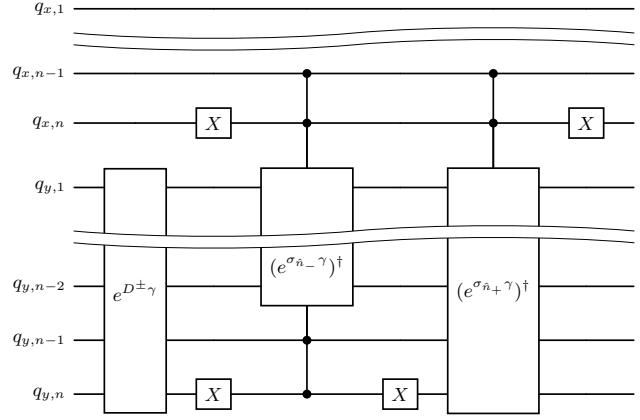
We will perform this construction explicitly, without loss of generality for the x coordinate. Let the position of the obstacle on the x axis be specified by the prefix $b_{x,1} \dots b_{x,k_x}$. It is easily seen that the matrix elements of the D^\pm operator (for the x coordinate) corresponding to the two entries marked in red in the top left of Fig. 3b are given by:

$$\begin{aligned} & |(b_{x,1} \dots b_{x,k_x} - 1)111 \dots \rangle \langle b_{x,1} \dots b_{x,k_x} 000 \dots| \\ & - |b_{x,1} \dots b_{x,k_x} 000 \dots \rangle \langle (b_{x,1} \dots b_{x,k_x} - 1)111 \dots|, \end{aligned} \quad (15)$$

while the bottom-right pair matrix elements are given by:

$$\begin{aligned} & |b_{x,1} \dots b_{x,k_x} 111 \dots \rangle \langle (b_{x,1} \dots b_{x,k_x} + 1)000 \dots| \\ & - |(b_{x,1} \dots b_{x,k_x} + 1)000 \dots \rangle \langle b_{x,1} \dots b_{x,k_x} 111 \dots|, \end{aligned} \quad (16)$$

where the ± 1 additions in the prefixes denote bit strings obtained from increasing by one the integer, whose binary representation the original prefix was. In other words:



we change the row/column index in the matrix by one, and take its binary representation, as dictated by the tridiagonal structure of the matrix D^\pm .

Let p_x^- be the size of largest common prefix of $b_{x,1} \dots b_{x,k_x}$ and $b_{x,1} \dots b_{x,k_x} - 1$, and p_x^+ the largest common prefix of $b_{x,1} \dots b_{x,k_x}$ and $b_{x,1} \dots b_{x,k_x} + 1$ with $0 \leq p_x^-, p_x^+ < k_x$. Then it is easy to verify that

$$\begin{aligned} & |(b_{x,1} \dots b_{x,k_x} - 1)111 \dots \rangle \langle b_{x,1} \dots b_{x,k_x} 000 \dots| \\ & - |b_{x,1} \dots b_{x,k_x} 000 \dots \rangle \langle (b_{x,1} \dots b_{x,k_x} - 1)111 \dots| = \\ & = |b_{x,1} \dots b_{x,p_x^-} \rangle \langle b_{x,1} \dots b_{x,p_x^-}| \otimes \\ & \quad (|(0111 \dots \rangle \langle 1000 \dots| - |1000 \dots \rangle \langle 0111 \dots|) = \\ & \equiv |b_{x,1} \dots b_{x,p_x^-} \rangle \langle b_{x,1} \dots b_{x,p_x^-}| \otimes 2l \cdot \sigma_{\hat{n}_-}, \end{aligned} \quad (17)$$

where the operator $\sigma_{\hat{n}}$ appearing in the last line with $\hat{n} := \hat{n}_-$ (here: $\hat{n}_- = n - p_x^-$) comprises the suffix part acting on \hat{n} qubits and is defined as:

$$\sigma_{\hat{n}} \equiv \frac{1}{2l} (|0111 \dots \rangle \langle 1000 \dots| - |1000 \dots \rangle \langle 0111 \dots|). \quad (18)$$

Similarly it is easily seen that the operator in Eq. 16 can now be expressed as (with $\hat{n}_+ = n - p_x^+$):

$$|b_{x,1} \dots b_{x,p_x^+} \rangle \langle b_{x,1} \dots b_{x,p_x^+}| \otimes 2l \cdot \sigma_{\hat{n}_+}. \quad (19)$$

Eqs. 17 and 19 are valid for all prefixes except the edge cases at the boundary of the domain where they are all-0 or all-1. However, in this case the corresponding matrix elements are already absent in the full operator and there is no need for the correction.

With the above observations and definitions the difference operator without the matrix elements in Eqs. 15, 16

(see Fig. 3) can be written as:

$$D^\pm = |b_{x,1} \dots b_{x,p_x^-}\rangle\langle b_{x,1} \dots b_{x,p_x^-}| \otimes \sigma_{\hat{n}_-} \quad (20)$$

$$- |b_{x,1} \dots b_{x,p_x^+}\rangle\langle b_{x,1} \dots b_{x,p_x^+}| \otimes \sigma_{\hat{n}_+}.$$

To implement the evolution generated by the operator Eq. 20 we use the first order Trotter decomposition. First, we implement the evolution $\exp(D^\pm \gamma)$ according to the original difference operator (see Eq. 12). For the remaining “obstacle” part, observe that the explicit matrix form of the operator $\sigma_{\hat{n}}$ in Eq. 18 consists of a single ± 1 pair above/below diagonal, *i.e.* has the same form as the $j = \hat{n}$ term in Eq. 11. The evolution it generates can thus be written as:

$$e^{\sigma_{\hat{n}} \gamma} = e^{-i(i\sigma_{\hat{n}})\gamma} = e^{-i\frac{j}{2l}(\sigma_{01} \otimes \sigma_{10}^{\otimes(\hat{n}-1)} - \sigma_{10} \otimes \sigma_{01}^{\otimes(\hat{n}-1)})}$$

$$= U_{\hat{n}}(-\pi/2) MCRZ_{\hat{n}}^{1, \dots, \hat{n}-1} \left(\frac{\gamma}{l} \right) U_{\hat{n}}(-\pi/2)^\dagger. \quad (21)$$

Furthermore, the complete obstacle terms in Eq. 20 have the form of a tensor product of the $\sigma_{\hat{n}}$ operator and a projector on a specific prefix bit string, and thus the evolution according to them is obtained by taking a multi-controlled version of Eq. 21, with the control selecting precisely the qubit states corresponding to the bit string required. Note, that here for clarity we have only considered the x -coordinate, but the obstacle operators have to also include the controls on the bit strings defining the extent of the obstacle in the other spatial directions; this is shown schematically in Fig. 4. We also remark that in this work we use the “little-endian” convention between bits and qubits: the left-most bits of the bit-string correspond to qubits with the largest indices.

As an illustration of the above construction in a 2D example, consider again the obstacle depicted in Fig. 3, defined by the string $b_{x,1}b_{x,2} = 01$ on the x coordinate and by $b_{y,1}b_{y,2}b_{y,3} = 011$ on the y coordinate. To construct the x derivative with obstacles we first compute the common prefixes which have lengths $p_x^- = 1$ and $p_x^+ = 0$. The resulting circuit is shown in Fig. 4: first, the full D^\pm block, and then the two obstacle terms, controlled on the common prefixes in the q_x register, and also on the full 011 bit string in the q_y register. The difference operator for the y derivative is obtained analogously (*i.e.* controlled on the prefixes in the y direction and the full 01 bitstring in the x direction).

Though the above procedure provides a blueprint for implementing in the flow an obstacle of arbitrary shape (composed of multiple binary cells of different sizes) its practical implementation can be further improved to reduce both the theoretical and practical errors. For the former we note that a direct Trotter approximation of the evolution generated by the operator in Eq. 20 would violate the boundary conditions on the obstacles, due to the non-commutation of the complete obstacle and the D^\pm terms. This, however, can be prevented entirely. As shown in Appendix E the generator in Eq. 20 can be rearranged so that the obstacle terms involving the $\sigma_{\hat{n}}$ generator are grouped with the corresponding $j = \hat{n}$ terms in

D^\pm , with which they commute, even in their controlled version. This allows to implement the obstacle *exactly*, without incurring any additional Trotter error with respect to the obstacle free evolution and thus ensures that the obstacles are indeed perfectly impermeable.

Furthermore, the above re-arrangement can be combined with additional circuit-level simplifications. Particularly the construction of multi-controlled and multi-target operators seen in Fig. 4 can be made much more efficient, reducing the required circuit depth. While the gains obtained in this fashion amount to constant factors, these can have a very strong effect on the practical performance, especially on early QPUs currently (and in the near- to mid-term future) available. For details we refer to the Appendix E.

IV. LINEARIZED EULER EQUATIONS WITH A MEAN FLOW

Here we demonstrate the application of our methods to the quantum simulation of linearized Euler equations (LEE) in the presence of a background fluid flow, and in the presence of obstacles, for which we develop explicit circuit representations.

LEE, obtained by linearizing the complete Euler equations around a mean flow, allow the simulation of *i.a.* acoustic wave propagation in aerospace and automotive design simulations, and are thus of key practical importance. From the point of view of quantum simulation a key challenge is that the presence of a mean flow generically hinders the mapping of LEE to a Hermitian operator described Sec. II. Here we show that for specific parameter values corresponding to the conservative regime this is still possible. Furthermore, the quantum circuits we derive in this Hermitian case are also the key building block for the general problem of LEE simulation with arbitrary parameter values. For more details we refer the reader to App. G.

For concreteness let us now consider LEE with a background flow in the x direction:

$$\begin{cases} \frac{\partial p}{\partial t} = -\bar{\rho}c^2 \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) - \bar{u} \frac{\partial p}{\partial x}, \\ \frac{\partial u}{\partial t} = -\frac{1}{\bar{\rho}} \frac{\partial p}{\partial x} - \bar{u} \frac{\partial u}{\partial x}, \\ \frac{\partial v}{\partial t} = -\frac{1}{\bar{\rho}} \frac{\partial p}{\partial y} - \bar{u} \frac{\partial v}{\partial x}, \end{cases} \quad (22)$$

where p is the deviation from the mean pressure (*i.e.* can be negative), u and v are the velocity components in the x and y directions, respectively, $\bar{\rho}$ is a static density of the environment, \bar{u} is a speed of flow in x direction and c is the speed of sound in the environment.

We represent the solution at each point as a four-component vector $f = (p, u, v, 0)$, which requires $m = 2$ ancilla qubits to encode (see discussion in Sec. II A); we denote them by a_1 and a_2 . The zero component in the solution vector appears as we require it to be a power of 2 in the qubit encoding. With this LEE can be represented

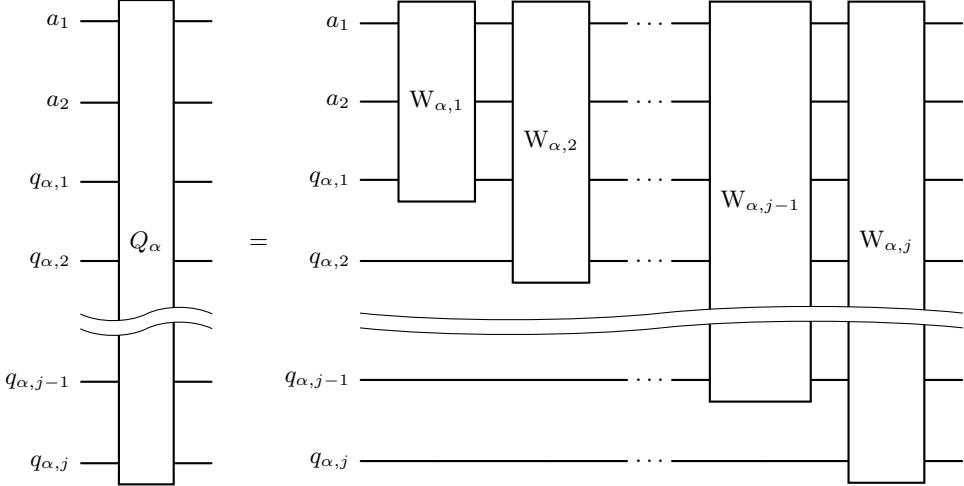


Figure 5: The quantum circuit implementing the Q_α time evolution Eq. 27 for the $\alpha = x, y$ spatial direction.

in the form of Eq. 1 with the matrix A given by:

$$A = \begin{pmatrix} -\bar{u} \frac{\partial}{\partial x} & -\bar{\rho} c^2 \frac{\partial}{\partial x} & -\bar{\rho} c^2 \frac{\partial}{\partial y} & 0 \\ -\frac{1}{\bar{\rho}} \frac{\partial}{\partial x} & -\bar{u} \frac{\partial}{\partial x} & 0 & 0 \\ -\frac{1}{\bar{\rho}} \frac{\partial}{\partial y} & 0 & -\bar{u} \frac{\partial}{\partial x} & 0 \\ 0 & 0 & 0 & -\bar{u} \frac{\partial}{\partial x} \end{pmatrix}. \quad (23)$$

Note the appearance of the derivative on the diagonal in the auxiliary dimension introduced. While it has no effect on the dynamics (the corresponding entry in the solution is always zero), it simplifies the circuit construction.

It is now easy to see that in the special case when the speed of sound is $c = 1/\bar{\rho}$ the matrix in Eq. 23 is symmetric. This case corresponds to the regime, where the total energy (kinetic and pressure) is conserved exactly. When applying the D^\pm difference operator prescription for discretization, the whole matrix A becomes *anti-symmetric*, and thus the problem can be directly rewritten as a Schrödinger equation, as in Eq. 2, with the Hamiltonian given by:

$$H = -i \left[\bar{u} I^{a_1} \otimes I^{a_2} \otimes D_x^\pm + \frac{1}{\bar{\rho}} \sigma_{00}^{a_1} \otimes X^{a_2} \otimes D_x^\pm + \frac{1}{\bar{\rho}} X^{a_1} \otimes \sigma_{00}^{a_2} \otimes D_y^\pm \right]. \quad (24)$$

Here and later, if necessary, we indicate the qubits an operator acts on as an upper index; σ_{00} is defined in Eq. 6. We further group the Hamiltonian terms related to

the x and y derivatives, respectively, and write them out explicitly using the shift operators introduced in Eq. 5:

$$H = H_x + H_y = \sum_{j=1}^n H_{x,j} + \sum_{j=1}^n H_{y,j}, \quad (25)$$

where

$$\begin{aligned} H_{x,j} &= -\frac{i}{2l} \left(\bar{u} I^{a_1} \otimes I^{a_2} + \frac{1}{\bar{\rho}} \sigma_{00}^{a_1} \otimes X^{a_2} \right) \\ &\otimes I^{\otimes(n-j)} \otimes (\sigma_{01} \otimes \sigma_{10}^{\otimes(j-1)} - \sigma_{10} \otimes \sigma_{01}^{\otimes(j-1)}) \otimes I^{\otimes n}, \\ H_{y,j} &= -\frac{i}{2l\bar{\rho}} X^{a_1} \otimes \sigma_{00}^{a_2} \\ &\otimes I^{\otimes n} \otimes I^{\otimes(n-j)} \otimes (\sigma_{01} \otimes \sigma_{10}^{\otimes(j-1)} - \sigma_{10} \otimes \sigma_{01}^{\otimes(j-1)}). \end{aligned}$$

The time-evolution $V(\tau)$ over a small interval τ can thus be written using first-order Trotterization as:

$$V(\tau) = e^{-iH\tau} \approx e^{-iH_y\tau} e^{-iH_x\tau} \equiv Q_y(\tau) Q_x(\tau). \quad (26)$$

The non-commuting “per-direction” evolution operators Q_x and Q_y can be further written as products:

$$Q_x = \prod_{j=1}^n W_{x,j}, \quad Q_y = \prod_{j=1}^n W_{y,j}. \quad (27)$$

The operators $W_{x,j}$ and $W_{y,j}$ act on the two ancillas and j spatial discretization qubits (see Fig. 5). Their circuit representations are shown in Figs. 6, 7 and the corresponding explicit expressions are given below:

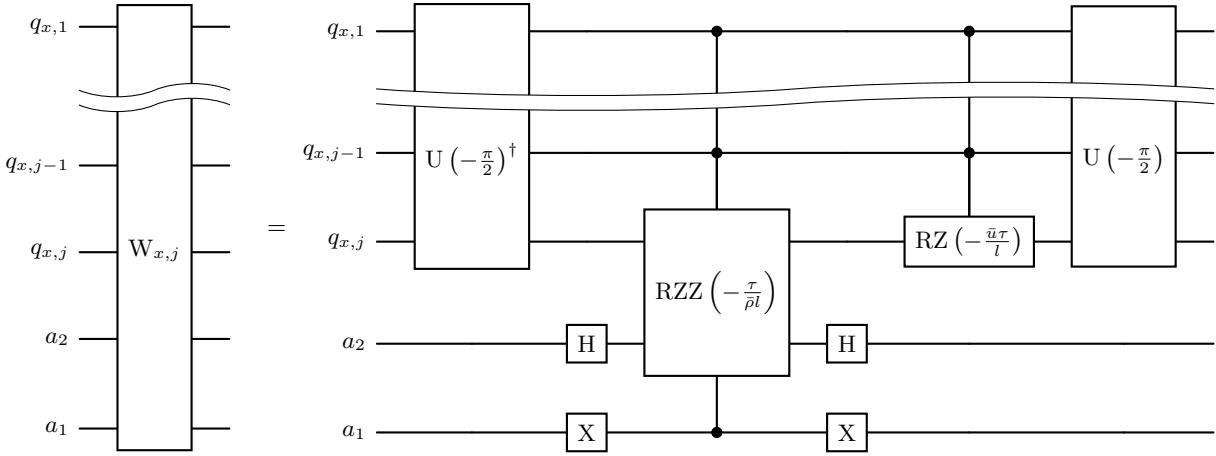


Figure 6: The quantum circuit for the $W_{x,j}$ operator acting on the $q_{x,1}, \dots, q_{x,j}$ spatial and a_1, a_2 ancilla registers.

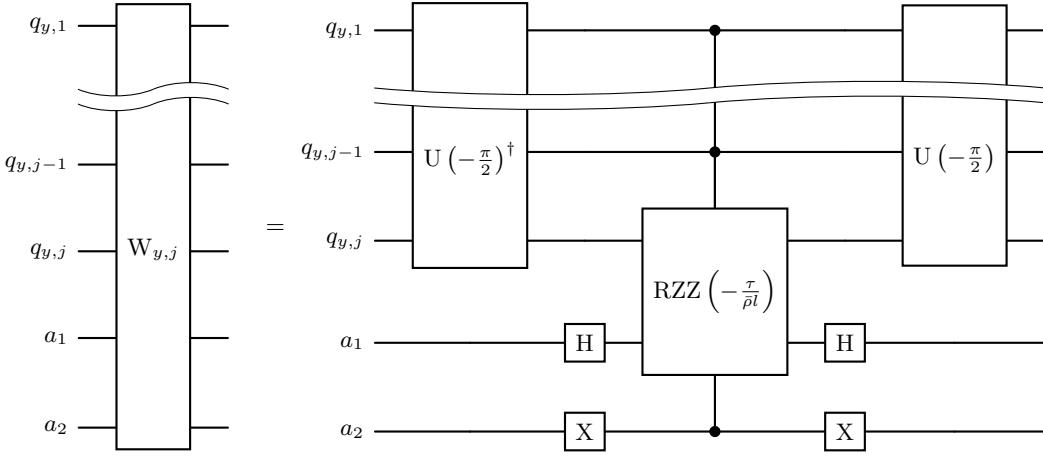


Figure 7: The quantum circuit for the $W_{y,j}$ operator acting on the $q_{y,1}, \dots, q_{y,j}$ spatial and a_1, a_2 ancilla registers.

$$\begin{aligned}
 W_{x,j} &= I^{a_1} \otimes I^{a_2} \otimes I^{\otimes(n-j)} \otimes \left(U_j(-\pi/2) MCRZ_j^{1,\dots,j-1}(-\bar{u}\tau/l) U_j(-\pi/2)^\dagger \right) \otimes I^{\otimes n} \\
 &\quad \times I^{\otimes(n-j)} \otimes (H^{a_1} \otimes X^{a_2} \otimes U_j(-\pi/2)) MCRZZ_{a_2,j}^{a_2,1,\dots,j-1}(-\tau/(\bar{\rho}l)) (U_j(-\pi/2)^\dagger \otimes X^{a_1} \otimes H^{a_2}) \otimes I^{\otimes n}, \\
 W_{y,j} &= I^{\otimes n} \otimes I^{\otimes(n-j)} \otimes (H^{a_1} \otimes X^{a_2} \otimes U_j(-\pi/2)) MCRZZ_{a_1,j}^{a_2,1,\dots,j-1}(-\tau/(\bar{\rho}l)) (U_j(-\pi/2)^\dagger \otimes H^{a_1} \otimes X^{a_2}), \quad (28)
 \end{aligned}$$

where the multi-controlled Z -rotation operator $MCRZ$ was introduced in Sec. II B and where, analogously, $MCRZZ_{a_1(a_2),j}^{a_2(a_1),1,\dots,j-1}(-\tau/(\bar{\rho}l))$ provides an $\exp(iZ_{a_1(a_2)}Z_j\tau/(2\bar{\rho}l))$ Ising rotation of qubits a_1 (respectively: a_2) and j , controlled on the qubits a_2 (respectively: a_1) and $1, \dots, j-1$. A detailed derivation of the operator $V(\tau)$ is presented in Appendix A.

The total evolution over the time period $T = s\tau$ is obtained in the first-order Trotter scheme as $f(T) \approx [V(\tau)]^s f(0)$, where the quantum state encoding the initial condition should be properly normalized $\|f(0)\| = 1$.

The operators Q_x and Q_y can also be used to implement higher-order Trotter-Suzuki evolutions [48]. Note also that the complexity of quantum circuit preparing the initial state is highly dependent on the state itself, and may in general be considerable.

This completes the derivation of the evolution circuit for the LEE Eq. 22 with a constant background flow in the conservative regime. It can now be combined with the obstacle circuits derived in Sec. III and App. E to allow the quantum simulation of acoustic phenomena in non-trivial settings. The only difference is that the ob-

stacle terms are now implemented for the D^\pm difference operators in the Hamiltonian Eq. 24. We emphasize that while the conservative regime is a special case, the obtained circuits are the building blocks also in the general cases where the evolution cannot be directly mapped to a Hamiltonian evolution (see Appendix G).

Finally, we remark that the circuits depicted in Figs. 6, 7 contain multi-controlled and multi-target rotation MCRZZ operators. In Appendix B we show that using the D^\pm and the D^+, D^- difference operators simultaneously the evolution operator construction can be simplified, in particular avoiding multi-target gates. We further derive the circuit for the periodic boundary conditions case in Appendix C.

V. ALGORITHM COMPLEXITY

We briefly describe the resources required to implement the methods we discussed. We use n qubits to represent the discretized grid of 2^n points in each direction (we focus on two spatial dimensions here). The most important practical metric in analysing the circuit complexity is the time increment $\tau = T/s$ (s is the number of Trotter steps), the number of two-qubit (typically CNOT) gates; the number of one-qubit gates is of a similar order.

Lemma 1. *The evolution operator $\exp(-iH\tau)$ with the Hamiltonian H given in Eq. 24 and a time increment τ can be approximated in the first-order Lie-Trotter-Suzuki decomposition by the unitary operator $V(\tau)$ Eq. 26 with an approximation error (in the sense of operator spectral norm) upper bounded bounded by:*

$$\begin{aligned} \|\exp(-iH\tau) - V(\tau)\| &\leq \left[\left(\frac{\bar{u}}{2} \right)^2 + 2 \left(\frac{1}{2\bar{\rho}} \right)^2 + \frac{\bar{u}}{2\bar{\rho}} \right] \frac{\tau^2(n-1)}{2l^2} \\ &\quad + \left(\frac{1}{2\bar{\rho}} \right)^2 \frac{\tau^2 n^2}{2l^2}. \end{aligned} \quad (29)$$

For the proof of Lemma 1 we refer to the Appendix D. We also show that the gate complexity of the unitary operator $V(\tau)$ is quadratic in the number of qubits, as formalized by the following lemma:

Lemma 2. *The evolution operator $V(\tau)$ Eq. 26 depicted in Figs. 5, 6, 7 can be implemented with $\mathcal{O}(n^2)$ CNOT gates in all-to-all qubit connectivity.*

Proof. The operator $V(\tau)$ consists of multiple $U_j(\lambda)$ operators and multi-controlled rotations. We show that the implementation of both groups does not exceed quadratic complexity. First, note that by Eq. 13 a single $U_j(\lambda)$ contains $j-1$ CNOTs, and $V(\tau)$ contains a linear number of operators W , and thus of $U_j(\lambda)$, as per Figs. 5, 6, 7. In total $2n(n-1)$ CNOT gates are enough to implement all $U_j(\lambda)$ and $U_j(\lambda)^\dagger$ operators.

Since each $W_{x,j}$ contains a MCRZZ and a MCRZ gate (see Fig. 6) and $W_{y,j}$ an MCRZZ gate (Fig. 7) the evolution $V(\tau)$ contains n MCRZ gates and $2n$ MCRZZ gates in total. The latter can be decomposed into two MCRZ gates with an additional control (see Fig. 13). While decompositions of general multi-controlled gates may require quadratic number of CNOTs [49–51], in special cases, particularly with RZ as the controlled unitary, efficient decompositions linear in the number of CNOT gates are known [52, 53]. This provides an upper boundary $\mathcal{O}(n^2)$ for the complexity of the Trotter step, completing the proof. \square

Similarly, it can be easily shown that the total amount of single-qubit gates is $\mathcal{O}(n^2)$ per Trotter step. The exact number is dependent on the decomposition used and device topology and gate set. We remark that in sparse topologies the SWAP gates can increase the gate complexity by a factor of n . For the all-to-all topology, using the decompositions in Ref. [52], at most $42n^2 - 34n + 34$ CNOT gates are needed to implement $V(\tau)$ for $n \geq 3$.

The total simulation complexity over time interval $T = \tau s$ can now estimated:

Lemma 3. *The CNOT cost of simulating the evolution operator $\exp(-iHT)$ with the 2D Hamiltonian Eq. 24 with $2n+2$ qubits on a grid with size l to an additive error $\varepsilon > 0$ is:*

$$\mathcal{O}\left(\frac{T^2 n^4}{l^2 \varepsilon}\right) \quad (30)$$

Proof. Using the estimate from Lemma 1 we have that to achieve the additive error ε over the time interval $T = \tau s$ we need s to be of order:

$$s = \mathcal{O}\left(\frac{T^2 n^2}{l^2 \varepsilon}\right). \quad (31)$$

At each step we implement an operator $V(\tau)$ with a quadratic CNOT complexity, as given by Lemma 2. Combining the two results we obtain Eq. 30. \square

We have thus obtained that the total simulation cost is polynomial in the number of qubits, *i.e.* it offers an exponential advantage over classical schemes explicitly updating a $2^n \times 2^n$ spatial grid, at least when $l = 1/\text{poly}(n)$. This holds for an obstacle implemented to a finite resolution, *i.e.* with a constant amount of binary cells (see App. E).

We also draw attention of the reader to the interesting fact that, contrary to classical schemes, the quantum solution is always stable, it only contains the Trotter error. This is due to the fact that it directly realizes the matrix exponential form of the solution, which would be the case for a classical integration scheme only in the limit $\tau \rightarrow 0$ (see also Sec. VI).

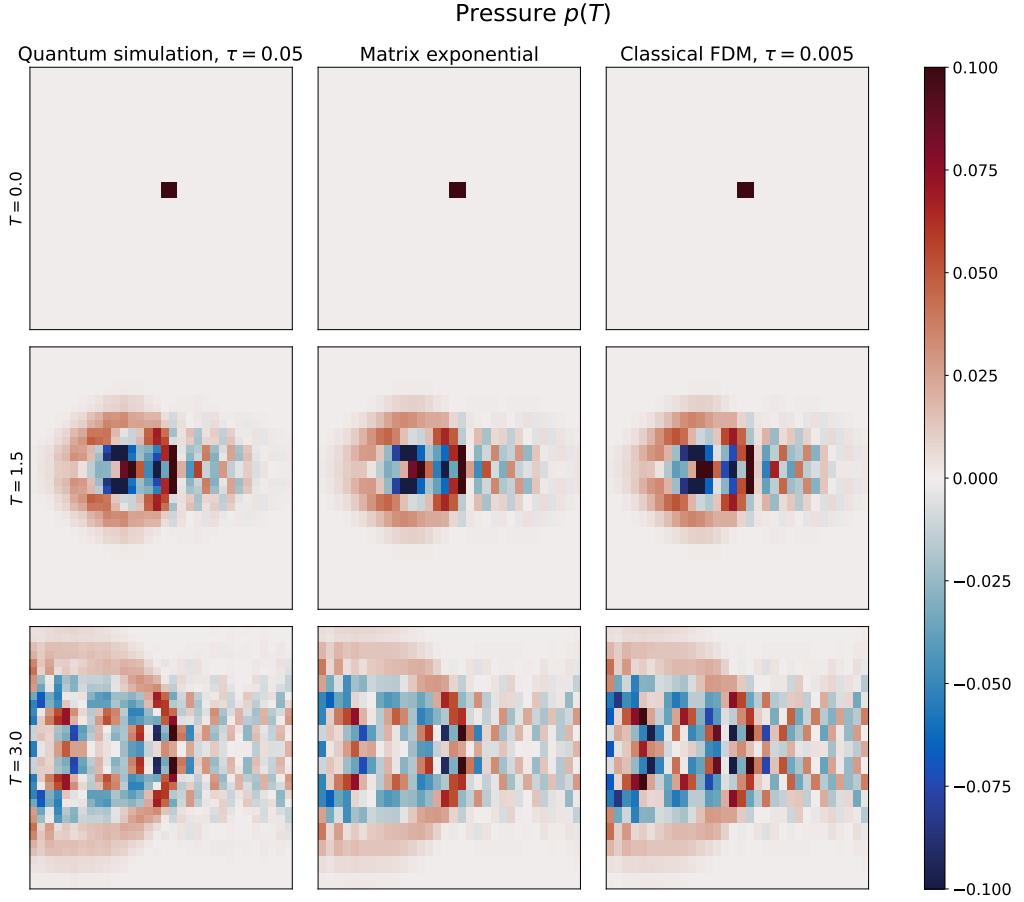


Figure 8: Numerical comparison of the pressure component of the solution at the different time T between quantum simulation (this work), exact matrix exponential solution and the classical finite difference method (FDM) for $n = 5$.

VI. NUMERICAL SIMULATION AND EXPERIMENTS

Here we show the results of executing the obtained quantum circuits on an ideal state-vector simulator, and compare them to the classical finite difference method (FDM) using the forward Euler scheme. We found that the FDM scheme required an order of magnitude smaller time step than used in the quantum simulation τ to maintain numerical stability. For this reason we also run a direct (classical) matrix exponential solution $e^{At}f(0)$, equivalent to the other two in the limit $\tau \rightarrow 0$. In all of the plots we show the pressure component p of the solution (note that in the LEE p is actually the deviation from the mean pressure, and can assume both positive and negative values).

If not stated otherwise, for the initial conditions we consider a single high pressure point-source of noise. We put $u(0) = v(0) = 0$ everywhere and set $p(0) = 0$ in all points except for a small 2×2 square in the center, where the pressure is set to $p = 0.5$. Such initial conditions are easily prepared with several X and H gates. For more complex initial conditions shallow circuits may be

prepared using Tensor Network approaches [54].

In Fig. 8 we compare the three methods discussed. We set $n = 5$, $c = \bar{\rho} = 1$, $\bar{u} = -1$ and $l = 0.25$. The quantum and classical solutions closely agree. Note though that in the quantum case we used a time step $\tau = 0.05$, while in the classical numerical scheme we required a finer step of $\tau = 0.005$ to maintain numerical stability (see Fig. 15). This is because, unlike the classical one, the quantum solution only contains the Trotter error, which, furthermore, in some cases can even be independent of system size and simulation time [55]. The solution error (in the pressure distribution) measured as the L_2 distance between either the quantum or classical FDM scheme and the exact matrix exponential is shown in Fig. 9, and demonstrates the advantage of the quantum approach.

Further, in Fig. 1 we show the results of executing the quantum circuit derived using our method for the 2D pressure wave propagating on a constant background constant flow and scattering from an impenetrable object inside the environment. The object in this case is a low-scale representation of an airfoil. To simplify the circuit construction we assume that the airfoil does not disturb the uniform background flow, but forms a reflective

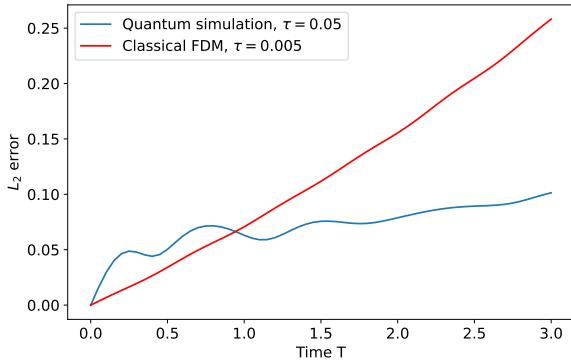


Figure 9: The L_2 -error of the quantum and classical FDM solutions for the pressure $p(T)$ in Fig.8 with respect to the exact matrix exponential solution. Note that the classical FDM required an order of magnitude finer time step to maintain stability (see also Fig.15).

boundary for the acoustic wave. Nonuniform background flows can be implemented using methods described in Ref.[35]. We set $c = \bar{\rho} = 1$, $\bar{u} = 2$, $\tau = 0.05$, $l = 0.5$. To avoid reflection from the boundaries of the domain (where we also have Dirichlet boundary conditions) we perform the simulation on a large 512×512 grid ($n = 9$) and observe the wave propagating before it reaches the edges of the environment.

While above we considered square domains and single-point sources, this is not a limitation. In Fig. 10 we show the results of executing quantum circuits generated for the propagation of a 2D acoustic wave inside an elongated pipe with reflective boundaries. The wave itself is generated by two sources with different intensities, and propagates on top of a uniform constant fluid flow. We used $c = \bar{\rho} = 1$, $\bar{u} = 0.5$, $l = 0.5$, $\tau = 0.05$, and employed 8 qubits for x -direction and 6 qubits for y -direction discretization.

Finally, to illustrate the flexibility of our methods, in Appendix F we show quantum simulations of the 2D acoustic wave propagation with circuits derived for arbitrarily complex shapes (see Fig. 14).

VII. CONCLUSIONS

We investigated extending the recently introduced methods of solving PDEs via mapping them to a Hamiltonian evolution problem to realistic settings. Towards this goal we devised a method to generate explicit quantum circuit incorporating in the discretized derivative operators boundary conditions for arbitrarily complex shapes. This allows modeling phenomena such as acoustic wave scattering from airfoils. We showed that our construction does not change the asymptotic complexity of the circuit, as long as the obstacle is implemented to a finite resolution, preserving the exponential advantage of

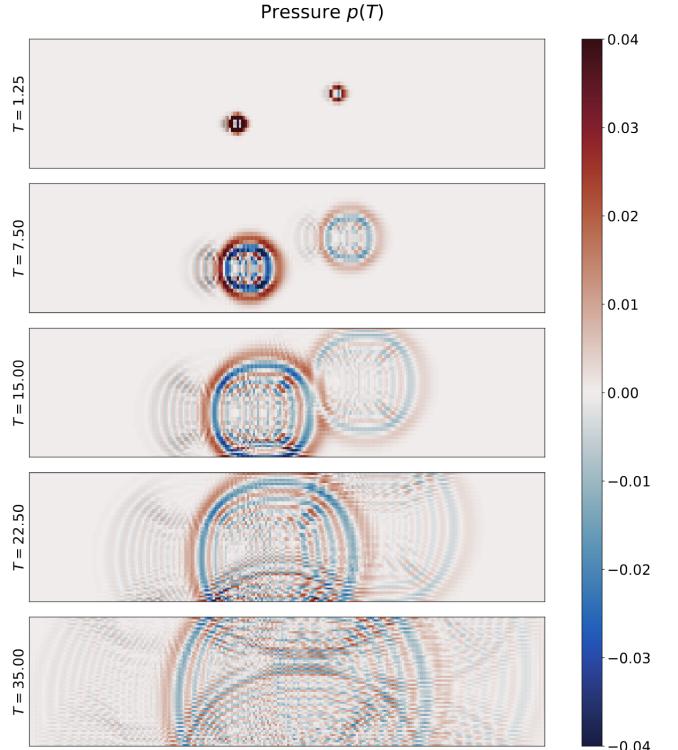


Figure 10: Quantum simulation of a 2D acoustic wave generated by two point sources of differing intensity propagating on top of a constant fluid flow in a pipe. We used 256×64 grid and reflective boundaries.

the free-space quantum method over classical solutions.

We further applied this technique to the problem of simulating the linearized Euler equations (LEE) Eq. 22. We demonstrated that in the conservative regime LEE with background flow ($c = 1/\bar{\rho}$) can be implemented as a Hamiltonian evolution, and constructed explicit quantum circuits for it. We further incorporated the boundary conditions corresponding to obstacles placed in the flow and simulated the quantum solution to acoustic wave scattering from them. While the general LEE with background flow are non-conservative, the method we proposed can be used to implement the key building blocks of the more general solutions, based *e.g.* on the linear combination of unitaries (LCU) or the ‘‘Schroödingerization’’ methods ([36, 40, 42]). All throughout we took care to describe optimized gate constructions, reducing both the theoretical as well practical implementation errors.

We expect that the methods we describe as well as the detailed circuit constructions will prove very useful to researchers investigating solving PDEs on the existing and upcoming quantum computers. Our intention was to reduce some of the complexities of implementing computations of theoretical or applied relevance on the QPUs. Many more such problems remain outstanding, and require solution before the asymptotic advantages of quantum PDE solving methods can materialize

in relevant cases. A key example is the readout problem of avoiding the full state tomography of the computed

PDE solution. Despite some interesting ideas [56–59] it remains one of the major challenges, and we would like to draw researchers attention to this class of problems.

-
- [1] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.
 - [2] Dolev Bluvstein, Simon J Evered, Alexandra A Geim, Sophie H Li, Hengyun Zhou, Tom Manovitz, Sepehr Ebadi, Madelyn Cain, Marcin Kalinowski, Dominik Hangleiter, et al. Logical quantum processor based on reconfigurable atom arrays. *Nature*, 626(7997):58–65, 2024.
 - [3] Rajeev Acharya et al. Quantum error correction below the surface code threshold. *Nature*, 638(8052):920–926, 2025.
 - [4] Andrew M Childs and Nathan Wiebe. Hamiltonian simulation using linear combinations of unitary operations. *arXiv:1202.5822*, 2012.
 - [5] Y. Cao, A. Papageorgiou, I. Petras, J. Traub, and S. Kais. Quantum algorithm and circuit design solving the Poisson equation. *New. J. Phys.*, 15:013021, 2013.
 - [6] Guang Hao Low and Isaac L. Chuang. Optimal hamiltonian simulation by quantum signal processing. *Phys. Rev. Lett.*, 118:010501, Jan 2017.
 - [7] A. W. Harrow, A. Hassidim, and S. Lloyd. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.*, 103:150502, 2009.
 - [8] John M. Martyn, Zane M. Rossi, Andrew K. Tan, and Isaac L. Chuang. Grand unification of quantum algorithms. *PRX Quantum*, 2:040203, 2021.
 - [9] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In *Proceedings of the 51st annual ACM SIGACT symposium on theory of computing*, pages 193–204, 2019.
 - [10] Danial Motagh and Nathan Wiebe. Generalized quantum signal processing. *PRX Quantum*, 5(2):020368, 2024.
 - [11] A. D. B. Zamora, L. Budinski, O. Niemimäki, and V. Lahtinen. Efficient quantum lattice gas automata. *Computers & Fluids*, 286:106476, 2025.
 - [12] Alexander Engel, Graeme Smith, and Scott E. Parker. Quantum algorithm for the vlasov equation. *Physical Review A*, 100(6), December 2019.
 - [13] D. W. Berry. High-order quantum algorithm for solving linear differential equation. *J. Phys. A*, 47:105301, 2014.
 - [14] Ljubomir Budinski. Quantum algorithm for the Navier–Stokes equations by using the streamfunction–vorticity formulation and the lattice Boltzmann method. *International Journal of Quantum Information*, 20(02):2150039, 2022.
 - [15] A. Montanaro and S. Pallister. Quantum algorithms and the finite element method. *Phys. Rev. A*, 93:032324, 2016.
 - [16] D. W. Berry, A. M. Childs, A. Ostrander, and G. Wang. Quantum algorithm for linear differential equations with exponentially improved dependence on precision. *Commun. Math. Phys.*, 356:1057–1081, 2017.
 - [17] A. Engel, G. Smith, and S. E. Parker. Quantum algorithm for the Vlasov equation. *Phys. Rev. A*, 100:062315, 2019.
 - [18] N. Linden, A. Montanaro, and C. Shao. Quantum vs. classical algorithms for solving the heat equation. *Commun. Math. Phys.*, 395:601–641, 2022.
 - [19] A. M. Childs and J.-P. Liu. Quantum spectral methods for differential equations. *Commun. Math. Phys.*, 375:1427–1457, 2020.
 - [20] A. M. Childs, J.-P. Liu, and A. Ostrander. High-precision quantum algorithms for partial differential equations. *Quantum*, 5:574, 2021.
 - [21] F. Golse, S. Jin, and N. Liu. Quantum algorithms for uncertainty quantification: Application to partial differential equations. *Sci. China Phys. Mech. Astron.*, 68:104704, 2025.
 - [22] S. Jin, N. Liu, and Y. Yu. Time complexity analysis of quantum algorithms via linear representations for nonlinear ordinary and partial differential equations. *J. Comput. Phys.*, 487:112149, 2023.
 - [23] F. Y. Leong, D. E. Koh, W.-B. Ewe, and J. F. Kong. Variational quantum simulation of partial differential equations: Applications in colloidal transport. *International Journal of Numerical Methods for Heat and Fluid Flow*, 33:3669–3690, 2023.
 - [24] M. Lubasch, J. Joo, P. Moinier, M. Kiffner, and D. Jaksch. Variational quantum algorithms for nonlinear problems. *Phys. Rev. A*, 101:010301, 2020.
 - [25] M. Ali and M. Kabel. Performance study of variational quantum algorithms for solving the Poisson equation on a quantum computer. *Phys. Rev. Appl.*, 20:014054, 2023.
 - [26] Y. Sato, R. Kondo, S. Koide, H. Takamatsu, and N. Imoto. Variational quantum algorithm based on the minimum potential energy for solving the Poisson equation. *Phys. Rev. A*, 104:052409, 2021.
 - [27] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles. Variational quantum algorithms. *Nat. Rev. Phys.*, 3:625–644, 2021.
 - [28] J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, V. Li, E. Grant, L. Wossnig, I. Rungger, G. H. Booth, and J. Tennyson. The variational quantum eigensolver: A review of methods and best practices. *Phys. Rep.*, 986:1–128, 2022.
 - [29] Y. Sato, H. C. Watanabe, R. Raymond, R. Kondo, K. Wada, K. Endo, M. Sugawara, and N. Yamamoto. Variational quantum algorithm for generalized eigenvalue problems and its application to the finite-element method. *Phys. Rev. A*, 108:022429, 2023.
 - [30] Kiichiro Toyoizumi, Naoki Yamamoto, and Kazuo Hoshino. Hamiltonian simulation using quantum singular value transformation: complexity analysis and application to the linearized vlasov-poisson equation, 2023.
 - [31] R. Babbush, D. W. Berry, R. Kothari, R. D. Somma, and N. Wiebe. Exponential quantum speedup in simulating coupled classical oscillators. *Phys. Rev. X*, 13:041041,

- 2023.
- [32] S. Jin, N. Liu, and Y. Yu. Time complexity analysis of quantum difference methods for linear high dimensional and multiscale partial differential equations. *J. Comput. Phys.*, 471:111641, 2022.
- [33] S. Jin and N. Liu. Quantum simulation of discrete linear dynamical systems and simple iterative methods in linear algebra. *Proc. R. Soc. A.*, 480:20230370, 2024.
- [34] Y. Sato, R. Kondo, I. Hamamura, T. Onodera, and N. Yamamoto. Hamiltonian simulation for hyperbolic partial differential equations by scalable quantum circuits. *Phys. Rev. Res.*, 6:033246, 2024.
- [35] Y. Sato, H. Tezuka, R. Kondo, and N. Yamamoto. Quantum algorithm for partial differential equations of non-conservative systems with spatially varying parameters. *Phys. Rev. Appl.*, 23:014063, 2025.
- [36] S. Jin, N. Liu, and C. Ma. On Schrödingerization based quantum algorithms for linear dynamical systems with inhomogeneous terms. *SIAM J. Numer. Anal.*, 63:1861–1885, 2025.
- [37] S. Jin, X. Li, N. Liu, and Y. Yu. Quantum simulation for quantum dynamics with artificial boundary conditions. *SIAM Journal on Scientific Computing*, 46:B403–B421, 2024.
- [38] S. Jin, N. Liu, and Y. Yu. Quantum simulation of partial differential equations via Schrödingerization. *Phys. Rev. Lett.*, 133:230602, 2024.
- [39] J. M. Leamer, A. B. Magann, D. I. Bondar, and G. McCaul. Quantum dynamical emulation of imaginary time evolution. *arXiv:2403.03350*, 2024.
- [40] N. Guseynov, X. Huang, and N. Liu. Gate construction of block-encoding for Hamiltonians needed for simulating partial differential equations. *Phys. Rev. Res.*, 7:033100, 2025.
- [41] I. Novikau and I. Joseph. Quantum algorithm for the advection-diffusion equation and the Koopman-von Neumann approach to nonlinear dynamical systems. *Comput. Phys. Commun.*, 309:109498, 2025.
- [42] D. An, J.-P. Liu, and L. Lin. Linear combination of Hamiltonian simulation for nonunitary dynamics with optimal state preparation cost. *Phys. Rev. Lett.*, 131:150603, 2023.
- [43] P. C. S. Costa, S. Jordan, and A. Ostrander. Quantum algorithm for simulating the wave equation. *Phys. Rev. A*, 99:012323, 2019.
- [44] 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, February 2021.
- [45] Lewis Wright, Conor Mc Keever, Jeremy T. First, Rory Johnston, Jeremy Tillay, Skylar Chaney, Matthias Rosenkranz, and Michael Lubasch. Noisy intermediate-scale quantum simulation of the one-dimensional wave equation. *Physical Review Research*, 6(4), November 2024.
- [46] Xiao-Ming Zhang, Tongyang Li, and Xiao Yuan. Quantum state preparation with optimal circuit depth: Implementations and applications. *Physical Review Letters*, 129(23), November 2022.
- [47] Gary J Mooney, Gregory A L White, Charles D Hill, and Lloyd C L Hollenberg. Generation and verification of 27-qubit greenberger-horne-zeilinger states in a superconducting quantum computer. *Journal of Physics Communications*, 5(9):095004, September 2021.
- [48] M. Suzuki. General theory of fractal path integrals with applications to many-body theories and statistical physics. *J. Math. Phys.*, 32:400–407, 1991.
- [49] J. D. S. Silva, T. M. D. Azevedo, I. F. Araujo, and A. J. da Silva. Linear decomposition of approximate multi-controlled single qubit gates. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 44:1304–1312, 2024.
- [50] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter. Elementary gates for quantum computation. *Phys. Rev. A*, 52:3457–3467, 1995.
- [51] A. J. da Silva and D. K. Park. Linear-depth quantum circuits for multiqubit controlled gates. *Phys. Rev. A*, 106:042602, 2022.
- [52] R. Vale, T. M. D. Azevedo, I. C. S. Araújo, I. F. Araujo, and A. J. da Silva. Decomposition of multi-controlled special unitary single-qubit gates. *arXiv:2302.06377*, 2023.
- [53] R. Iten, R. Colbeck, I. Kukuljan, J. Home, and M. Christandl. Quantum circuits for isometries. *Phys. Rev. A*, 93:032318, 2016.
- [54] Vladyslav Bohun, Illia Lukin, Mykola Luhanko, Georgios Korpas, Philippe JS De Brouwer, Mykola Maksymenko, and Maciej Koch-Janusz. Entanglement scaling in matrix product state representation of smooth functions and their shallow quantum circuit approximations. *arXiv:2412.05202*, 2024.
- [55] Markus Heyl, Philipp Hauke, and Peter Zoller. Quantum localization bounds trotter errors in digital quantum simulation. *Science advances*, 5(4):eaau8342, 2019.
- [56] Z. Song, R. Deaton, B. Gard, and S. H. Bryngelson. Incompressible Navier-Stokes solve on noisy quantum hardware via a hybrid quantum-classical scheme. *Computers & Fluids*, 288:106507, 2025.
- [57] A. Termanova, A. Melnikov, E. Mamenchikov, N. Belokonev, S. Dolgov, A. Berezutskii, R. Ellerbrock, C. Mansell, and M. R. Perelshtein. Tensor quantum programming. *New J. Phys.*, 26:123019, 2024.
- [58] C. A. Williams, S. Scali, A. A. Gentile, D. Berger, and O. Kyriienko. Addressing the readout problem in quantum differential equation algorithms with quantum scientific machine learning. *arXiv:2411.14259*, 2024.
- [59] Yu Fang, Cheng Xue, Tai-Ping Sun, Xiao-Fan Xu, Xi-Ning Zhuang, Yun-Jie Wang, Chuang-Chao Ye, Teng-Yang Ma, Jia-Xuan Zhang, Huan-Yu Liu, Yu-Chun Wu, Zhao-Yun Chen, and Guo-Ping Guo. Polyqrom: Orthogonal-polynomial-based quantum reduced-order model for flow field analysis, 2025.
- [60] A. M. Childs, Y. Su, M. C. Tran, N. Wiebe, and S. Zhu. Theory of Trotter error with commutator scaling. *Phys. Rev. X*, 11:011020, 2021.
- [61] J. Hu, S. Jin, N. Liu, and L. Zhang. Quantum circuits for partial differential equations via Schrödingerisation. *Quantum*, 8:1563, 2024.
- [62] I. Joseph. Koopman–von Neumann approach to quantum simulation of nonlinear classical dynamics. *Phys. Rev. Res.*, 2:043102, 2020.
- [63] S. Jin, N. Liu, and C. Ma. Quantum simulation of Maxwell’s equations via Schrödingerisation. *ESAIM: M2AN*, 58:1853–1879, 2024.
- [64] S. Jin, N. Liu, and Y. Yu. Quantum simulation of the Fokker-Planck equation via Schrödingerization. *arXiv preprint arXiv:2404.13585*, 2024.

- [65] S. Jin, N. Liu, and Y. Yu. Quantum simulation of partial differential equations: Applications and detailed analysis. *Phys. Rev. A*, 108:032603, 2023.
- [66] M. Motta, C. Sun, A. T. K. Tan, M. J. O'Rourke, E. Ye, A. J. Minnich, F. G. S. L. Brandão, and G. K.-L. Chan. Determining eigenstates and thermal states on a quantum computer using quantum imaginary time evolution. *Nat. Phys.*, 16:205–210, 2020.
- [67] S. McArdle, T. Jones, S. Endo, Y. Li, S. C. Benjamin, and X. Yuan. Variational ansatz-based quantum simulation of imaginary time evolution. *npj Quantum Information*, 5:75, 2019.
- [68] C. Leadbeater, N. Fitzpatrick, D. M. Ramo, and A. J. W. Thom. Non-unitary Trotter circuits for imaginary time evolution. *Quantum Sci. Technol.*, 9:045007, 2024.
- [69] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe. Quantum singular value transformation and beyond: exponential

improvements for quantum matrix arithmetics. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC '19, page 193–204. ACM, 2019.

Appendix A: Circuit derivation for the conservative system with central finite difference operators

Here we explicitly derive the quantum circuit for the evolution described by the operator in Eq. 23 with $c = 1/\bar{\rho}$. Firstly, we rewrite some of the factors appearing in the definition of the $H_{x,j}$ and $H_{y,j}$ terms in the Hamiltonian Eq. 25 using the operators $U_j(\lambda)$ given in Eq. 13:

$$\begin{aligned}
& iI^{\otimes(n-j)} \otimes (\sigma_{01} \otimes \sigma_{10}^{\otimes(j-1)} - \sigma_{10} \otimes \sigma_{01}^{\otimes(j-1)}) = iI^{\otimes(n-j)} \otimes \left(|0\rangle\langle 1| \otimes |1\rangle\langle 0|^{\otimes(j-1)} - |1\rangle\langle 0| \otimes |0\rangle\langle 1|^{\otimes(j-1)} \right) \\
&= iI^{\otimes(n-j)} \otimes \left(|0\rangle \otimes |1\rangle^{\otimes(j-1)}\langle 1| \otimes |0|^{\otimes(j-1)} - |1\rangle \otimes |0\rangle^{\otimes(j-1)}\langle 0| \otimes |1|^{\otimes(j-1)} \right) \\
&= I^{\otimes(n-j)} \otimes \left(\frac{|0\rangle \otimes |1\rangle^{\otimes(j-1)} - i|1\rangle \otimes |0\rangle^{\otimes(j-1)}}{\sqrt{2}} \frac{\langle 0| \otimes \langle 1|^{\otimes(j-1)} + i\langle 1| \otimes \langle 0|^{\otimes(j-1)}}{\sqrt{2}} \right. \\
&\quad \left. - \frac{|0\rangle \otimes |1\rangle^{\otimes(j-1)} + i|1\rangle \otimes |0\rangle^{\otimes(j-1)}}{\sqrt{2}} \frac{\langle 0| \otimes \langle 1|^{\otimes(j-1)} - i\langle 1| \otimes \langle 0|^{\otimes(j-1)}}{\sqrt{2}} \right) \\
&= I^{\otimes(n-j)} \otimes U_j(-\pi/2) \left(Z \otimes \sigma_{11}^{\otimes(j-1)} \right) U_j(-\pi/2)^\dagger. \tag{A1}
\end{aligned}$$

The effect of unitary matrix $U_j(\lambda)$ (see Fig. 2 for the circuit representation) is to perform a basis change to the modified Bell basis so that:

$$\begin{aligned}
& U_j(\lambda)|0\rangle \otimes |1\rangle^{\otimes(j-1)} \\
&= \frac{1}{\sqrt{2}} \left(|0\rangle \otimes |1\rangle^{\otimes(j-1)} + e^{i\lambda}|1\rangle \otimes |0\rangle^{\otimes(j-1)} \right)
\end{aligned}$$

and

$$\begin{aligned}
& U_j(\lambda)|1\rangle \otimes |1\rangle^{\otimes(j-1)} \\
&= \frac{1}{\sqrt{2}} \left(|0\rangle \otimes |1\rangle^{\otimes(j-1)} - e^{i\lambda}|1\rangle \otimes |0\rangle^{\otimes(j-1)} \right).
\end{aligned}$$

Using Eq. A1 we immediately obtain Eqs. 12 and 21.

Continuing, the LEE Hamiltonian Eq. 23 can now be rewritten in the following form:

$$H = H_x^{diag} + H_x^{off} + H_y^{off}, \tag{A2}$$

where

$$H_x^{diag} = -\frac{\bar{u}}{2l} I^{a_1} \otimes I^{a_2} \otimes \sum_{j=1}^n I^{\otimes(n-j)} \otimes U_j(-\pi/2) \left(Z \otimes \sigma_{11}^{\otimes(j-1)} \right) U_j(-\pi/2)^\dagger \otimes I^{\otimes n}, \tag{A3}$$

$$H_x^{off} = -\frac{1}{2\bar{\rho}} \sigma_{00}^{a_1} \otimes X^{a_2} \otimes \sum_{j=1}^n I^{\otimes(n-j)} \otimes U_j(-\pi/2) \left(Z \otimes \sigma_{11}^{\otimes(j-1)} \right) U_j(-\pi/2)^\dagger \otimes I^{\otimes n}, \tag{A4}$$

$$H_y^{off} = -\frac{1}{2\bar{\rho}} X^{a_1} \otimes \sigma_{00}^{a_2} \otimes I^{\otimes n} \otimes \sum_{j=1}^n I^{\otimes(n-j)} \otimes U_j(-\pi/2) \left(Z \otimes \sigma_{11}^{\otimes(j-1)} \right) U_j(-\pi/2)^\dagger. \tag{A5}$$

Above we split the Hamiltonian into a diagonal part H_x^{diag} and two off-diagonal ones, corresponding to the

derivatives x and y , respectively. Using the Lie-Trotter-Suzuki decomposition we obtain the first-order approximation the time evolution operator:

$$\begin{aligned} \exp(-iH\tau) &\approx \exp\left(-iH_x^{diag}\tau\right) \exp\left(-iH_x^{off}\tau\right) \exp\left(-iH_y^{off}\tau\right) \\ &\approx \prod_{j=1}^n \exp\left(i\frac{\bar{u}\tau}{2l} I^{a_1} \otimes I^{a_2} \otimes I^{\otimes(n-j)} \otimes U_j(-\pi/2) \left(Z \otimes \sigma_{11}^{\otimes(j-1)}\right) U_j(-\pi/2)^\dagger \otimes I^{\otimes n}\right) \\ &\quad \times \prod_{j=1}^n \exp\left(i\frac{\tau}{2\rho l} \sigma_{00}^{a_1} \otimes X^{a_2} \otimes I^{\otimes(n-j)} \otimes U_j(-\pi/2) \left(Z \otimes \sigma_{11}^{\otimes(j-1)}\right) U_j(-\pi/2)^\dagger \otimes I^{\otimes n}\right) \\ &\quad \times \prod_{j=1}^n \exp\left(i\frac{\tau}{2\rho l} X^{a_1} \otimes \sigma_{00}^{a_2} \otimes I^{\otimes n} \otimes I^{\otimes(n-j)} \otimes U_j(-\pi/2) \left(Z \otimes \sigma_{11}^{\otimes(j-1)}\right) U_j(-\pi/2)^\dagger\right). \end{aligned} \quad (\text{A6})$$

Using the identities

$$\begin{aligned} \exp\left(-iUAU^\dagger\right) &= U \exp(-iA) U^\dagger, \\ X = HZH, \quad \sigma_{00} &= X\sigma_{11}X, \end{aligned} \quad (\text{A7})$$

for a unitary matrix U , the above expression is rewritten as

$$\begin{aligned} \exp(-iH\tau) &\approx \prod_{j=1}^n I^{a_1} \otimes I^{a_2} \otimes I^{\otimes(n-j)} \otimes U_j(-\pi/2) \exp\left(i\frac{\bar{u}\tau}{2l} Z \otimes \sigma_{11}^{\otimes(j-1)}\right) U_j(-\pi/2)^\dagger \otimes I^{\otimes n} \\ &\quad \times \prod_{j=1}^n I^{\otimes(n-j)} \otimes (X^{a_1} \otimes H^{a_2} \otimes U_j(-\pi/2)) \exp\left(i\frac{\tau}{2\rho l} \sigma_{11}^{a_1} \otimes Z^{a_2} \otimes Z \otimes \sigma_{11}^{\otimes(j-1)}\right) (U_j(-\pi/2)^\dagger \otimes X^{a_1} \otimes H^{a_2}) \otimes I^{\otimes n} \\ &\quad \times \prod_{j=1}^n I^{\otimes n} \otimes I^{\otimes(n-j)} \otimes (H^{a_1} \otimes X^{a_2} \otimes U_j(-\pi/2)) \exp\left(i\frac{\tau}{2\rho l} Z^{a_1} \otimes \sigma_{11}^{a_2} \otimes Z \otimes \sigma_{11}^{\otimes(j-1)}\right) (U_j(-\pi/2)^\dagger \otimes H^{a_1} \otimes X^{a_2}). \end{aligned} \quad (\text{A8})$$

We now observe that terms in the exponents are generators for multi-controlled RZ(γ) and RZZ(γ) rotation gates. Note that RZ(γ) = $\exp(-i\gamma Z/2)$, therefore division by 2 in the central difference scheme is absorbed into the gate. With the above, finally, the circuit can be expressed in the form given in Eqs. 26, 27, 28. In Fig. 6 we also grouped up some terms to reduce amount of $U_j(\lambda)$ gates.

Appendix B: Circuit derivation for the conservative system with D^+ and D^- difference operators

In this appendix we derive the quantum circuit for the Euler equations with the background flow (in the conservative regime $c = 1/\bar{\rho}$) with first order derivative operators. We use the D^\pm difference operator to quantize

derivatives of the diagonal part and D^+ , D^- difference operators to quantize derivatives of the off-diagonal part of the matrix Eq. 23. The D^+ and D^- are used for the super-diagonal and sub-diagonal terms, respectively. Then the evolution matrix remains Hermitian and can be expressed as:

$$\begin{aligned} \tilde{H} &= -i \left[\bar{u} I^{a_1} \otimes I^{a_2} \otimes D^\pm \otimes I^{\otimes n} \right. \\ &\quad + \frac{1}{\bar{\rho}} \left(\sigma_{00}^{a_1} \otimes \sigma_{01}^{a_2} \otimes D^+ \otimes I^{\otimes n} + \sigma_{00}^{a_1} \otimes \sigma_{10}^{a_2} \otimes D^- \otimes I^{\otimes n} \right) \\ &\quad \left. + \frac{1}{\bar{\rho}} \left(\sigma_{01}^{a_1} \otimes \sigma_{00}^{a_2} \otimes I^{\otimes n} \otimes D^+ + \sigma_{10}^{a_1} \otimes \sigma_{00}^{a_2} \otimes I^{\otimes n} \otimes D^- \right) \right]. \end{aligned} \quad (\text{B1})$$

Note that the first line gives the diagonal elements with the derivatives w.r.t. x only, while the second and third lines contain off-diagonal elements with derivatives w.r.t. to x and y , respectively. Similarly to the previous

section, the Hamiltonian Eq. B1 is also rewritten using the $U_j(\lambda)$ operators defined in Eq. 13. The first term with the D^\pm operator was already derived in Eq. A3. Let us now rewrite the x derivative off-diagonal terms.

Substituting the representation of the D^+ and D^- operators with Dirichlet boundary conditions Eq. 7 and the definitions of the shift operators Eq. 5 into these terms we obtain:

$$\begin{aligned} \tilde{H}_x^{off} &= -\frac{i}{\bar{\rho}} (\sigma_{00}^{a_1} \otimes \sigma_{01}^{a_2} \otimes D^+ \otimes I^{\otimes n} + \sigma_{00}^{a_1} \otimes \sigma_{10}^{a_2} \otimes D^- \otimes I^{\otimes n}) \\ &= -\frac{i}{\bar{\rho}l} (\sigma_{00}^{a_1} \otimes (\sigma_{01}^{a_2} \otimes S^- - \sigma_{10}^{a_2} \otimes S^+) - \sigma_{00}^{a_1} \otimes (\sigma_{01}^{a_2} - \sigma_{10}^{a_2}) \otimes I^{\otimes n}) \otimes I^{\otimes n} \\ &= -\frac{1}{\bar{\rho}l} \sum_{j=1}^n I^{\otimes(n-j)} \otimes \left(X^{a_1} \otimes X^j \otimes U_{a_2} \left(-\frac{\pi}{2} \right) \right) (\sigma_{11}^{a_1} \otimes Z^{a_2} \otimes \sigma_{11}^{\otimes j}) \left(U_{a_2} \left(-\frac{\pi}{2} \right)^\dagger \otimes X^j \otimes X^{a_1} \right) \otimes I^{\otimes n} \\ &\quad - \frac{1}{\bar{\rho}l} X^{a_1} (\sigma_{11}^{a_1} \otimes Y^{a_2}) X^{a_1} \otimes I^{\otimes n} \otimes I^{\otimes n}, \end{aligned} \quad (B2)$$

where we used the identities

$$\sigma_{01}^j = X^j \sigma_{10}^j X^j, \quad \sigma_{01}^{a_2} - \sigma_{10}^{a_2} = iY^{a_2}$$

and Eq. A1. Here, an upper index j refers to action on

the j -th qubit in q_x register and Y is a Pauli-Y matrix. With a small abuse of notation we also denoted by $U_{a_2}(\lambda)$ a $(j+1)$ -qubit operator Eq. 13 acting on the first j qubits and an ancillary qubit a_2 .

For the y -derivatives term of the Hamiltonian Eq. B1 we obtain a similar result:

$$\begin{aligned} \tilde{H}_y^{off} &= -\frac{i}{\bar{\rho}} (\sigma_{01}^{a_1} \otimes \sigma_{00}^{a_2} \otimes I^{\otimes n} \otimes D^+ + \sigma_{10}^{a_1} \otimes \sigma_{00}^{a_2} \otimes I^{\otimes n} \otimes D^-) \\ &= -\frac{1}{\bar{\rho}l} I^{\otimes n} \otimes \sum_{j=1}^n I^{\otimes(n-j)} \otimes \left(X^{a_2} \otimes X^j \otimes U_{a_1} \left(-\frac{\pi}{2} \right) \right) (Z^{a_1} \otimes \sigma_{11}^{a_2} \otimes \sigma_{11}^{\otimes j}) \left(U_{a_1} \left(-\frac{\pi}{2} \right)^\dagger \otimes X^j \otimes X^{a_2} \right) \\ &\quad - \frac{1}{\bar{\rho}l} I^{\otimes n} \otimes X^{a_2} (Y^{a_1} \otimes \sigma_{11}^{a_2}) X^{a_2} \otimes I^{\otimes n}. \end{aligned} \quad (B3)$$

Using first-order Lie-Trotter-Suzuki decomposition the time-evolution operator takes the form

$$\begin{aligned} \tilde{V}(\tau) &= \exp(-i\tilde{H}\tau) \\ &\approx \exp(-iH_x^{diag}\tau) \exp(-i\tilde{H}_x^{off}\tau) \exp(-i\tilde{H}_y^{off}\tau) \\ &\approx \prod_{j=1}^n \tilde{W}_{x,j} \prod_{j=1}^n \tilde{W}_{y,j}, \end{aligned} \quad (B4)$$

where the operators $\tilde{W}_{x,j}$ and $\tilde{W}_{y,j}$ take the form

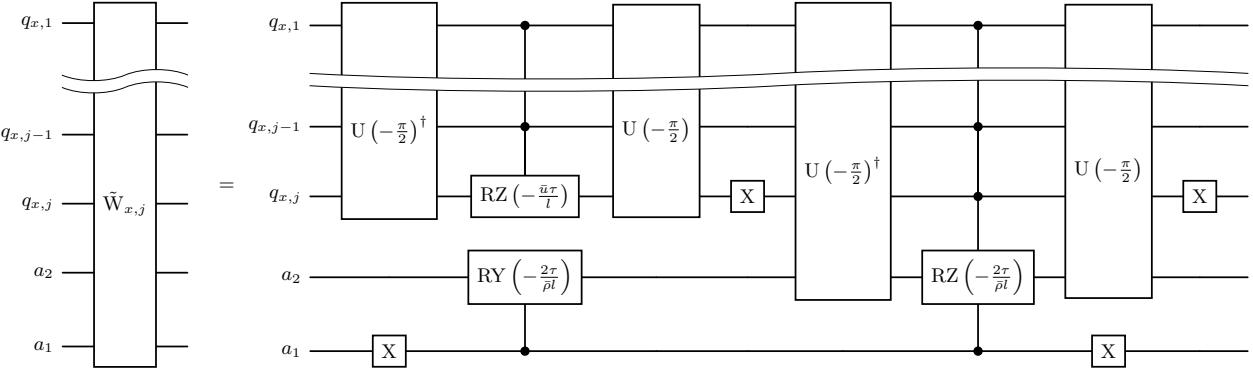


Figure 11: Quantum circuit for the $\tilde{W}_{x,j}$ operator acting on the $q_{x,1}, \dots, q_{x,j}, a_1, a_2$ register.

$$\begin{aligned}
 \tilde{W}_{x,j} &= I^{a_1} \otimes I^{a_2} \otimes I^{\otimes(n-j)} \otimes U_j(-\pi/2) \exp\left(i\frac{\bar{u}\tau}{2l} Z \otimes \sigma_{11}^{\otimes(j-1)}\right) U_j(-\pi/2)^\dagger \otimes I^{\otimes n} \\
 &\times I^{\otimes(n-j)} \otimes \left(X^{a_1} \otimes X^j \otimes U_{a_2}\left(-\frac{\pi}{2}\right)\right) \exp\left(i\frac{\tau}{\rho l} \sigma_{11}^{a_1} \otimes Z^{a_2} \otimes \sigma_{11}^{\otimes j}\right) \left(U_{a_2}\left(-\frac{\pi}{2}\right)^\dagger \otimes X^j \otimes X^{a_1}\right) \otimes I^{\otimes n} \\
 &\times X^{a_1} \exp\left(i\frac{\tau}{\rho l} \sigma_{11}^{a_1} \otimes Y^{a_2}\right) X^{a_1} \otimes I^{\otimes n} \otimes I^{\otimes n}, \\
 \tilde{W}_{y,j} &= I^{\otimes n} \otimes I^{\otimes(n-j)} \otimes \left(X^{a_2} \otimes X_j \otimes U_{a_1}\left(-\frac{\pi}{2}\right)\right) \exp\left(i\frac{\tau}{\rho l} Z^{a_1} \otimes \sigma_{11}^{a_2} \otimes \sigma_{11}^{\otimes j}\right) \left(U_{a_1}\left(-\frac{\pi}{2}\right)^\dagger \otimes X^j \otimes X^{a_2}\right) \\
 &\times I^{\otimes n} \otimes X^{a_2} \exp\left(i\frac{\tau}{\rho l} Y^{a_1} \otimes \sigma_{11}^{a_2}\right) X^{a_2} \otimes I^{\otimes n}.
 \end{aligned} \tag{B5}$$

In the above we recognize one of the factors as a controlled RY rotation with the control on the a_1 qubit and the target on a_2 :

$$\text{CRY}_{a_2}^{a_1}\left(-\frac{2\tau}{\rho l}\right) = \exp\left(i\frac{\tau}{\rho l} \sigma_{11}^{a_1} \otimes Y^{a_2}\right),$$

As before, we separately implement the x and y evolution terms $\tilde{W}_{x,j}$ (see Fig. 11) and $\tilde{W}_{y,j}$ (see Fig. 12). Note though the appearance of a factor of 2 in the rotation angles. Comparing obtained circuits to the ones with only D^\pm operator in the main text we see that no MCRZZ gates appear, which results in a more economical decomposition in terms of the elementary CNOT numbers. However, usage of three types of derivatives at once can disturb the symmetry of the solution.

Appendix C: Circuit derivation for the D^\pm operator with periodic boundary conditions

The central finite difference operator with periodic boundary conditions can be encoded on n qubits as fol-

lows :

$$D_P^\pm = \frac{1}{2l} \left(S^- - S^+ - \sigma_{01}^{\otimes n} + \sigma_{10}^{\otimes n} \right). \tag{C1}$$

In the first order Trotter approximation the unitary evolution it generates is given by a product of two factors:

$$\begin{aligned}
 e^{D_P^\pm \gamma} &= e^{-i(iD_P^\pm)\gamma} \approx \exp\left[-i\left(\frac{i\gamma}{2l}(S^- - S^+)\right)\right] \\
 &\times \exp\left[-i\left(\frac{i\gamma}{2l}(-\sigma_{01}^{\otimes n} + \sigma_{10}^{\otimes n})\right)\right],
 \end{aligned} \tag{C2}$$

where the first factor is given in the Eq. 12 and the second one can be shown in a similar way to be:

$$\begin{aligned}
 &\exp\left[-i\left(\frac{i\gamma}{2l}(-\sigma_{01}^{\otimes n} + \sigma_{10}^{\otimes n})\right)\right] \\
 &= \left(I \otimes X^{\otimes(n-1)}\right) U_n\left(-\frac{\pi}{2}\right) \text{MCRZ}_n^{1,\dots,n-1}\left(-\frac{\gamma}{l}\right) \\
 &\times U_n\left(-\frac{\pi}{2}\right)^\dagger \left(I \otimes X^{\otimes(n-1)}\right).
 \end{aligned} \tag{C3}$$

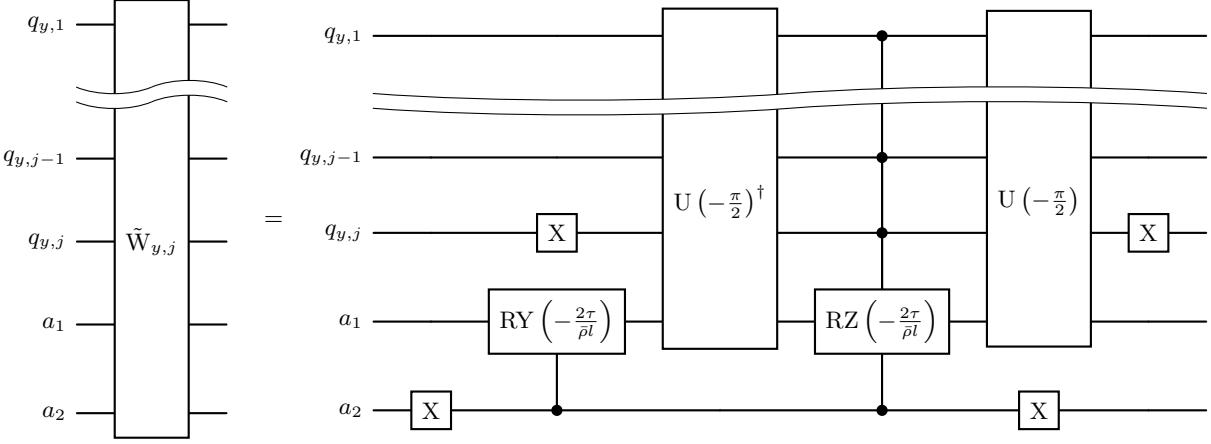


Figure 12: Quantum circuit for the $\tilde{W}_{y,j}$ operator acting on the $q_{y,1}, \dots, q_{y,j}, a_1, a_2$ register.

Here we used the following identity, which can be easily verified (see Eq. A1)

$$\begin{aligned} i(-\sigma_{01}^{\otimes n} + \sigma_{10}^{\otimes n}) &= \left(\mathbf{I} \otimes \mathbf{X}^{\otimes(n-1)}\right) \mathbf{U}_n \left(-\frac{\pi}{2}\right) \\ &\times \left(-Z \otimes \sigma_{11}^{\otimes(n-1)}\right) \mathbf{U}_n \left(-\frac{\pi}{2}\right)^\dagger \left(\mathbf{I} \otimes \mathbf{X}^{\otimes(n-1)}\right). \end{aligned} \quad (\text{C4})$$

To write down the Euler equations with periodic boundary conditions we only need to add to the Dirichlet boundary conditions case Eq. 24 an additional term

representing ‘‘corner’’ entries in the difference operator:

$$\begin{aligned} H_P = -\frac{i}{2l} &\left(\bar{u} \mathbf{I}^{a_1} \otimes \mathbf{I}^{a_2} + \frac{1}{\bar{\rho}} \sigma_{00}^{a_1} \otimes \mathbf{X}^{a_2} \right) \\ &\otimes (-\sigma_{01}^{\otimes n} + \sigma_{10}^{\otimes n}) \otimes \mathbf{I}^{\otimes n} \\ &- \frac{i}{2l\bar{\rho}} \mathbf{X}^{a_1} \otimes \sigma_{00}^{a_2} \otimes \mathbf{I}^{\otimes n} \otimes (-\sigma_{01}^{\otimes n} + \sigma_{10}^{\otimes n}). \end{aligned} \quad (\text{C5})$$

Making similar calculations as in the Appendix A, in the first order Trotter approximation we obtain the following evolution operator:

$$V(\tau) = e^{-iH\tau} \approx Q_y W_{y,P} Q_x W_{x,P}, \quad (\text{C6})$$

where Q_x and Q_y are defined by Eqs. 27 and 28, and the $W_{x,P}$, $W_{y,P}$ operators are given by

$$\begin{aligned} W_{x,P} &= \mathbf{I}^{a_1} \otimes \mathbf{I}^{a_2} \otimes \left(\mathbf{I} \otimes \mathbf{X}^{\otimes(n-1)}\right) \mathbf{U}_n \left(-\frac{\pi}{2}\right) \text{MCRZ}_n^{1,\dots,n-1} \left(\frac{\bar{u}\tau}{l}\right) \mathbf{U}_n \left(-\frac{\pi}{2}\right)^\dagger \left(\mathbf{I} \otimes \mathbf{X}^{\otimes(n-1)}\right) \\ &\times \left(\mathbf{I} \otimes \mathbf{X}^{\otimes(n-1)}\right) \left(\mathbf{X}^{a_1} \otimes \mathbf{H}^{a_2} \otimes \mathbf{U}_n \left(-\frac{\pi}{2}\right)\right) \text{MCRZZ}_{n,a_2}^{1,\dots,n-1,a_1} \left(\frac{\tau}{\bar{\rho}l}\right) \left(\mathbf{X}^{a_1} \otimes \mathbf{H}^{a_2} \otimes \mathbf{U}_n \left(-\frac{\pi}{2}\right)^\dagger\right) \left(\mathbf{I} \otimes \mathbf{X}^{\otimes(n-1)}\right) \otimes \mathbf{I}^{\otimes n}, \\ W_{y,P} &= \mathbf{I}^{\otimes n} \otimes \left(\mathbf{I} \otimes \mathbf{X}^{\otimes(n-1)}\right) \left(\mathbf{H}^{a_1} \otimes \mathbf{X}^{a_2} \otimes \mathbf{U}_n \left(-\frac{\pi}{2}\right)\right) \text{MCRZZ}_{n,a_1}^{1,\dots,n-1,a_2} \left(\frac{\tau}{\bar{\rho}l}\right) \left(\mathbf{H}^{a_1} \otimes \mathbf{X}^{a_2} \otimes \mathbf{U}_n \left(-\frac{\pi}{2}\right)^\dagger\right) \left(\mathbf{I} \otimes \mathbf{X}^{\otimes(n-1)}\right). \end{aligned}$$

Note the changed sign in the multi-controlled rotations compared to circuit in the main text.

Appendix D: Proof of Lemma 1

Consider the Hamiltonian Eq. 24 in the form given in Eqs. A2-A5. From [60, Proposition 9] and the triangle inequality the additive Trotter error between $\exp(-iH\tau)$

and $V(\tau)$ can be bounded by

$$\begin{aligned} &\sum_{j=1}^n \sum_{j'=j+1}^n \left\| \left[H_{x,j}^{\text{diag}}, H_{x,j'}^{\text{diag}} \right] \right\| + \sum_{j=1}^n \sum_{j'=j+1}^n \left\| \left[H_{x,j}^{\text{off}}, H_{x,j'}^{\text{off}} \right] \right\| \\ &+ \sum_{j=1}^n \sum_{j'=j+1}^n \left\| \left[H_{y,j}^{\text{off}}, H_{y,j'}^{\text{off}} \right] \right\| + \sum_{j=1}^n \sum_{j'=1}^n \left\| \left[H_{x,j}^{\text{diag}}, H_{x,j'}^{\text{off}} \right] \right\| \\ &+ \sum_{j=1}^n \sum_{j'=1}^n \left\| \left[H_{x,j}^{\text{diag}}, H_{y,j'}^{\text{off}} \right] \right\| + \sum_{j=1}^n \sum_{j'=1}^n \left\| \left[H_{x,j}^{\text{off}}, H_{y,j'}^{\text{off}} \right] \right\|, \end{aligned} \quad (\text{D1})$$

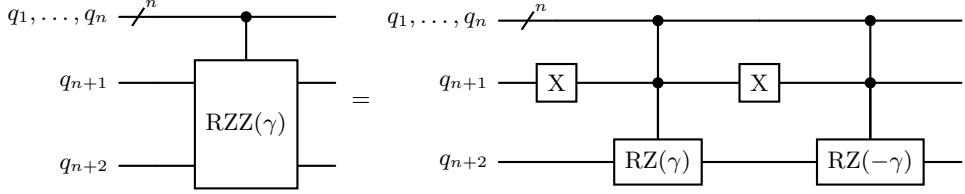


Figure 13: A decomposition of $\text{MCRZZ}_{n+1,n+2}^{1,\dots,n}$ gate into two $\text{MCRZ}_{n+2}^{1,\dots,n+1}$ gates.

where the terms with subindex j or j' correspond to elements under the sum in Eq. A3-A5.

In Refs. [34, 61] it was shown that two terms Eq. A1 with indices $j' > j$ do not commute with each other only when $j = 1$; in this case the norm of their commutator is 1. From this we can instantly give estimates for first three sums in Eq. D1:

$$\begin{aligned} \sum_{j=1}^n \sum_{j'=j+1}^n \left\| [H_{x,j}^{\text{diag}}, H_{x,j'}^{\text{diag}}] \right\| &= \left(\frac{\bar{u}}{2l} \right)^2 (n-1), \\ \sum_{j=1}^n \sum_{j'=j+1}^n \left\| [H_{x,j}^{\text{off}}, H_{x,j'}^{\text{off}}] \right\| &= \left(\frac{1}{2l\bar{\rho}} \right)^2 (n-1), \\ \sum_{j=1}^n \sum_{j'=j+1}^n \left\| [H_{y,j}^{\text{off}}, H_{y,j'}^{\text{off}}] \right\| &= \left(\frac{1}{2l\bar{\rho}} \right)^2 (n-1). \end{aligned} \quad (\text{D2})$$

Let us calculate the fourth term in Eq. D1. Using the explicit form of the Hamiltonian $H_{x,j}^{\text{diag}}$ and $H_{x,j'}^{\text{off}}$ the commutator between them takes the form

$$\begin{aligned} [H_{x,j}^{\text{diag}}, H_{x,j'}^{\text{off}}] &= -\frac{\bar{u}}{\bar{\rho}} \left(\frac{1}{2l} \right)^2 \sigma_{00}^{a_1} \otimes X^{a_2} \\ &\otimes \left[I^{\otimes(n-j)} \otimes (\sigma_{01} \otimes \sigma_{10}^{\otimes(j-1)} - \sigma_{10} \otimes \sigma_{01}^{\otimes(j-1)}) , I^{\otimes(n-j')} \otimes (\sigma_{01} \otimes \sigma_{10}^{\otimes(j'-1)} - \sigma_{10} \otimes \sigma_{01}^{\otimes(j'-1)}) \right] \otimes I^{\otimes n}. \end{aligned} \quad (\text{D3})$$

Upon a closer inspection we recognize that this commutator has precisely the same structure as the ones considered before, therefore its norm can be evaluated to be

$$\sum_{j=1}^n \sum_{j'=1}^n \left\| [H_{x,j}^{\text{diag}}, H_{x,j'}^{\text{off}}] \right\| = \frac{\bar{u}}{\bar{\rho}} \frac{2}{(2l)^2} (n-1). \quad (\text{D4})$$

It is easy to verify that the commutator between the $H_{x,j}^{\text{diag}}$ and $H_{y,j}^{\text{off}}$ operators is equal to zero. This is be-

cause they define the evolution on disjoint sets of qubits. Therefore, we obtain

$$\sum_{j=1}^n \sum_{j'=1}^n \left\| [H_{x,j}^{\text{diag}}, H_{y,j'}^{\text{off}}] \right\| = 0. \quad (\text{D5})$$

Finally, let us compute the commutator between the $H_{x,j}^{\text{off}}$ and $H_{y,j}^{\text{off}}$ operators:

$$\begin{aligned} [H_{x,j}^{\text{off}}, H_{y,j'}^{\text{off}}] &= -\frac{1}{(2l\bar{\rho})^2} [\sigma_{00}^{a_1} \otimes X^{a_2}, X^{a_1} \otimes \sigma_{00}^{a_2}] \\ &\otimes I^{\otimes(n-j)} \otimes (\sigma_{01} \otimes \sigma_{10}^{\otimes(j-1)} - \sigma_{10} \otimes \sigma_{01}^{\otimes(j-1)}) \otimes I^{\otimes(n-j')} \otimes (\sigma_{01} \otimes \sigma_{10}^{\otimes(j'-1)} - \sigma_{10} \otimes \sigma_{01}^{\otimes(j'-1)}) \\ &= -\frac{1}{(2l\bar{\rho})^2} (\sigma_{01}^{a_1} \otimes \sigma_{10}^{a_2} - \sigma_{10}^{a_1} \otimes \sigma_{01}^{a_2}) \\ &\otimes I^{\otimes(n-j)} \otimes (\sigma_{01} \otimes \sigma_{10}^{\otimes(j-1)} - \sigma_{10} \otimes \sigma_{01}^{\otimes(j-1)}) \otimes I^{\otimes(n-j')} \otimes (\sigma_{01} \otimes \sigma_{10}^{\otimes(j'-1)} - \sigma_{10} \otimes \sigma_{01}^{\otimes(j'-1)}). \end{aligned} \quad (\text{D6})$$

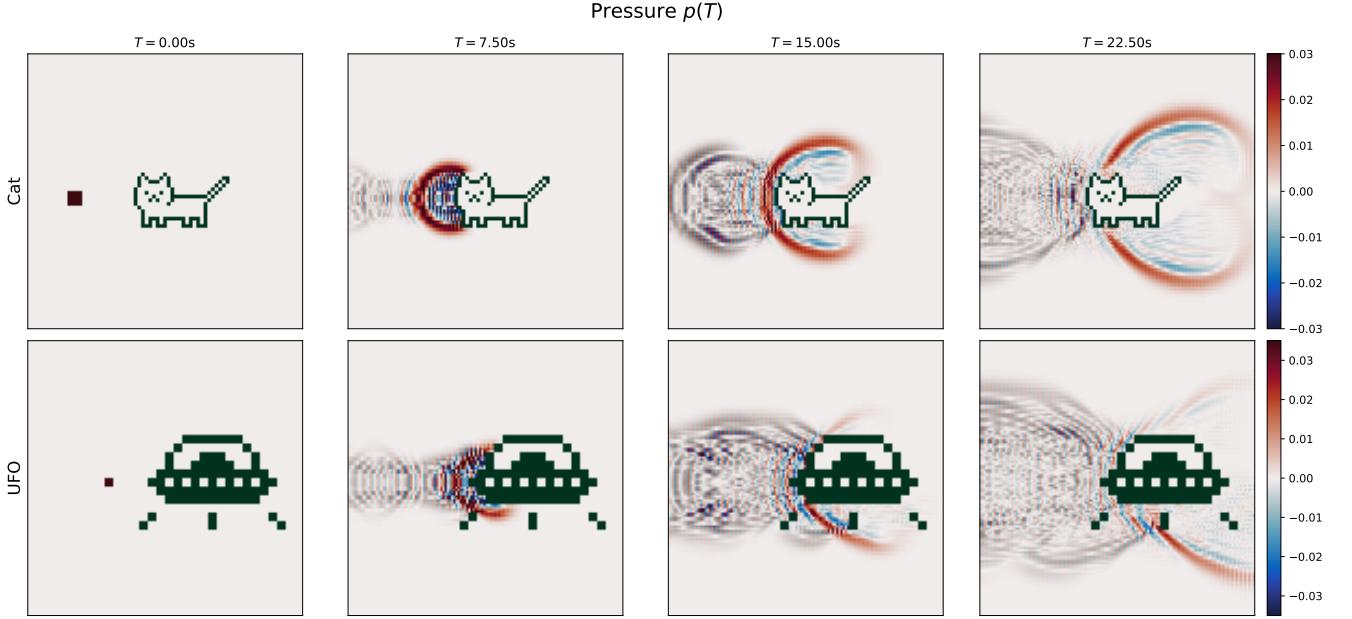


Figure 14: Quantum simulation of an acoustic wave propagating on top of a constant background flow and scattering from a complex shaped obstacle, obtained by executing quantum circuits derived using our construction. Here we plot the pressure (difference) at four distinct time points for the cat- (**top row**) and UFO-shaped obstacles (**bottom row**). We used $n = 8$ qubits, mean flow velocity $\bar{u} = 2$ and a time step of $\tau = 0.05$.

The spectral norm of this commutator is equal to $1/(2l\bar{\rho})^2$, therefore we have:

$$\begin{aligned} \sum_{j=1}^n \sum_{j'=1}^n \left\| [H_{x,j}^{off}, H_{y,j'}^{off}] \right\| &\leq \frac{1}{(2l\bar{\rho})^2} \sum_{j=1}^n \sum_{j'=1}^1 1 \\ &= \frac{1}{(2l\bar{\rho})^2} n^2. \end{aligned} \quad (\text{D7})$$

Summing everything up we obtain Eq. 29, which finishes the proof. \square

Appendix E: Improved obstacle circuit implementation on a QPU

In Sec. III we described the general algorithm for implementing arbitrary-shaped obstacles in the flow. For executing simulation on near- to mid-term QPUs, however, even constant factor improvements can make a crucial difference. Here we describe several such optimizations, which reduce the circuit depth and other sources of error (*e.g.* the Trotter error).

First, we show that obstacles can be implemented without increasing the overall Trotter error, which is not immediately apparent from the form of the generator of evolution with obstacles in Eq. 20. To this end recall that a rectangular binary cell obstacle with Dirichlet boundary conditions was implemented in Sec. III by canceling the action of the finite difference operator on its boundaries. The difference operator D^\pm was given in Eq. 12 as

a product of factors which we here denote by $W_j(\gamma)$:

$$W_j(\gamma) = U_j(-\pi/2) MCRZ_j^{1,\dots,j-1} \left(\frac{\gamma}{l} \right) U_j(-\pi/2)^\dagger. \quad (\text{E1})$$

As mentioned in Sec. III each $W_j(\gamma)$ implements 2^{n-j} disjoint sets of ± 1 elements on the super- and sub-diagonals in the explicit matrix representation of D^\pm . Thus, cancellation of these ± 1 pairs by the obstacle terms only affects the corresponding W_j operator. Furthermore, we have seen that the obstacle cancellation factor $e^{\sigma_{\hat{n}}\gamma}$ in Eq. 21 is also equal to W_j with $j = \hat{n}$, up to the controls on both registers which depend on the location of the obstacle, *i.e.* that $\sigma_{\hat{n}}$ is the generator of $W_{\hat{n}}$.

Note now that if arbitrary generators A and B commute, then so do their controlled versions $I \otimes A$, $|1\rangle\langle 1| \otimes B$ and $|0\rangle\langle 0| \otimes B$. This is also true of their multi-controlled versions, as easily shown by the mathematical induction. Taking advantage of the above observation, we can group all the generators of the D^\pm evolution factors W_j in Eq. 11 with the obstacle terms sharing the same value of $\hat{n} = j$ (a general obstacle can be composed of a multitude of binary cells of different sizes). Since for each j we now have the generator of W_j and (possibly) its multi-controlled version from the obstacle, which commute, the Trotter implementation of their sum is *exact*. The only Trotter error comes from non-commuting terms for different j (see App. D) and does not increase, since putting some of the matrix elements to zero does not increase its spectral norm. This is to be contrasted with direct Trot-

terization of the evolution generated by Eq. 20, where there would be additional error from the un-rearranged obstacle terms.

The above technical manipulation has a very clear physical interpretation: the exact implementation of the obstacle in the Trotter procedure ensures that the boundaries of the obstacle remain *impenetrable*, which would not hold in the naive Trotterization of Eq. 20.

In addition to the Trotter error improvement, a number of circuit simplifications are possible. First, note that in the construction of the D^\pm operator with obstacles there is no need to implement the hermitian conjugate of $e^{\sigma_{\hat{n}} \gamma}$ as an inverse circuit (see Fig. 4). Instead it suffices to change the sign in the RZ rotation in Eq. E1 or Eq. 21.

Next, observe that the implementation of the obstacle circuit in Fig. 4 contains multi-controlled versions of the multi-target operator $e^{\sigma_{\hat{n}} \gamma} \equiv W_{\hat{n}}$. The naive implementation of it would put the multi-controls on all of the factors in the definition in Eq. E1, resulting in a very complex circuit (after transpilation to a local gate basis). Instead we note that the effect of putting the controls on the $U_j(-\pi/2)$ operator and its inverse is redundant, and exactly the same action is obtained by only putting the controls on the MCRZ gate. Since this is in itself a multi-controlled gate, we obtain an MCRZ gate with a larger control set. The total amount of controls may not exceed the register size and remains $\mathcal{O}(n)$, and therefore the total circuit complexity remains quadratic (see the proof of Lemma 2).

Recalling that in the re-arrangement minimizing the Trotter error all terms of the same size j are grouped together, we observe that the “inner” basis change operators $U_j(\pi/2)$ coming from the original generator of D^\pm and from the obstacle cancel exactly (since the latter can also be made un-controlled, as discussed above). There is thus a single pair of $U_j(\pi/2)$, $U_j(\pi/2)^\dagger$ per each j group left, further simplifying the circuit.

Finally, we note that for a more complex obstacle shape which decomposes into multiple binary cells of different sizes, potentially adjacent to each other, there is no need to include cancellation terms for the boundaries they share.

Appendix F: Additional simulations with complex obstacles

To illustrate the applicability of our circuit construction to general obstacle types we supplement the experiments in Section VI with additional demonstration, with more complex – if not entirely realistic – shapes, the outlines of which form a cat and a UFO. The results of executing the circuits for the acoustic wave propagation in a background flow obtained using our algorithm are shown in Fig. 14. Here we used $c = \bar{\rho} = 1$, $\bar{u} = 2$, $l = 0.5$, $\tau = 0.05$ and employed $n = 8$ qubits.

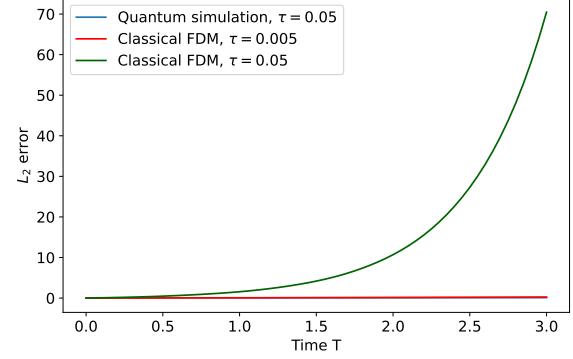


Figure 15: The L_2 -error of the quantum and classical FDM solutions for $p(T)$ in Fig. 8 with respect to the exact matrix exponential solution. When using the same time increment of $\tau = 0.05$ as the quantum solution the classical method becomes numerically unstable. To ensure a comparable (but still larger) error the classical FDM required $\tau = 0.005$ (see Fig. 9)

Appendix G: Extending quantum simulations to non-conservative regimes

Here we give a brief overview of how simulation of PDEs, in particular of LEE Eq. 22, can be extended to the case when then underlying numerical simulation can not be written as the Schrödinger equation directly. For the LEE this happens when $c \neq 1/\bar{\rho}$ or different boundary conditions, e.g. von Neumann, are used.

Let A be the matrix, which represents the numerical scheme of the simulation on a grid with already substituted differential operators and boundary conditions. We can split it in its real and imaginary parts by:

$$A_1 = \frac{A + A^\dagger}{2}, \quad A_2 = \frac{A - A^\dagger}{2i}, \quad (\text{G1})$$

where both matrices are Hermitian ($A_{1,2}^\dagger = A_{1,2}$). By the Trotter formula we have:

$$e^{A\tau} = e^{A_1\tau + iA_2\tau} \approx e^{A_1\tau} e^{iA_2\tau}. \quad (\text{G2})$$

Thus simulation of the PDE on a quantum computer can be reduced to implementing alternating steps of imaginary and real time evolutions. While efficient implementation of such quantum circuits is an active area of research, multiple methods already exist. In particular, Refs. [41, 62] extend the PDE with additional phase space via Koopman-von Neumann approach, so that solution can be stored and implemented on a quantum computer. A similar idea of extending the solution space with additional dimension to implement $e^{A_1\tau}$ is the basis of the “Schrödingerization” method [36–38, 63, 64]. In [61, 65] this technique is applied to non-conservative heat and advection equations. More general imaginary time evolutions methods can also be used [39, 66, 67], in particular

methods based on linear combination of unitaries (LCU) are the most common [42, 68]. In [40] Schrödingerization and LCU methods are combined and generalized, using block-encoding for Hamiltonians.

Importantly, in all of these techniques the Hilbert space is extended, and unitary operators such as $e^{iA_1\tau}$ (notice the appearance of the imaginary number i) are used as the building blocks in the implementation of the required non-unitary $e^{A_1\tau}$. Typically the unitaries become target operators controlled by variables from the extended space using quantum matrix arithmetics [69]. These controlled versions are still implementable with polynomial resources, therefore the quantum advantage in simulating the exponentially large grid outlined in Sec-

tion V remains. Thus, our results on efficient implementation of the unitary operators are also directly relevant to the more general non-unitary case.

For LEE the matrix A_2 is structurally the same as in Eq. 23 but with off-diagonal coefficients equal to $(1/\bar{\rho} + \bar{\rho}c^2)/2i$. A_1 is also structurally similar, except with $(1/\bar{\rho} - \bar{\rho}c^2)/2$ off-diagonal coefficients and nullified diagonal part. Consequently, $e^{iA_1\tau}$ can be implemented following the prescription described in the main text with removing the gates corresponding to the now missing diagonal part. Note that in the conservative regime ($c = 1/\bar{\rho}$) we have $A_1 = 0$ and $e^{iA_2\tau} = e^{-iH\tau}$ for H defined in Eq. 24.