# Quantum algorithm for partial differential equations with spatially varying parameters

**Technical Test at QunaSys Inc.**

Mingshuo Zhu (Institute of Science Tokyo)

2407.05019 by Yuki Sato, Hiroyuki Tezuka, Ruho Kondo and Naoki Yamamoto

August 2, 2025

# Contents

# Background and Introduction

- PDEs underpin many areas of physics and engineering (heat flow, fluid dynamics, acoustics, electromagnetism).
- Large grids with spatially varying parameters make classical simulations expensive; CAE demands high resolution and speed.
- Quantum simulation has been explored for conservative systems; we seek a method for non-conservative PDEs with spatial variation.
- A quantum algorithm based on linear combination of Hamiltonian simulation(LCHS) is applied to solve the PDEs with reduced costs.

# Contents

# From ODEs to LCHS

Let us start from an $N$-dimensional ordinary differential equation (ODE)

$$\frac{\mathrm{d}\boldsymbol{w}(t)}{\mathrm{d}t} = -\boldsymbol{A}(t)\boldsymbol{w}(t) + \boldsymbol{b}(t).$$

If $\boldsymbol{b} = 0$, the solution $\boldsymbol{w}(t) = e^{-\boldsymbol{A}t}\boldsymbol{w}(0)$. LCHS provides a general solution with a continuous superposition of unitary evolutions:

$$\begin{aligned}\boldsymbol{w}(t) = & \int_{\mathbb{R}} \frac{1}{\pi\left(1+k^2\right)} \mathscr{T} \exp\left(-i\int_0^t (\boldsymbol{H}(\tau) + k\boldsymbol{L}(\tau))\mathrm{d}\tau\right) \boldsymbol{w}(0)\mathrm{d}k \\ & + \int_0^t \int_{\mathbb{R}} \frac{1}{\pi\left(1+k^2\right)} \mathscr{T} \exp\left(-i\int_\tau^t \left(\boldsymbol{H}\left(\tau'\right) + k\boldsymbol{L}\left(\tau'\right)\right)\mathrm{d}\tau'\right) \boldsymbol{b}(\tau)\mathrm{d}k\,\mathrm{d}\tau\end{aligned}$$

where $\boldsymbol{L} = \frac{\boldsymbol{A}+\boldsymbol{A}^\dagger}{2}$ and $\boldsymbol{H} = \frac{\boldsymbol{A}-\boldsymbol{A}^\dagger}{2\mathrm{i}}$ are Hermitian. Approximating the integral by a finite sum yields a linear combination.

# From ODEs to LCHS

The two Hermitian operators are given by

$$\boldsymbol{L} = \frac{\boldsymbol{A} + \boldsymbol{A}^\dagger}{2}, \quad \boldsymbol{H} = \frac{\boldsymbol{A} - \boldsymbol{A}^\dagger}{2\mathrm{i}}.$$

Then, the integral can be by replaced by

$$\boldsymbol{w}(t) \approx \sum_{a=0}^{M-1} c_a \exp\left(-i\left(\boldsymbol{H} + k_a \boldsymbol{L}\right)t\right)\boldsymbol{w}(0).$$

$k_a$ is the discretization of $k$ with $M$ the number of integration points, and $c_a := \omega_a/\pi\left(1 + k_a^2\right)$ where $\omega_a$ is the weight for numerical integration.

Now let us apply quantum simulation to obatin the solutions $\boldsymbol{w}(t)$.

# Quantum procedure for LCHS

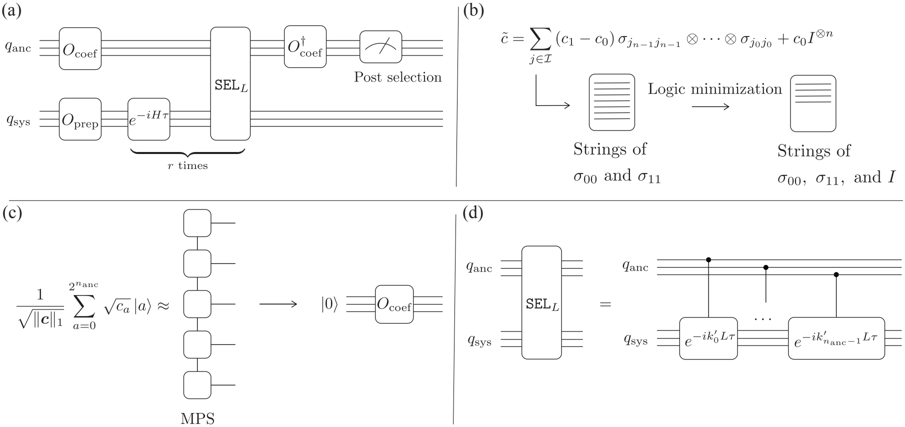
The discrete LCHS circuit can be decomposed into several oracles:

- Initialise $n$ system qubits to embed the solution $\boldsymbol{w}(t)$, and $n_{\mathrm{anc}} = \lceil \log_2 M \rceil$ ancilla qubits to embed integration points.
- Prepare the initial state $|w(0)\rangle$ (by $O_{\mathrm{prep}}$) on the system register.

$$|0\rangle^{\otimes n_{\mathrm{anc}}} \otimes |0\rangle^{\otimes n} \to |0\rangle^{\otimes n_{\mathrm{anc}}} \otimes |w(0)\rangle.$$

- Create a superposition of integration weights $\{c_a\}$ on the ancilla register (the coefficient oracle $O_{\mathrm{coef}}$).
- Apply $r$ times of the Hamiltonian simulation oracle $O_H(\tau) = e^{-\mathrm{i}\boldsymbol{H}\tau}$ and the select oracle $\mathsf{SEL}_L(\tau) := \sum_{a=0}^{M-1} |a\rangle\langle a| \otimes e^{-ik_a \boldsymbol{L}\tau}$ to sum up the coefficient part.
- Unprepare the ancilla by $O_{\mathrm{coef}}^{\dagger}$ and measure (all zeros) ancilla states yields the evolved state.

# Quantum procedure for LCHS



# Contents

# Target Partial Differential Equations

There are two types of the PDEs in this research

$$\rho(\boldsymbol{x})\frac{\partial^2 u(t,\boldsymbol{x})}{\partial t^2}+\zeta(\boldsymbol{x})\frac{\partial u(t,\boldsymbol{x})}{\partial t}-\nabla\cdot\kappa(\boldsymbol{x})\nabla u(t,\boldsymbol{x})+\alpha(\boldsymbol{x})u(t,\boldsymbol{x})=0,$$

$$\frac{\partial u(t,\boldsymbol{x})}{\partial t}-\nabla\cdot\kappa(\boldsymbol{x})\nabla u(t,\boldsymbol{x})+\boldsymbol{\beta}(\boldsymbol{x})\cdot\nabla u(t,\boldsymbol{x})+\alpha(\boldsymbol{x})u(t,\boldsymbol{x})=0.$$

$u(t,\boldsymbol{x})$ is a scalar field, $\rho(\boldsymbol{x})>0, \zeta(\boldsymbol{x})\geq 0, \kappa(\boldsymbol{x})\geq 0$ (diffusion), $\boldsymbol{\beta}(\boldsymbol{x})$ (velocity), and $\alpha(\boldsymbol{x})\geq 0$ (absorption) are spatially varying parameters.

The boundary condition is

$$\begin{cases} \boldsymbol{n}(\boldsymbol{x})\cdot\kappa(\boldsymbol{x})\nabla u(t,\boldsymbol{x})=0 & \text{for } (t,\boldsymbol{x})\in(0,T]\times\Gamma_{\mathrm{N}} \\ u(t,\boldsymbol{x})=0 & \text{for } (t,\boldsymbol{x})\in(0,T]\times\Gamma_{\mathrm{D}} \\ u(t,\boldsymbol{x})=u(t,0) & \text{for } (t,\boldsymbol{x})\in(0,T]\times\Gamma_{\mathrm{P}} \end{cases}$$

# Finite difference discretization (2402.18398)

For 1D domain $\Omega \equiv (0, L)$, it can be discretized uniformly into $N = 2^n$ points with interval $h \equiv L/(N+1)$.

$\boldsymbol{u}$ on domain $\Omega$ can be discretized into $N$ points $\boldsymbol{u} \equiv [u_0, u_1, \ldots, u_{N-1}]$.

The forward and backward spatial difference operators $D^{\pm}$ becomes

$$\left(D^{+}\boldsymbol{u}\right)_j = \frac{u_{j+1} - u_j}{h}, \quad \left(D^{-}\boldsymbol{u}\right)_j = \frac{u_j - u_{j-1}}{h} \quad \text{for } j = 0, 1, \ldots, N-1,$$

where $u_N(u_{-1})$ is determined from the boundary condition:

$$u_N := \begin{cases} 0 & \text{for Dirichlet BC} \\ u_{N-1} & \text{for Neumann BC} \\ u_0 & \text{for periodic BC} \end{cases} \qquad u_{-1} := \begin{cases} 0 & \text{for Dirichlet BC} \\ u_0 & \text{for Neumann BC} \\ u_{N-1} & \text{for periodic BC} \end{cases}$$

Quantize the discretized field $\boldsymbol{u}$ as

$$|u\rangle := \sum_{j=0}^{2^n-1} u_j |j\rangle,$$

where $|j\rangle := |j_{n-1} j_{n-2} \dots j_0\rangle$ with $j_{n-1}, j_{n-2}, \dots, j_0 \in \{0, 1\}$, and $\|\boldsymbol{u}\|_2 = 1$.

Then the difference operators above can be represented as matrix product operators (MPOs) among the Identity $I$ and
The Ladder Operators

$$\sigma_{01} := \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \sigma_{10} := \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix},$$

The Projector Operators

$$\sigma_{00} := \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad \sigma_{11} := \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

The point of quantization is the MPO representation of the shift operators

$$S^{-} = \sum_{j=1}^{2^{\prime\prime}-1} |j-1\rangle\langle j| = \sum_{j=1}^{n} I^{\otimes(n-j)} \otimes \sigma_{01} \otimes \sigma_{10}^{\otimes(j-1)},$$

$$S^{+} = \sum_{j=1}^{2^{n}-1} |j\rangle\langle j-1| = \sum_{j=1}^{n} I^{\otimes(n-j)} \otimes \sigma_{10} \otimes \sigma_{01}^{\otimes(j-1)}$$

Then the quantum difference operators are given by

$$\begin{cases} D_{\mathrm{D}}^{-} = \frac{1}{h}\left(I^{\otimes n} - S^{+}\right), \\ D_{\mathrm{N}}^{-} = \frac{1}{h}\left(I^{\otimes n} - S^{+} - \sigma_{00}^{\otimes n}\right), \\ D_{\mathrm{P}}^{-} = \frac{1}{h}\left(I^{\otimes n} - S^{+} - \sigma_{01}^{\otimes n}\right) \end{cases} \qquad \begin{cases} D_{\mathrm{D}}^{+} = \frac{1}{h}\left(S^{-} - I^{\otimes n}\right), \\ D_{\mathrm{N}}^{+} = \frac{1}{h}\left(S^{-} - I^{\otimes n} + \sigma_{11}^{\otimes n}\right), \\ D_{\mathrm{P}}^{+} := \frac{1}{h}\left(S^{-} - I^{\otimes n} + \sigma_{10}^{\otimes n}\right) \end{cases}$$

For multi-dimensional case like $(x, y)$

$$\left(D_{\mathrm{B}}^{\mu}\right)_{\alpha} = I^{\otimes(\alpha-1)n} \otimes D_{\mathrm{B}}^{\mu} \otimes I^{\otimes(d-\alpha)n},$$

The first type of partial differential equation can be represented by matrix $\boldsymbol{A}$:

$$\frac{\partial}{\partial t}\begin{pmatrix}\sqrt{\rho}\dot{u}\\ \sqrt{\kappa}\nabla u\\ \sqrt{\alpha}u\end{pmatrix} = -\begin{pmatrix}\frac{\zeta(\boldsymbol{x})}{\rho(\boldsymbol{x})} & -\frac{1}{\sqrt{\rho(\boldsymbol{x})}}\nabla^\top\sqrt{\kappa(\boldsymbol{x})} & \sqrt{\frac{\alpha(\boldsymbol{x})}{\rho(\boldsymbol{x})}}\\ -\sqrt{\kappa(\boldsymbol{x})}\nabla\frac{1}{\sqrt{\rho(\boldsymbol{x})}} & 0 & 0\\ -\sqrt{\frac{\alpha(\boldsymbol{x})}{\rho(\boldsymbol{x})}} & 0 & 0\end{pmatrix}\begin{pmatrix}\sqrt{\rho}\dot{u}\\ \sqrt{\kappa}\nabla u\\ \sqrt{\alpha}u\end{pmatrix},$$

It leads to a $(d+2)\times 2^n$ dimensional $w_{\mu,j}(t,\boldsymbol{x}^{[j]})$ after

- The domain $\Omega\subset\mathbb{R}^d$ is discretized into $2^n$ nodes
- The position of the $j$-th node is denoted as $\boldsymbol{x}^{[j]}$
- The number of nodes along with the $x_\mu$-axis is $2^{n_\mu}$ and $n=\sum_{\mu=0}^{d-1}n_\mu$

$$\boldsymbol{w}(t)=\sum_{j=0}^{2^n-1}\Big(\underbrace{w_{0,j}|0\rangle|j\rangle}_{\sqrt{\rho}\dot{u}}+\underbrace{\sum_{\mu=0}^{d-1}w_{\mu+1,j}|\mu+1\rangle|j\rangle}_{\sqrt{\kappa}\nabla u}+\underbrace{w_{d+1,j}|d+1\rangle|j\rangle}_{\sqrt{\alpha}u}\Big).$$

The red spatial varying parameter should also be discretized. How?

Discrete the spatial varying parameter $c(\boldsymbol{x})$ into diagonal operator $\tilde{c}$

$$\tilde{c} = \sum_{j=0}^{2^n-1} c\left(\boldsymbol{x}^{[j]}\right) |j\rangle\langle j|$$

Then the mutiplication with a scalar field $\boldsymbol{u} := \sum_{j=0}^{2^n-1} u\left(\boldsymbol{x}^{[j]}\right) |j\rangle$ can be represented as

$$\tilde{c}\boldsymbol{u} = \sum_{j=0}^{2^n-1} c\left(\boldsymbol{x}^{[j]}\right) u\left(\boldsymbol{x}^{[j]}\right) |j\rangle,$$

which represents the product $c(\boldsymbol{x})u(\boldsymbol{x})$. Combine the difference operators defined before, we can give the discrete matrix $\boldsymbol{A}$!

$$\begin{aligned}\boldsymbol{A} :=& |0\rangle\langle 0| \otimes \tilde{\rho}^{-1}\tilde{\zeta} - \sum_{\mu=0}^{d-1} |0\rangle\langle \mu+1| \otimes \tilde{\rho}^{-\frac{1}{2}} D_\mu^+ \tilde{\kappa}^{\frac{1}{2}} + |0\rangle\langle d+1| \otimes \tilde{\rho}^{-\frac{1}{2}}\tilde{\alpha}^{\frac{1}{2}} \\ & - \sum_{\mu=0}^{d-1} |\mu+1\rangle\langle 0| \otimes \tilde{\kappa}^{\frac{1}{2}} D_\mu^- \tilde{\rho}^{-\frac{1}{2}} - |d+1\rangle\langle 0| \otimes \tilde{\rho}^{-\frac{1}{2}}\tilde{\alpha}^{\frac{1}{2}} + \text{b.c. corrections}\end{aligned}$$

# Logic minimization

If the $c(\boldsymbol{x})$ takes the value either $c_0$ or $c_1$. The operator $\tilde{c}$ will be simplified. Let $\mathscr{I}$ denote the index set where $c\left(\boldsymbol{x}^{[j]}\right) = c_1$, $\forall j \in \mathscr{I}$. Then we can represent $\tilde{c}$ as

$$\begin{aligned}\tilde{c} &= \sum_{j\in\mathscr{I}} (c_1 - c_0)\,|j\rangle\langle j| + c_0 I^{\otimes n} \\ &= \sum_{j\in\mathscr{I}} (c_1 - c_0)\,\sigma_{j_{n-1}j_{n-1}} \otimes \cdots \otimes \sigma_{j_0 j_0} + c_0 I^{\otimes n}\end{aligned}$$

where we write $j = (j_{n-1}\dots j_0)_2$ and use the projector operator $\sigma_{j_i,j_i}$.

The number of terms by utilizing the relationship can be reduced by

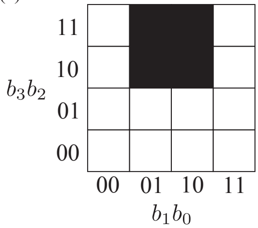
$$\sigma_{jj} \otimes \sigma_{00} + \sigma_{jj} \otimes \sigma_{11} = \sigma_{jj} \otimes (\sigma_{00} + \sigma_{11}) = \sigma_{jj} \otimes I.$$

corresponding to the transformation of Boolean function
$f = ab + a\bar{b} = a(b + \bar{b}) = a,$

# Logic minimization

For simplicity, let us first set $c_1 = 1$ and $c_0 = 0$ and consider a $2^2 \times 2^2$ grid which corresponds 4 digits $b_3b_2b_1b_0$ with $b_3b_2$ the $y$-axis and $b_1b_0$ the $x$-axis.

(a)




(b)

| $b_3b_2b_1b_0$ | $f(b_3,b_2,b_1,b_0)$ |
|---|---|
| 1101 | 1 |
| 1001 | 1 |
| 1110 | 1 |
| 1010 | 1 |

(c)

$$\begin{aligned}\tilde{c} = \sigma_{11} \otimes \sigma_{11} \otimes \sigma_{00} \otimes \sigma_{11} \\ + \sigma_{11} \otimes \sigma_{00} \otimes \sigma_{00} \otimes \sigma_{11} \\ + \sigma_{11} \otimes \sigma_{11} \otimes \sigma_{11} \otimes \sigma_{00} \\ + \sigma_{11} \otimes \sigma_{00} \otimes \sigma_{11} \otimes \sigma_{00}\end{aligned}$$

(d)

| $b_3b_2b_1b_0$ | $f(b_3,b_2,b_1,b_0)$ |
|---|---|
| 1-01 | 1 |
| 1-10 | 1 |

(e)

$$\begin{aligned}\tilde{c} = \sigma_{11} \otimes I \otimes \sigma_{00} \otimes \sigma_{11} \\ + \sigma_{11} \otimes I \otimes \sigma_{11} \otimes \sigma_{00}\end{aligned}$$

There may exist duplicate coordinate, like 1-01 and 10-1 both give 1001. To resolve it, the search and resolve algorithm is necessary! (1-01 $\rightarrow$ 1101)

# Matrix Product State to $k_a$

Finally, we need discrete the coefficient $k_a$ appearing in the LCHS

$$\sum_{a=0}^{M-1} c_a \exp\left(-i\left(\boldsymbol{H}+k_a\boldsymbol{L}\right)t\right), \quad c_a := \omega_a/\pi\left(1+k_a^2\right)$$

Rather than loading the full vector explicitly (exponential cost), we choose a matrix product state (MPS) representation because

- it approximates $\{\sqrt{c_a}\}$ using a small number of parameters (polynomial cost).
- it is mathematically equal to the tensor network which can be done by known package (scikit-tt).

Let us set $a = \left(a_{n_{\text{anc}}-1}\dots a_0\right)_2$, interval $\omega_a = 2^{-n_{\text{frac}}}$, and

$$k_{a=\left(a_{n_{\text{anc}}-1}\dots a_0\right)_2} := \left(-a_{n_{\text{anc}}-1}2^{n_{\text{anc}}-1} + \sum_{m=0}^{n_{\text{anc}}-2} a_m 2^m\right) 2^{-n_{\text{frac}}}.$$

Our ultimate goal is to build $\left(1/\sqrt{\|c\|_1}\right)\sum_{a=0}^{2^{n\,\mathrm{anc}}-1}\sqrt{c_a}|a\rangle$ with MPS method. First, it is not difficult to write

$$\sum_{a=0}^{2^{n_{\mathrm{anc}}}-1} k_a\,|a\rangle = \sum_{a=0}^{2^{n_{\mathrm{anc}}}-1}\sum_{b_{n_{\mathrm{anc}}}=1}^{n_{\mathrm{anc}}}\cdots\sum_{b_1=1}^{n_{\mathrm{anc}}} K^{[n_{\mathrm{anc}}]}_{a_{n_{\mathrm{anc}}}b_{n_{\mathrm{anc}}}} K^{[n_{\mathrm{anc}}-1]}_{b_{n_{\mathrm{anc}}}a_{n_{\mathrm{anc}}-1}b_{n_{\mathrm{anc}}-1}}\cdots$$

$$\cdots K^{[2]}_{b_3a_2b_2}K^{[1]}_{b_2a_1}\,|a_{n_{\mathrm{anc}}}a_{n_{\mathrm{anc}}-1}\ldots a_1\rangle$$

where $b_i$ are virtual sites with bond dimensional $n_{\mathrm{anc}}$ and the tensors are

$$K^{[n_{\mathrm{anc}}]}_{a_{n_{\mathrm{anc}}}b_{\mathrm{anc}}} = \begin{cases} -a_{n_{\mathrm{anc}}}2^{n_{\mathrm{anc}}-1-n_{\mathrm{frac}}} & \text{if } b_{n_{\mathrm{anc}}} = n_{\mathrm{anc}} \\ 1 & \text{otherwise} \end{cases}$$

$$K^{[m]}_{b_{m+1}a_mb_m} = \begin{cases} a_m 2^{m-1-n_{\mathrm{frac}}} & \text{if } b_{m+1} = b_m = m \\ \delta_{b_{m+1}b_m} & \text{otherwise} \end{cases}$$

$$K^{[1]}_{b_2a_1} = \begin{cases} a_1 2^{-n_{\mathrm{frac}}} & \text{if } b_2 = 1 \\ 1 & \text{otherwise} \end{cases}.$$

Similarly, it is possible to construct tensor representation of $\sum_{a=0}^{2^{n_{\mathrm{anc}}}-1}(1+k_a^2)\,|a\rangle$

To represent the target $\sum_{a=0}^{2^{n_{\text{anc}}}-1}\sqrt{1+k_a^2}|a\rangle$, define the following tensor-valued equation with fixed bond dimension $r_{\Psi}$:

$$\mathscr{F}(\mathbf{\Psi}) := \mathrm{diag}(\mathbf{\Psi})\mathbf{\Psi} - \sum_{a=0}^{2^{n_{\text{anc}}}-1}\left(1+k_a^2\right)|a\rangle = 0,$$

and solved it recursively to find $\mathbf{\Psi}$ which is the numerical target!

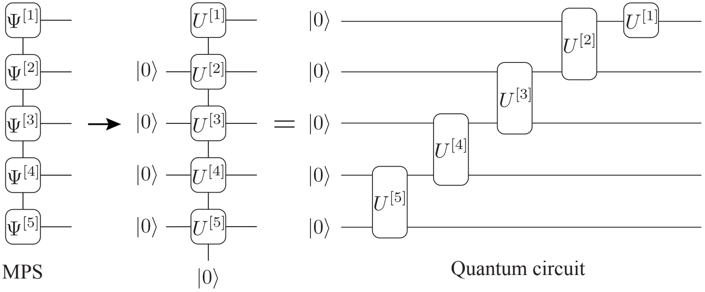
Finally, numerically solve the linear equation with the modified alternating least square

$$\mathrm{diag}(\mathbf{\Psi})\mathbf{\Phi} = \sqrt{\frac{2^{n_{\text{frac}}}}{\pi}2^{n_{\text{anc}}-1}}|a\rangle,$$

$\mathbf{\Phi}$ with bond dimension $r_{\Phi}$ is the approximation of state $\sum_{a=0}^{2^{n_{anc}}-1}\sqrt{c_a}|a\rangle$.

The final step is to transform $\mathbf{\Phi}$ into quantum circuit based on (1908.07958). The concrete procedure is not our main part so we skip it.

$$\frac{1}{\sqrt{\|\boldsymbol{c}\|_1}} \sum_{a=0}^{2^{n_{\mathrm{anc}}}} \sqrt{c_a}\, |a\rangle \approx$$




Finally, add the select oracle to sum $k_a \boldsymbol{L}$ to $\boldsymbol{H}$

$$\begin{aligned}
\mathrm{SEL}_L(\tau) &:= \sum_{a=0}^{2^{n_{\mathrm{anc}}}-1} |a\rangle\langle a| \otimes e^{-ik_a \boldsymbol{L}\tau} \\
&= \left( |0\rangle \langle 0|_{n_{\mathrm{anc}}-1} \otimes I^{\otimes n} + |1\rangle \langle 1|_{n_{\mathrm{anc}}-1} \otimes O_L\left(-2^{n_{\mathrm{anc}}-1-n_{\mathrm{frac}}}\tau\right)\right) \\
&\prod_{m=0}^{n_{\mathrm{anc}}-2} \left( |0\rangle \langle 0|_m \otimes I^{\otimes n} + |1\rangle \langle 1|_m \otimes O_L\left(2^{m-n_{\mathrm{frac}}}\tau\right)\right),
\end{aligned}$$

The full procedure combines the previous ingredients:

- Discretize space and load the field into $n$ qubits.
- Translate the PDE into a first-order ODE $\dot{\boldsymbol{w}} = -\boldsymbol{A}\boldsymbol{w}$.
- Split $\boldsymbol{A}$ into Hermitian and anti-Hermitian parts to define $\boldsymbol{L}$ and $\boldsymbol{H}$.
- Compress the Hamiltonian via logic minimization; approximate the integration weights with an MPS.
- Run the LCHS circuit including ($O_{\boldsymbol{H}}$, $O_{\text{coef}}$ and $O_{\boldsymbol{L}}$ ) and measure the ancillas to extract the solution.

The quantum simulation with Hamiltonian in terms of ladder operators is realized by the Bell basis in 2402.18398.

# Contents

# Acoustic equation with spatially varying sound speed

We test the algorithm on the $2D$ acoustic wave equation

$$\partial_t^2 p - c(\boldsymbol{x})^2 \nabla^2 p = 0,$$

with spatially varying sound speed $c(\boldsymbol{x})$. Its matrix form is

$$\frac{\partial}{\partial t}\begin{pmatrix} -\frac{1}{c}p \\ \rho_0 \boldsymbol{v} \end{pmatrix} = -\begin{pmatrix} 0 & -c\nabla^\top \\ -\nabla c & 0 \end{pmatrix}\begin{pmatrix} -\frac{1}{c}p \\ \rho_0 \boldsymbol{v} \end{pmatrix}$$

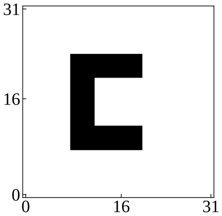
Here we impose we impose the Dirichlet conditions for $x$-axis and periodic boundary condition for $y$-axis. The discrete $\boldsymbol{A}$ can be decomposed into

$$\boldsymbol{L} = \frac{\boldsymbol{A} + \boldsymbol{A}^\dagger}{2} = 0, \quad \text{No LCH circuit is necessary!}$$

$$\boldsymbol{H} = \frac{\boldsymbol{A} - \boldsymbol{A}^\dagger}{2i} = i \sum_{\mu=0}^{d-1} \left( |0\rangle\langle \mu+1| \otimes \tilde{c} D_\mu^+ + |\mu+1\rangle\langle 0| \otimes D_\mu^- \tilde{c} \right).$$

There exist spatial varying operator $\tilde{c}$!

Set the discrete space as a $2^5 \times 2^5$ grid where $c(\boldsymbol{x}) = 10$ on the 128 nodes as




with $n_0 = n_1 = 5, \tau = 1.0 \times 10^{-3}, T = 20, h = 1$, and initial condition

$$
p\left(0, \boldsymbol{x}^{[j]}\right) = \begin{cases} -\frac{\sqrt{2}}{4} & \text{for } x_1^{[j]} \in \{14, 15, 16, 17\}, x_0^{[j]} \in \{14, 15\} \\ 0 & \text{otherwise} \end{cases},
$$

$$
\boldsymbol{v}\left(0, \boldsymbol{x}^{[j]}\right) = 0.
$$

or in discrete form

$$
|w(0)\rangle = \frac{\sqrt{2}}{4}|00\rangle \otimes (|01110\rangle + |01111\rangle + |10000\rangle + |10001\rangle) \otimes (|01110\rangle + |01111\rangle),
$$

The numerical solution on qiskit unitary simulator is as follows

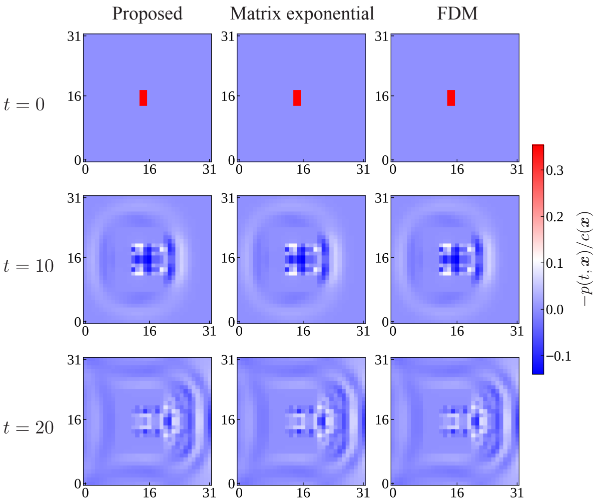



**Figure:** The first column represents the quantum simulation, the second one represents the matrix exponential evolution and the last one is classical Euler's method

# Efficient observables for quantum computer

The direct visualization above is impractical in a real quantum computation.

We have to set some observables to extract meaningful information from the quantum state prepared through the Hamiltonian simulation algorithm.

One simple observable is the squared acoustic pressure

$$\int_{\Omega} \chi_{\Omega_{\text{eval}}}(\boldsymbol{x}) p^2 \, \mathrm{d}\boldsymbol{x} \approx \boldsymbol{w}^{\dagger} (|0\rangle\langle 0| \otimes \tilde{c}\tilde{\chi}\tilde{c}) \boldsymbol{w}$$

with the step function $\chi_{\text{eval}} = 1$ in the domain $\Omega$. Since $|0\rangle\langle 0| \otimes \tilde{c}\tilde{\chi}\tilde{c}$ is completely diagonal, the expectation value can be efficiently estimated by measurement.

# Efficient observables for quantum computer

It is similar with the choice of Hamiltonian in quantum simulation. The diagonal ones always provide the efficient result. But it is possible to generalize

- The observables consists of sparse local Pauli matrices.

In PDEs, the Hermitian operators are decomposed into a summation of the tensor products of the projectors ($Z$) and ladder operators ($X$, $Y$):

$$\mathscr{O} = \sum_i c_i X^{\otimes n} \tilde{\chi}, \quad X \in \{I, U, D, R, L\}$$

- only polynomially many non-zero coefficients in this decomposition
- tensor products contains large number of projectors and identity
- non-zero observables only in some specific domain.

# Contents

# Conclusion and future work

- Presented a quantum algorithm for solving second-order linear PDEs of non-conservative systems with spatially varying parameters via LCHS.
- Key innovations include efficient spatial discretizations mapped to qubits, logic minimization for reducing Hamiltonian complexity, and a MPS-based coefficient oracle.
- Demonstrated the method on acoustic and heat equations, showing agreement with classical solutions.
- Future directions: extend to higher-dimensional systems and complex boundary conditions and optimize the MPS approximation.

# Thank you for watching.