

Gate construction of block-encoding for Hamiltonians needed for simulating partial differential equations

Nikita Guseynov ^{*}

University of Michigan-Shanghai Jiao Tong University Joint Institute, Shanghai 200240, China

Xiajie Huang [†]

School of Mathematical Sciences, Shanghai Jiao Tong University, Shanghai 200240, China

Nana Liu[‡]

Institute of Natural Sciences, School of Mathematical Sciences, Shanghai Jiao Tong University, Shanghai 200240; China Ministry of Education Key Laboratory in Scientific and Engineering Computing, Shanghai Jiao Tong University, Shanghai 200240 China; Shanghai Artificial Intelligence Laboratory, Shanghai, China; and University of Michigan-Shanghai Jiao Tong University Joint Institute, Shanghai 200240, China



(Received 2 June 2024; accepted 8 July 2025; published 29 July 2025)

One of the most promising applications of quantum computers is solving partial differential equations (PDEs). By using the Schrödingerization technique—which converts nonconservative PDEs into Schrödinger equations—the problem can be reduced to Hamiltonian simulations. The particular class of Hamiltonians we consider is shown to be sufficient for simulating almost any linear PDE. In particular, these Hamiltonians consist of discretizations of polynomial products and sums of position and momentum operators. This paper addresses an important gap by efficiently loading these Hamiltonians into the quantum computer through block-encoding. The construction is explicit and efficient in terms of one- and two-qubit operations, forming a fundamental building block for constructing the unitary evolution operator for that class of Hamiltonians. The proposed algorithm demonstrates a squared logarithmic scaling with respect to the spatial partitioning size, offering a polynomial speedup over classical finite-difference methods in the context of spatial partitioning for solving PDEs. Furthermore, the algorithm is extended to the multidimensional case, achieving an exponential acceleration with respect to the number of dimensions, alleviating the curse of dimensionality problem. This work provides an essential foundation for developing explicit and efficient quantum circuits for PDEs, Hamiltonian simulations, and ground state and thermal state preparation.

DOI: [10.1103/xlpd-fb1g](https://doi.org/10.1103/xlpd-fb1g)

I. INTRODUCTION

Quantum computation [1] is an emerging technology that has the potential to tackle certain computational problems intractable on classical devices. The most influential algorithms include Shor's factorization algorithm [2,3], Grover's search algorithm [4,5], simulation of quantum chemistry models [6], quantum algorithms based on random walks [7], and the quantum linear systems (HHL) algorithm [8]. Other important applications include quantum simulation for partial differential equations, which form the backbone of mathematical modeling for various physical phenomena and often

prove intractable for classical computers as the dimensionality of the problem and the size of discretization increase [9]. Quantum algorithms have the potential to improve scaling in both the dimension and the size of the discretization used. Thus, these algorithms have the potential to improve diverse areas in physics [10,11], engineering [12–15], and finance [16,17].

Despite significant progress in quantum simulation and advancements in applying these techniques to partial differential equations, few works demonstrate explicit gate construction. This step is crucial for treating quantum computation as a viable future tool and for implementation on fault-tolerant quantum computers. Many algorithms rely on oracles—black boxes that are simply assumed to exist, with no further inquiry. However, measuring efficiency solely by the number of oracle calls does not accurately reflect the total implementation cost, as the construction cost of these black boxes often remains unspecified. If the complexity of constructing these black boxes scales nonpolynomially with the number of qubits, any quantum advantage may be negated. This issue highlights a significant gap in the current literature.

^{*}Contact author: guseynov.nm@gmail.com

[†]Contact author: xj.huang@sjtu.edu.cn

[‡]Contact author: nana.liu@quantumlab.org

In this paper, our key contribution is providing such an efficient explicit gate construction for a block-encoding of a particular class of multimode Hamiltonians with the schematic form $\hat{H} = \sum \alpha \hat{x}^q \hat{p}^m + \alpha^* \hat{p}^m \hat{x}^q$ for positive integers q, m , where $\hat{x}$ and $\hat{p}$ are, respectively, position and momentum operators. The block-encoding of $\hat{H}$ can be used as a fundamental building block for near-optimal quantum simulation and provides a block-encoding of the evolution operator $\exp(-i\hat{H}t)$, with optimal scaling in time $\mathcal{O}(N^m t)$ [18]. Using the Schrödingerization technique [14, 19–21], we demonstrate that this class of Hamiltonians plays a pivotal role in simulating any linear partial differential equation with polynomial coefficient functions.

To benchmark our quantum method, we compare against a standard classical numerical scheme; specifically, the forward Euler method applied to the finite-difference discretization of the target partial differential equation, followed by solving the resulting system of linear equations at each time step. Relative to this baseline, our approach achieves a small polynomial improvement in terms of spatial partitioning, scaling as N^m instead of N^{m+1} , where m is the maximum degree of the spatial derivative. Furthermore, in the multidimensional case, we attain an exponential acceleration with respect to the number of dimensions d , alleviating the curse of dimensionality problem (see Sec. IV for details). It is also potentially important for quantum problems like the quantum simulation of $\hat{H}$, as well as ground state and thermal state preparation.

There are many other approaches for Hamiltonian simulations. For example, the quantum random walk combined with quantum signal processing has been recently developed to address sparse Hamiltonian simulations [22]; the algorithm demonstrated optimal scaling on all parameters. However, their direct application is hindered by the absence of a method to construct a digital quantum oracle query access to the Hamiltonian, which is a crucial component for constructing the quantum walk isometry operator. Similarly, another approach exploiting fractional and continuous-query models for efficient sparse Hamiltonian simulations [23] encounters the same obstacle, as it too requires digital quantum oracle query access to the Hamiltonian under consideration.

Others [15, 24] attempt to construct optimal circuits via decomposition of $\hat{H}$ into 2×2 ladder operators—an alternative to the well-established Pauli basis [25]. These methods are suited to toy models with separate terms for momentum and coordinate quantum operators, not their multiplication.

In this paper, we introduce a general framework for block-encoding construction, which accepts any Hamiltonian composed of superpositions of terms $\hat{x}^q \hat{p}^m$ and $\hat{p}^m \hat{x}^q$. Our proposed algorithm exhibits (polylogarithmic) $\mathcal{O}(n^2)$ scaling in the number of finite-difference partitionings 2^n for spatial dimensions and enjoys linear scaling with time $\mathcal{O}(t)$, as we construct the evolution operator for extensive periods.

Inspired by the explicitness of Shor’s factoring algorithm, which is the first to theoretically offer robust quantum speedup, we present explicit quantum circuits and estimate their complexity in terms of single-qubit operations and the simplest entangling operations (C-NOT) [25], with all the constants included. Furthermore, we provide a systematic guide showing step-by-step the necessary quantum circuits. One could think that the foundational role of the T gate

simplifies its usage in fault-tolerant quantum computing. However, even a simple one-qubit R_z gate can require a varying number of T gates, heavily dependent on the gate’s parameters, especially when using the Solovay-Kitaev algorithm [26]. Thus, we rely on C-NOT and one-qubit gates as fundamental components, given their consistency and simplicity in gate counting.

The organization of this paper is as follows. In Sec. II, we begin by providing the main results and reviewing how they are useful in Hamiltonian simulations. We also explain in more detail why we choose to construct the block-encoding of the $\hat{H}$ considered in this paper: explicit gate construction for the block-encoding for arbitrary $\hat{H}$ simulation is an open problem and too difficult for now. The areas of application in quantum problems and the simulation of partial differential equations are considered. The overall algorithm contains many relatively small steps. We provide a road map of our algorithm in Fig. 2 in Sec. II A, making it easier to follow. This can be consulted while reading the paper. The following sections then show definitions and step-by-step implementation of each step in the road map. In Sec. II B, we introduce essential mathematical objects. The main results are in Sec. III, which details the process of how to implement nearly optimal block-encoding for the single-mode $\hat{H}$. This culminates with the construction an efficient evolution operator for the Hamiltonian (Theorem 9), providing the exact number of required gates. We then dedicate Sec. IV to an important generalization to the multidimensional $\hat{H}$. Section V concludes the paper.

To clarify our exposition, we include Tables I–IV. The first and the third tables explain all the notation introduced and indicate their usage for one- and multidimensional cases, respectively. The second and fourth tables summarize all milestones from Fig. 2 and detail the complexity of each step in the implementation for both cases.

II. RESULTS, MOTIVATION, AND ROAD MAP

We begin with the following Hamiltonian as a sum of η terms:

$$H = \sum_{k=0}^{\eta-1} (\alpha_k P_{q_k}(x) p^{m_k} + \alpha_k^* p^{m_k} P_{q_k}(x));$$

$$\alpha_k \in \mathbb{C}; \quad q_k, m_k \in \mathbb{N}, \quad (1)$$

where x and p denote coordinate and momentum operators, respectively, and the symbol $*$ represents the complex conjugation operation; $P_{q_k}(x)$ is a degree- q_k polynomial satisfying that

- (1) $P_{q_k}(x)$ has parity- $(q_k \bmod 2)$, which means that it has only odd or even degrees of x ;
- (2) $P_{q_k}(x)$ is a real polynomial, which means that all the coefficients are real; and
- (3) for all $x \in [-1, 1]$: $|P_{q_k}| < 1$.

We note that this Hamiltonian is just a more sophisticated view of the already mentioned model

$$H = \sum_{k=0}^{\eta-1} (\alpha'_k x^{q_k} p^{m_k} + \alpha'^*_k p^{m_k} x^{q_k});$$

$$\alpha'_k \in \mathbb{C}; \quad q_k, m_k \in \mathbb{N}. \quad (2)$$

To simulate this Hamiltonian numerically (on both classical and quantum devices), finite-difference schemes are commonly employed [27,28].

The main contribution in this paper is to provide explicit circuit constructions for the block-encoding of the discretized form of the Hamiltonian in Eq. (1). It is now a $2^n \times 2^n$ matrix of the form

$$\hat{H} = \sum_{k=0}^{\eta-1} (\alpha_k P_{q_k}(\hat{x}) \hat{p}^{m_k} + \alpha_k^* \hat{p}^{m_k} P_{q_k}(\hat{x}));$$

$$\alpha_k \in \mathbb{C}; \quad q_k, m_k \in \mathbb{N}, \quad (3)$$

with $\hat{x}$ and $\hat{p}$ being finite difference matrices for the coordinate and momentum operators. For example, we can introduce the central symmetrical finite-difference scheme for derivative $\frac{\partial U}{\partial x} \rightarrow \frac{U_{h+1} - U_{h-1}}{2\Delta x}$; and we impose periodic spatial boundary conditions over the interval (a, b) :

$$\hat{x} = \begin{pmatrix} a & 0 & \cdots & 0 & 0 \\ 0 & a + \Delta x & \cdots & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & b - \Delta x & 0 \\ 0 & 0 & \cdots & 0 & b \end{pmatrix};$$

$$\hat{p} = -\frac{i}{2\Delta x} \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 & 0 & -1 \\ -1 & 0 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & & \ddots & & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 0 & 1 \\ 1 & 0 & 0 & \cdots & 0 & -1 & 0 \end{pmatrix}. \quad (4)$$

Without loss of generality, we can impose that $-1 \leq a < b \leq 1$. We use this condition later on in the paper. Moreover, we impose that $\sum_{\kappa=0}^{2^n-1} (a + \kappa \Delta x)^2 = 1$. Both those assumption are introduced to simplify normalization constants, making important parts more vivid.

Block-encoding of an operator is defined in the following way.

Definition 1 (Block-encoding, modified Definition 43 from [18]). Suppose that A is an n -qubit operator, $\alpha, \epsilon \in \mathbb{R}_+$; $a, s \in \mathbb{Z}_+$, and let U be an $(a + s + n)$ -qubit unitary such that for any quantum state $|\psi\rangle^n$,

$$U|0\rangle^a|0\rangle^s|\psi\rangle^n = |0\rangle^a|\phi\rangle^{n+s};$$

$$(|0\rangle^s \otimes I^{\otimes n})|\phi\rangle^{n+s} = \tilde{A}|\psi\rangle^n,$$

then we say that U is an (α, s, ϵ) -block-encoding of A , if

$$\|A - \alpha \tilde{A}\| \leq \epsilon.$$

Here we underline that the unitary U can exploit some auxiliary qubits setting them back to zero state. Later we call such qubits "pure ancillas." The general scheme of block-encoding is presented in Fig. 1.

This paper's central contributions are encapsulated in the following two Theorems, which establish a framework for efficient quantum simulation methods and demonstrate the scalability of the proposed approach. Later in the paper, we provide the detailed version of them (see Theorems 8 and 10).

Theorem 1 (Hamiltonian block-encoding). Let H be a Hamiltonian as in Eq. (1) with the number of terms $\eta = 2^\gamma$, $m = \max_k m_k$, $2^l = 2m + 1$, $q = \max_k q_k$, and with matrix representation $\hat{H}$ as in Eq. (3) with normalization constant $\mathcal{N}_H$

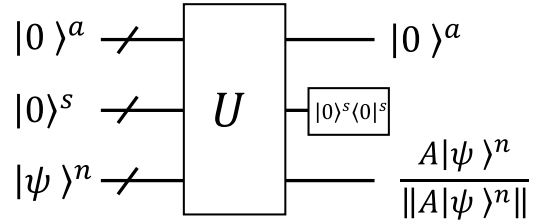


FIG. 1. The general scheme of block-encoding for matrix A . The latest operation for the second register indicates measurement of zero, which project main n -qubit register onto the desired state.

(see Corollary 2), then we can implement a $(\sqrt{2^l} \mathcal{N}_H, l + \gamma + 4, 0)$ -block-encoding of $2^n \times 2^n$ matrix $\hat{H}$

$$U_H|0\rangle^{l+4}|\phi\rangle^n = \frac{1}{\sqrt{2^l} \mathcal{N}_H} \sum_{\substack{i=0, \dots, 2^n-1 \\ j \in F_i^H}} (\hat{H}_{ij} \sigma_j |0\rangle^{l+4} |i\rangle^n + J_{\hat{H}_{ij}}^U |\perp_0\rangle^{n+l+4});$$

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

with resources no greater than:

- (1) $\mathcal{O}(\eta q n^2 + \eta m n)$ quantum gates;
- (2) $3n + \gamma - l - 1$ pure ancillas.

Remark 1. The normalization constant, $\mathcal{N}_H \sim 2^{mn} \eta x_{\max}^q$, should not be considered related to the probability of successfully applying the block-encoded matrix. Instead, it serves solely as a normalization factor for the resulting wave function after applying U_H .

The total probability of success for this operation is $\mathcal{O}(1/m)$. This implies that with probability $\mathcal{O}(1/m)$, we achieve the action of:

$$\frac{H|\psi\rangle}{\|H|\psi\rangle\|}.$$

However, the normalization constant $\mathcal{N}_H$ becomes significant in Theorem 3, where it plays an essential role. Consequently, the total gate count will necessarily include a factor of 2^{mn} , limiting the quantum advantage in the number of partitions N to a polynomial scale, as the forward Euler method has complexity $2^{(m+1)n}$.

We also extend our construction from Eq. (1) to multimode Hamiltonians of the form

$$H^{(d)} = \sum_{k=0}^{\eta_d-1} \alpha_k \prod_{y=0}^{d-1} \otimes R(L_{ky}, P_{q_k^{(y)}}(x_y), p_y^{m_k^{(y)}});$$

$$R(0, P_{q_k^{(y)}}(x_y), p_y^{m_k^{(y)}}) = P_{q_k^{(y)}}(x_y) p_y^{m_k^{(y)}};$$

$$R(1, P_{q_k^{(y)}}(x_y), p_y^{m_k^{(y)}}) = p_y^{m_k^{(y)}} P_{q_k^{(y)}}(x_y), \quad (5)$$

where L_{ky} is a binary number indicating the ordering of the operators; we show how to construct the block-encoding of the discretization of this d -mode Hamiltonian (see Sec. IV).

Theorem 2 (Multidimensional block-encoding). Let $H^{(d)}$ be a Hamiltonian as in Eq. (5) with the number of terms $\eta_d = 2^\gamma$, sparsity 2^l , $n = \sum_y n_y$ -qubit matrix representation $\hat{H}^{(d)}$,

and normalization constant $\mathcal{N}_H^{(d)}$ (see Corollary 3), then we can implement a $(\sqrt{2^l} \mathcal{N}_H^{(d)}, l + \gamma + 3d, 0)$ -block-encoding of $\hat{H}$ with

- (1) $\mathcal{O}(\sum_{k=0}^{\eta_d-1} \sum_{y=0}^{d-1} (q_k^{(y)} n_y^2 + m_k^{(y)} n_y))$ quantum gates;
- (2) $\mathcal{O}(\sum_{y=0}^d n_y + \log(\eta))$ pure ancillas.

Classically, solving such partial differential equations with partitioning $N = 2^n$ requires at least $O(2^n)$ operations, while our quantum approach achieves $O(n^2)$ scaling for the same setup. This remarkable difference opens the door to quantum speedup, enabling the handling of complex problems that are beyond the reach of classical computing.

Main motivating questions. For those not familiar with this area, we must address two questions that motivate our work:

- (1) What is block-encoding, and why is its explicit circuit construction for a Hamiltonian important?
- (2) Why is our particular Hamiltonian $\hat{H}$ in Eq. (3) relevant to many problems?

From this definition, it is clear that if the block-encoding of a matrix can be constructed, the matrix itself can be recovered. The main advantage of dilating the matrix to its block-encoding is so that, given access to the block-encoding, sums of matrices and polynomial approximations of positive and negative power functions of matrices can often be efficiently constructed. Thus, block-encoding can be considered a fundamental building block from which more complex functions of a matrix can be constructed.

There is one particular function of $\hat{H}$ that is of particular interest, and this is the unitary $\exp(i\hat{H}t)$. This is exactly the unitary evolution operator required for Hamiltonian simulation with respect to a time-independent Hamiltonian $\hat{H}$. It turns out (Theorem 3) that by having access to the block-encoding of $\hat{H}$, one can construct a near-optimal implementation of $\exp(i\hat{H}t)$ with respect to time t (linear in t) and has an additive logarithmic dependence on inverse error ϵ .

Theorem 3 (Optimal block-Hamiltonian simulation; Theorem 58 from Ref. [18]). Let $t \in \mathbb{R}/\{0\}$, $\epsilon \in (0, 1)$ and let U_H be an $(\alpha, a, 0)$ -block-encoding of the $2^n \times 2^n$ Hamiltonian H . Then we can implement an ϵ -precise Hamiltonian simulation unitary V , which is a $(2, a + 2, \epsilon)$ -block-encoding of e^{-itH} , with resources no greater than

- (1) $\Omega(t, \epsilon)$ uses of U_H or its inverse;
- (2) 1 uses of controlled-U or its inverse;
- (3) $\Omega(\alpha t, \epsilon)(16a + 50) + 4$ one-qubit gates;
- (4) $\Omega(\alpha t, \epsilon)(12a + 38)$ C-NOTs;
- (5) $\mathcal{O}(n)$ ancilla qubits.

$\Omega(\alpha t, \epsilon)$ is implicitly defined through inequality for the truncation parameter $g \geq \Omega(\alpha t, \epsilon)$

$$\frac{1.07}{\sqrt{g}} \left(\frac{\alpha e t}{2g} \right)^g \leq \epsilon.$$

The inequality determines the scaling of the algorithm (Lemma 59 from Ref. [18])

$$\Omega(\alpha t, \epsilon) = \mathcal{O} \left(\alpha t + \frac{\ln(1/\epsilon)}{\ln(e + \frac{\ln(1/\epsilon)}{\alpha t})} \right).$$

Since all Hamiltonian simulation problems are concerned with simulating the unitary evolution operator for a given

Hamiltonian $\hat{H}$, having the block-encoding of $\hat{H}$ is a necessary starting point if we choose to use the block-encoding formalism for Hamiltonian simulation. Often, in most applications of block-encoding, this is assumed to be provided, like a black box, and the total cost for an algorithm only counts the total number of queries to this black box. However, for a real implementation of the quantum simulation scheme, it is absolutely necessary to explicitly construct this black box. The total number of queries of this black box for an algorithm, even if efficient, is not a meaningful measure when considering the total cost for real implementation, unless we know that this black box itself is also efficient to construct.

However, explicitly constructing this block-encoding for arbitrary $\hat{H}$ is very difficult and is an open problem. In this paper, we focus on a particular class of $\hat{H}$ —the discretizations of Eqs. (1) and (5). We briefly discuss some examples of applications including (a) time-independent and time-dependent Hamiltonian simulation, (b) the quantum simulation of all linear partial differential equations (and some nonlinear problems too), and (c) ground state and thermal state preparation of $\hat{H}$. The detailed implementation of these schemes we leave for future work.

Hamiltonian simulation. Schrödinger's equation for η particles in $d = 3$ dimensions with time-independent interacting potential $V(\hat{x}_0, \dots, \hat{x}_{\eta-1}, \hat{y}_0, \dots, \hat{y}_{\eta-1}, \hat{z}_0, \dots, \hat{z}_{\eta-1})$ has the corresponding Hamiltonian

$$\hat{H} = \sum_{k=0}^{\eta-1} \frac{\hat{p}_{x_k}^2}{2m_k} + \frac{\hat{p}_{y_k}^2}{2m_k} + \frac{\hat{p}_{z_k}^2}{2m_k} + V(\hat{x}_0, \dots, \hat{x}_{\eta-1}, \hat{y}_0, \dots, \hat{y}_{\eta-1}, \hat{z}_0, \dots, \hat{z}_{\eta-1}). \quad (6)$$

Clearly the kinetic term is of the form in Eq. (5). The potential belongs to Eq. (5) only if it is polynomial in x, y, z , for instance, a sum of harmonic wells. Polynomial potential functions can also be approximate representations in Taylor expansions of more complex potentials.

We can also easily extend to the case of time-dependent potentials V . We can use a simple transformation [21], turning a time-dependent Hamiltonian $H(t)$ into a time-independent Hamiltonian $\hat{H}$ with one extra mode labeled s

$$H(t) \rightarrow \hat{H} = \hat{p}_s \otimes \mathbf{1} + H(\hat{x}_s), \quad (7)$$

where $[\hat{x}_s, \hat{p}_s] = i\mathbf{1}_s$. From the method in Ref. [21], if time-independent quantum simulation of $\hat{H}$ is performed, one can easily and efficiently recover the evolution with respect to $H(t)$. Thus, for almost any $H(t)$, we can claim that Eq. (7) is also of the form in Eq. (5).

Quantum simulation of partial differential equations. It is known that for any d -dimensional linear partial differential equation for $u(x, t)$, $x = x_1, \dots, x_d$, can be written in the form [20]

$$\frac{d\mathbf{u}(t)}{dt} = -i\mathbf{A}(t, \hat{x}, \hat{p})\mathbf{u}(t), \quad \mathbf{u}_0 = \mathbf{u}(0),$$

$$\hat{x} = \hat{x}_1, \dots, \hat{x}_d, \quad \hat{p} = \hat{p}_1, \dots, \hat{p}_d, \quad (8)$$

where $\mathbf{u}(t) = \int_{-\infty}^{\infty} u(t, x) |x\rangle dx$. Here, $\mathbf{A}(t, \hat{x}, \hat{p})$ is a linear operator consisting of factors of $\hat{x}, \hat{p}$ depending on the form of the original partial differential equation. Each factor of x in the original partial differential equation gives rise to $\hat{x}$, and

each derivative $\partial^l/\partial x_j^l$ in the partial differential equation gives a contribution of $(i\hat{p}_j)^l$. However, here, $\mathbf{A}(t) \neq \mathbf{A}^\dagger(t)$ is not generally Hermitian, so cannot be seen directly as a Hamiltonian.

To turn this simulation problem into a Hamiltonian simulation problem, we want to find a corresponding $\hat{H}$, which is also a function of $\hat{x}$ and $\hat{p}$. This is possible by applying the method of Schrödingerization [14,20,29], which can transform $\mathbf{A} = \mathbf{A}_1 - i\mathbf{A}_2$ into a Hamiltonian $\hat{H}$ acting on one extra mode labeled ξ

$$\mathbf{A}(t) \rightarrow \hat{H} = \mathbf{A}_2(t) \otimes \hat{x}_\xi + \mathbf{A}_1(t) \otimes \mathbf{1}_\xi. \quad (9)$$

Here, $\mathbf{A}_1(t) = (\mathbf{A}(t) + \mathbf{A}^\dagger(t))/2 = \mathbf{A}_1^\dagger$ corresponds to the completely Hermitian part of $\mathbf{A}(t)$ and $\mathbf{A}_2(t) = i(\mathbf{A}(t) - \mathbf{A}^\dagger(t))/2 = \mathbf{A}_2^\dagger(t)$ is associated with the completely anti-Hermitian part of $\mathbf{A}(t)$. The method of Schrödingerization allows the simple recovery of the quantum state proportional to $\mathbf{u}(t)$ by using Hamiltonian simulation in $\hat{H}(t)$. We can easily turn this into a time-independent Hamiltonian using the method previously described in Eq. (7).

This means if $\mathbf{A}(t)$ is a polynomial sum of $\hat{x}$ and $\hat{p}$, then clearly $\hat{H}(t)$ will also be a polynomial sum of $\hat{x}$, $\hat{p}$. We underline here that for more complex functions like cosine, sine, and so on, we can transform it to polynomials using some expansions (e.g., Taylor-like). Every partial differential equation has only integer-valued derivatives of order $l > 0$ (with the exception of fractional partial differential equations), so these will give a contribution to $\hat{H}(t)$ of the polynomial form $\hat{p}^l$. This means, for instance, any constant coefficient partial differential equation are in the form of Eq. (5).

In addition, some nonlinear partial differential equations like the Hamilton-Jacobi and scalar hyperbolic, as well as all nonlinear ordinary differential equations, can also be written in the linear form Eq. (8) [12,20,30].

Ground state and thermal state preparation. Given the block-encoding to $\hat{H}$, one can also perform ground state preparation or thermal state preparation with respect to the Hamiltonian $\hat{H}$, where $\hat{H}$ is of the form in Eq. (5). There are two ways to go about this. A simple method that involves only block-encoding construction of $\exp(-i\hat{H}t)$, that is, Hamiltonian simulation, is again to apply Schrödingerization, with the choice of $\mathbf{A} = -i\mathbf{A}_2 = -i\hat{H}$; see Ref. [29] for more details.

An alternative is not to construct block-encodings of $\exp(-i\hat{H}t)$, but to construct block-encodings of functions like $\exp(-\hat{H}t)$ or $\exp(-\beta\hat{H})$ (see Ref. [18] for more details). These operators can be similarly used for ground state and thermal state preparation.

A. Road map

The steps in constructing the block-encoding for our $\hat{H}$ requires many steps, and it can be easy to get lost while compiling many pieces of the puzzle. To make it easier to use our technique, the road map for our algorithm is shown in Fig. 2. For convenience, Table I clarifies the notation used.

B. Preliminaries

Now we introduce some mathematical and quantum computing objects, and the system of indices. We start with a

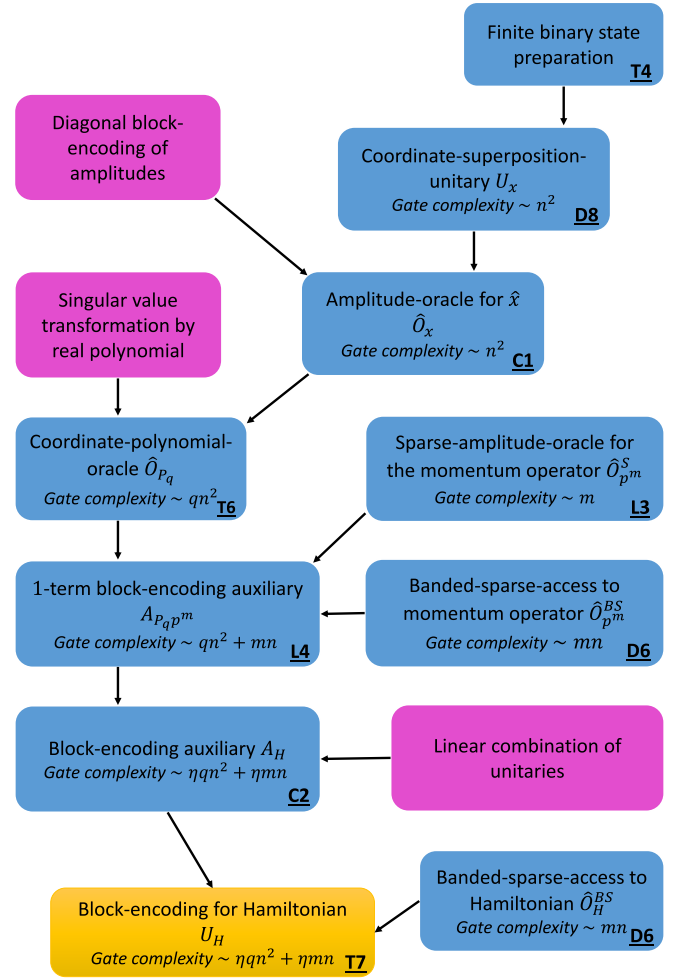


FIG. 2. Road map for implementing an efficient block-encoding for Hamiltonian with form (1). Each block indicates an important step in the overall algorithm. Moreover, we provide the notation for each structure and the scaling for its construction. The precise number of gates is provided in the text. Some blocks indicate Theorems (T), Corollaries (C), Lemmas (L), and Definitions (D) where it can be found. The boxes are blue if the step is an original concept developed in this paper. On the contrary, pink indicates research from other papers. We also provide tips for gate complexity notation understanding: n = number of qubits, m = the maximum degree of momentum operator contributing to the Hamiltonian, q = the maximum degree of coordinate operator contributing to the Hamiltonian, and η = number of terms in the Hamiltonian.

multicontrol operator, which is a common structure in quantum computing allowing for the application a certain gate with a given condition.

Definition 2 (Multicontrol operator). Let U be an m -qubit quantum unitary and b a bit string of length n . We define C_U^b as an $n + m$ quantum unitary that applies U to an m -qubit quantum register if and only if the n -qubit quantum register is in state $|b\rangle$:

$$C_U^b = |b\rangle^n \langle b|^n \otimes U + \sum_{\substack{i=0, \dots, 2^n-1 \\ i \neq b}} |i\rangle^n \langle i|^n \otimes I^{\otimes m}.$$

The general structure of a multicontrol operator is illustrated in Fig. 3. It is known [25] that Toffoli gate can be realized using six C-NOTs and eight single-qubit operations. Consequently, the resource requirement for a single multicontrol operator includes no more than one C_U^1 operator, $16n - 16$ single-qubit operations, and $12n - 12$ C-NOTs, simplifying the complexity of C_X^b to $C_X^{11\dots 1}$ for simplicity. Additionally, we utilize $n - 1$ ancillary qubits initialized to the zero state, which we further refer to as pure ancillas.

Definition 3 (Pure ancilla). Let U be an $n + m$ -qubit unitary operator such that for any arbitrary $|\psi\rangle^n$,

$$U|0\rangle^m|\psi\rangle^n = |0\rangle^m|\phi\rangle^n,$$

where $|\phi\rangle^n$ is some quantum state; then we say that the m -qubit quantum register is pure ancillas for operator U , if before and after the action it is in the state $|0\rangle^m$. In Fig. 3(c) we use $m - 2$ auxiliary qubits for efficient C-NOT construction. Exploitation of this qubits is divided into two epochs: (i) an entangling stage that allows us to achieve some sophisticated quantum states, and (ii) a purification stage that sets ancillas back to zero state.

Definition 5 (Banded-sparse matrix index). For any given banded sparse matrix A , we denote $A^{(s)}$ as the s -th nonzero element in the matrix's first row. Here is an example:

$$A = \begin{pmatrix} 2 & 3 & 0 & 0 & 0 & 0 & 0 & 4 \\ 4 & 2 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 2 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 2 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 2 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 2 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 2 & 3 \\ 3 & 0 & 0 & 0 & 0 & 0 & 4 & 2 \end{pmatrix} = \begin{pmatrix} A^{(0)} & A^{(1)} & 0 & 0 & 0 & 0 & 0 & A^{(2)} \\ A^{(2)} & A^{(0)} & A^{(1)} & 0 & 0 & 0 & 0 & 0 \\ 0 & A^{(2)} & A^{(0)} & A^{(1)} & 0 & 0 & 0 & 0 \\ 0 & 0 & A^{(2)} & A^{(0)} & A^{(1)} & 0 & 0 & 0 \\ 0 & 0 & 0 & A^{(2)} & A^{(0)} & A^{(1)} & 0 & 0 \\ 0 & 0 & 0 & 0 & A^{(2)} & A^{(0)} & A^{(1)} & 0 \\ 0 & 0 & 0 & 0 & 0 & A^{(2)} & A^{(0)} & A^{(1)} \\ A^{(1)} & 0 & 0 & 0 & 0 & 0 & A^{(2)} & A^{(0)} \end{pmatrix}.$$

Remark 2. We extend the definition of the banded-sparse matrix index to cases where a sparse $N \times N$ matrix A becomes a banded matrix upon substituting all nonzero elements with 1, reflecting the band structure of nonzero elements. We underline that our main interest here is to reflect the band position of the nonzero elements. Thus, the notation $A_i^{(s)}$ means that we first address an index of s -th nonzero elements in the first row, then add i (modulo N) to get the column number. Here is an example:

$$A = \begin{pmatrix} 2 & 6 & 0 & 0 & 0 & 0 & 0 & 4 \\ 1 & 2 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6 & 7 & 7 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 6 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 8 & 9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 9 & 4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6 & 6 & 6 \\ 9 & 0 & 0 & 0 & 0 & 0 & 5 & 2 \end{pmatrix} = \begin{pmatrix} A_0^{(0)} & A_0^{(1)} & 0 & 0 & 0 & 0 & 0 & A_0^{(2)} \\ A_1^{(2)} & A_1^{(0)} & A_1^{(1)} & 0 & 0 & 0 & 0 & 0 \\ 0 & A_2^{(2)} & A_2^{(0)} & A_2^{(1)} & 0 & 0 & 0 & 0 \\ 0 & 0 & A_3^{(2)} & A_3^{(0)} & A_3^{(1)} & 0 & 0 & 0 \\ 0 & 0 & 0 & A_4^{(2)} & A_4^{(0)} & A_4^{(1)} & 0 & 0 \\ 0 & 0 & 0 & 0 & A_5^{(2)} & A_5^{(0)} & A_5^{(1)} & 0 \\ 0 & 0 & 0 & 0 & 0 & A_6^{(2)} & A_6^{(0)} & A_6^{(1)} \\ A_7^{(1)} & 0 & 0 & 0 & 0 & 0 & A_7^{(2)} & A_7^{(0)} \end{pmatrix}.$$

Based on the introduced sparse access we define a quantum operation that transforms that index into the column index of a nonzero element.

Definition 6 (Banded-sparse-access). Given a 2^l sparse $2^n \times 2^n$ matrix A with nonzero elements arranged to form a banded matrix when replacing all nonzero elements with 1,

Now, we introduce a banded matrix whose nonzero elements form a band shape. This object is particularly interesting as $\hat{p}^m$ is a banded matrix.

Definition 4 (Banded matrix). A banded matrix is defined as an $N \times N$ matrix in which each i -th row is the $i - 1$ times right-cyclic permuted first row

$$\forall i, j : A_{ij} = A_{0j-i},$$

where $j - i$ means subtraction modulo N . Thus, the set of indices is fully determined by its first row. The following 5×5 matrix is an example of such banded matrices

$$\begin{pmatrix} 6 & 1 & 0 & 9 & 8 \\ 8 & 6 & 1 & 0 & 9 \\ 9 & 8 & 6 & 1 & 0 \\ 0 & 9 & 8 & 6 & 1 \\ 1 & 0 & 9 & 8 & 6 \end{pmatrix}.$$

We define a s sparse matrix as a matrix that contains no more than s nonzero elements in each row. In our research, we focus on sparse banded matrices. To analyze these matrices in a convenient way, we introduce a sparse index specifically tailored to them.

we define a Banded-sparse-access unitary oracle to A as

$$\hat{O}_A^{BS} |0\rangle^{n-l} |s\rangle^l |i\rangle^n := |r_{si}\rangle^n |i\rangle^n,$$

where $r_{si} = r_{s0} + i \mod 2^n$ corresponds to the (s) -th banded-sparse matrix index (s -th nonzero element in the i -th line). If

TABLE I. Notations.

Notation	Name	Formula	First app.
$\hat{H}$	Finite-difference representation of H	$\hat{H} = \sum_{k=0}^{\eta-1} (\alpha_k P_{q_k}(\hat{x}) \hat{p}^{m_k} + \alpha_k^* \hat{p}^{m_k} P_{q_k}(\hat{x}))$	Eq. (3)
$\hat{x}$	Finite-difference representation of coordinate operator x over the interval (a, b)	$\hat{x} = \begin{pmatrix} a & 0 & \cdots & 0 & 0 \\ 0 & a + \Delta x & \cdots & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & b - \Delta x & 0 \\ 0 & 0 & \cdots & 0 & b \end{pmatrix}$	Eq. (4)
$\hat{p}$	Finite-difference representation of momentum operator p for periodic spatial boundary conditions and the central symmetrical finite-difference scheme	$\hat{p} = -\frac{i}{2\Delta x} \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 & 0 & -1 \\ -1 & 0 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & & & \ddots & & & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 0 & 1 \\ 1 & 0 & 0 & \cdots & 0 & -1 & 0 \end{pmatrix}$	Eq. (4)
C_U^b	Multicontrol operator that apply U on the target if the control is in $ b\rangle$	$C_U^b = b\rangle^n \langle b ^n \otimes U + \sum_{i=0, \dots, 2^n-1, i \neq b} i\rangle^n \langle i ^n \otimes I^{\otimes m}$	Def. 2
$\hat{O}_A^{BS}$	Banded-sparse-access to matrix A	$\hat{O}_A^{BS} 0\rangle^{n-l} s\rangle^l i\rangle^n := r_{si}\rangle^n i\rangle^n$	Def. 6
$\hat{O}_{p^m}^S$	Amplitude-unitary-oracle-access to the operator $\hat{p}^m$	$\hat{O}_{p^m}^S 0\rangle^1 s\rangle^l := \frac{(\hat{p}^m)^{(s)}}{\sqrt{\mathcal{N}_{p^m}}} 0\rangle^1 s\rangle^l + \sqrt{1 - \frac{ (\hat{p}^m)^{(s)} ^2}{\mathcal{N}_{p^m}}} 1\rangle^1 s\rangle^l$	Eq. (15)
$\mathcal{N}_{p^m}$	Normalization constant for momentum operator	$\mathcal{N}_{p^m} \geq \ \hat{p}^m\ _{\max}^2$	Eq. (15)
U_{x^q}	Coordinate-superposition-unitary	$U_{x^q} 0\rangle^n = \frac{1}{\sqrt{\sum_{\kappa=0}^{2^n-1} \hat{x}_{\kappa}^q}} \sum_{i=0}^{2^n-1} \hat{x}_{ii}^q i\rangle^n$	Def. 8
$A_{p_q p^m}$	One-term block-encoding auxiliary $P_q(\hat{x}) \hat{p}^m$	$A_{p_q p^m} \sum_{j=0}^{2^n-1} \sigma_j 0\rangle^3 0\rangle^n j\rangle^n = \frac{(-1)^m}{\sqrt{2^l \mathcal{N}_{p^m}}} \sum_{i=0, \dots, 2^n-1} \underbrace{P_q(\hat{x}_{ii}) (\hat{p}^m)_{ij}}_{(P_q(\hat{x}) \hat{p}^m)_{ij}} \sigma_j 0\rangle^1 i\rangle^n j\rangle^n + \dots$	Eq. (16)
$A_{p^m p_q}$	One-term block-encoding auxiliary $\hat{p}^m P_q(\hat{x})$	$A_{p^m p_q} \sum_{j=0}^{2^n-1} \sigma_j 0\rangle^3 0\rangle^n j\rangle^n = \frac{(-1)^m}{\sqrt{2^l \mathcal{N}_{p^m}}} \sum_{i=0, \dots, 2^n-1} \underbrace{(\hat{p}^m)_{ij} P_q(\hat{x}_{jj})}_{(\hat{p}^m P_q(\hat{x}))_{ij}} \sigma_j 0\rangle^1 i\rangle^n j\rangle^n + \dots$	Eq. (16)
$F_i^{(m)}$	Set of indexes of nonzero elements in the i -th row of $\hat{p}^m$	None	Eq. (16)
A_H	Block-encoding auxiliary	$(\langle 0 ^{\gamma+4} \otimes I^{\otimes 2n}) A_H \sum_{j=0}^{2^n-1} \sigma_j 0\rangle^{\gamma+4} 0\rangle^n j\rangle^n \sim \sum_{i=0, \dots, 2^n-1, j \in F_i^H} \hat{H}_{ij} i\rangle^n j\rangle^n$	Cor. 2
F_i^H	Set of indexes of nonzero elements in the i -th row of matrix $\hat{H}$	None	Cor. 2
$\mathcal{N}_H$	Normalization constant for A_H	$2 \sum_{\kappa=0}^{2^\gamma-1} \alpha_\kappa \sqrt{2^{l_\kappa} \mathcal{N}_{p^{n_\kappa}}}$	Cor. 2
U_H	Block-encoding of the Hamiltonian H	$\Phi_H \sum_{i=0, \dots, 2^n-1, j \in F_i^H} U_H 0\rangle^\lambda \phi\rangle^n = (\hat{H}_{ij} \sigma_j 0\rangle^\lambda + J_{\hat{H}_{ij}}^U \perp_0\rangle^\lambda) i\rangle^n$	Thm. 8

the transposed matrix A^T shares the same nonzero element positions as A , then $\hat{O}_A^{BS} = \hat{O}_{A^T}^{BS}$.

Lemma 1. The Banded-sparse-access $\hat{O}_A^{BS}$ to 2^l sparse $2^n \times 2^n$ matrix A with nonzero elements arranged to form a banded matrix when replacing all nonzero elements

with 1 can be implemented with resources no greater than:

- (1) $(2^l + 1)(32n - 48)$ one-qubits operations;
- (2) $25 * 2^l n - 36 * 2^l + 32n - 48$ C-NOTs;
- (3) $n - 1$ pure ancillas.

TABLE II. Table of circuit's complexity. Here, n denotes the number of qubits such that 2^n corresponds to the size of the Hamiltonian $\hat{H}$ in Eq. (3) and γ denotes the number of qubits such that $\eta = 2^\gamma$ is the number of terms in Eq. (3); l is also the number of qubits such that 2^l indicates the sparsity of the corresponding matrix. For example, for $\hat{O}_A^{BS}$, the Banded-sparse-access to matrix A , 2^l is the sparsity of matrix A . Furthermore, we define $q_{\max} = \max_k q_k$, $2^{l_{\max}} = \max_k m_k$, $2^{l'_{\max}} = \max_k (2m_k + 1)$, where q_k and m_k are the exponent of finite-difference matrices for the coordinate $\hat{x}$ and momentum operator $\hat{p}$ of Hamiltonian $\hat{H}$ in Eq. (3), respectively.

Name	Notation	Scaling	C-NOTS	One-qubits operations	Ancillas Main Thm.
Banded-sparse-access to matrix A	$\hat{O}_A^{BS}$	$\mathcal{O}(mn)$	$(2^l + 1)(32n - 48)$	$25 \times 2^l n - 36 \times 2^l + 32n - 48$	$n - 1$ Lem. 1
Coordinate-superposition-unitary	U_{x^q}	$\mathcal{O}\left(\frac{n^{q+1}}{q!}\right)$	$\frac{12n^{q+1}}{q!}$	$\frac{16n^{q+1}}{q!}$	$n - 1$ Thm. 5
Amplitude-oracle for $\hat{x}$	$\hat{O}_x$	$\mathcal{O}(n^2)$	$1864n^2 - 860n - 92$	$2304n^2 - 1064n - 109$	$2n - 1$ Cor. 1
Coordinate-polynomial-oracle	$\hat{O}_{p_q}$	$\mathcal{O}(qn^2)$	$1864qn^2 - 860qn - 92q$	$2304qn^2 - 1064qn - 108q$	$2n - 1$ Thm. 7
Sparse-amplitude-oracle for the operator $\hat{p}^m$	$\hat{O}_{p^m}^S$	$\mathcal{O}(m)$	2^l	2^l	0 Lem. 4
One-term block-encoding auxiliary	$A_{p_q p^m}$ ($A_{p^m p_q}$)	$\mathcal{O}(qn^2 + mn)$	$1864qn^2 - 860qn + 25 \times 2^l n + 32n - 92q - 17 \times 2^l - 48$	$2304qn^2 - 1064qn + 32 \times 2^l n + 32n - 108q - 48 \times 2^l + 2^l - 48$	$2n - 1$ Lem. 4
Block-encoding auxiliary	A_H	$\mathcal{O}(\eta q_{\max} n^2 + \eta m_{\max} n)$	$2^{\gamma+1}[15792n^2 q + 107 \times 2^{l_{\max}+1} + 256n + 12\gamma - 7288nq - 768q - 49 \times 2^{l_{\max}+1} - 383]$	$2^{\gamma+2}[9760n^2 q + 33 \times 2^{l_{\max}+2} n + 160n + 8\gamma - 4504nq - 476q - 115 \times 2^{l_{\max}} - 239]$	$2n + \gamma - 1$ Cor. 2
Block-encoding of Hamiltonian H	U_H	$\mathcal{O}(\eta q_{\max} n^2 + \eta m_{\max} n)$	$2^{\gamma+1}[15792n^2 q + 107 \times 2^{l_{\max}+1} + 256n + 12\gamma - 7288nq - 768q - 49 \times 2^{l_{\max}+1} - 383] + 25 \times 2^l n - 36 \times 2^l + 32n - 48$	$2^{\gamma+2}[9760n^2 q + 33 \times 2^{l_{\max}+2} n + 160n + 8\gamma - 4504nq - 476q - 115 \times 2^{l_{\max}} - 239] + 2^{l+5} n - 3 \times 2^{l+4} + 32n + l - 48$	$3n + \gamma + l - 1$ Thm. 8

Proof. The explicit circuit design is shown and explained in Appendix A. It is important to note that when exponentiating the matrix $\hat{p}$ to the power of m , denoted $\hat{p}^m$, we obtain a matrix that also qualifies as a sparse banded matrix, with $m + 1$ nonzero elements in each row. ■

III. EFFICIENT BLOCK-ENCODING USING AMPLITUDE HAMILTONIAN QUERY ACCESS

In this section, we present efficient block-encoding techniques utilizing Amplitude Hamiltonian query access. This advanced approach enables us to harness the full potential of quantum computing for simulating complex Hamiltonian dynamics with high precision. Through a detailed exploration of Hamiltonian superposition, we establish an auxiliary unitary framework that significantly enhances our ability to analyze and interact with Hamiltonian operators in a quantum computational setting. This is achieved by synthesizing oracles to specific operators and employing a combination of sophisticated techniques, including banded-sparse-access and coordinate-polynomial-oracle, all supplemented by straightforward unitary operations. This section culminates in the development of a block-encoding for the Hamiltonian [Eq. (1)], providing us with a new method to access Hamiltonian information directly and efficiently.

A. Coordinate-polynomial-oracle

In this subsection, we construct an auxiliary unitary framework for the polynomial $P_q(\hat{x})$, denoted $\hat{O}_{P_q}$. We further combine this coordinate-polynomial-oracle with a similar structure for $\hat{p}^m$ to build the oracle for $P_q(\hat{x})\hat{p}^m$.

Definition 7 (Coordinate-polynomial-oracle). Let $P_q(x)$ be a degree- q polynomial satisfying that

- (1) $P_q(x)$ has parity- $(q \bmod 2)$;
- (2) $P_q(x) \in \mathbb{R}[x]$;

$|\psi\rangle^n$ contains computational basis states only with $|i|_b \leq \chi$

$$H_W^{\otimes n} \left(\frac{1}{\mathcal{N}_\psi} \sum_{q=0}^{\chi} \alpha_q U_{x^q} |0\rangle^n \right) = H_W^{\otimes n} \left(\frac{1}{\mathcal{N}_\psi} \sum_{\substack{q=0, \dots, \chi \\ i=0, \dots, 2^n-1}} \alpha_q \hat{x}_{ii}^q |i\rangle^n \right) = \sum_{|i|_b \leq \chi} \beta_i |i\rangle^n;$$

$$\beta_i = \frac{1}{2^{n/2} \mathcal{N}_\psi} \sum_{\substack{q=0, \dots, \chi \\ \kappa=0, \dots, 2^n-1}} \alpha_q \hat{x}_{\kappa\kappa}^q (-1)^{\sum_{m=0}^{n-1} \kappa_m i_m}; \quad H_W = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},$$

where m indicates a bit string index.

We note that the equation for β_i generally requires exponential resources of classical computer to be numerically find. However, we stress that in this work we use coordinate-superposition-unitary only for the first degree ($q = 1$). The β_i can be computed analytically when q is a small number. The following theorem shows an explicit quantum circuit for constructing U_{x^q} .

Theorem 5 (Finite binary norm state preparation).

Let $|\psi\rangle^n = \sum_{|i|_b \leq q} \beta_i |i\rangle^n$ be an arbitrary normalized

- (3) for all $x \in [-1, 1]$: $|P_q| < 1$.

And let $\hat{x}$ be n -qubit matrix representations of the coordinate operator, as in Eq. (4). Then we say that $\hat{O}_{P_q}$ is the coordinate-polynomial-oracle for P_q if

$$\hat{O}_{P_q} |0\rangle^\lambda |i\rangle^n = P_q(\hat{x}_{ii}) |0\rangle^\lambda |i\rangle^n + J_{P_q}^{(i)} |\perp_0\rangle^{\lambda+n},$$

where $|\perp_0\rangle^{\lambda+n}$ is orthogonal to $|0\rangle^\lambda$. In other words, coordinate-polynomial-oracle is a $(1, \lambda, 0)$ -block-encoding of the oracle access.

Next, we design an auxiliary unitary operation which encodes the diagonal of $\hat{x}^q$ operator from Eq. (4) into a quantum state.

Definition 8 (Coordinate-superposition-unitary). Let $\hat{x}$ be a n -qubit finite-difference representation of quantum coordinate operator as in Eq. (4), then we define a n -qubit unitary U_{x^q} that prepares a quantum states with amplitudes encoding the diagonal of $\hat{x}^q$:

$$U_{x^q} |0\rangle^n = \frac{1}{\sqrt{\sum_{\kappa=0}^{2^n-1} (a + \kappa \Delta x)^{2q}}} \sum_{i=0}^{2^n-1} (a + i \Delta x)^q |i\rangle^n$$

$$=: \frac{1}{\sqrt{\sum_{\kappa=0}^{2^n-1} \hat{x}_{\kappa\kappa}^{2q}}} \sum_{i=0}^{2^n-1} \hat{x}_{ii}^q |i\rangle^n.$$

Theorem 4 (Discrete Hadamard-Walsh transform for a polynomial superposition; Appendix A in [31]). Let $|\psi\rangle^n$ be a n -qubit quantum state that is an arbitrary superposition of U_{x^q} (see Definition 8) acting on $|0\rangle^n$

$$|\psi\rangle^n = \frac{1}{\mathcal{N}_\psi} \sum_{q=0}^{\chi} \alpha_q U_{x^q} |0\rangle^n; \quad \alpha_q \in \mathbb{C},$$

where $\mathcal{N}_\psi$ is a normalization factor, $|i|_b = \sum_{\kappa=0}^{n-1} i_\kappa$ is a binary norm, then the discrete Hadamard-Walsh transform [32] of the

superposition of n -qubit computational basis states with binary norm no greater than $q \in \mathbb{N}$ with $\beta_i \in \mathbb{R}$ then the state preparation unitary

$$A_\psi |0\rangle^n = |\psi\rangle^n$$

can be implemented using:

- (1) $\sum_{i=0}^{n-1} \sum_{s=i}^{\min\{q+i-1, n-1\}} \binom{i}{s} A(i)$ one-qubit operations;
- (2) $\sum_{i=0}^{n-1} \sum_{s=i}^{\min\{q+i-1, n-1\}} \binom{i}{s} B(i)$ C-NOTs;
- (3) $n - 2$ pure ancillas;

where

$$A(i) = \begin{cases} 1, & \text{if } i = 0, \\ 2, & \text{if } i = 1, \\ 16i - 14, & \text{if } i \geq 2 \end{cases} \quad B(i) = \begin{cases} 0, & \text{if } i = 0, \\ 2, & \text{if } i = 1, \\ 12i - 10, & \text{if } i \geq 2 \end{cases} \quad \binom{i}{s} = \frac{s!}{i!(s-i)!}.$$

Proof. An explicit implementation of A_ψ and a rule of choosing parameters of the quantum circuit are presented in Appendix C. The expressions for complexity are quite cumbersome, which is why we provide asymptotic with $n \gg q$, $q < 50$, the gate complexity becomes: (i) $\frac{16n^{q+1}}{q!}$ one-qubit operations, (ii) $\frac{12n^{q+1}}{q!}$ C-NOTs. ■

Theorems 4 and 5 allow us to build a coordinate-superposition-unitary U_x :

$$U_x|0\rangle^n = \sum_{i=0}^{2^n-1} \hat{x}_{ii}|i\rangle^n, \quad (10)$$

we note that the normalization constant is assumed to be 1 in this paper.

1. From a superposition unitary to amplitude-oracle

In this subsection, we exploit the technique from Refs. [33,34] to transform U_x to oracle $\hat{O}_x$:

$$\hat{O}_x|0\rangle^1|i\rangle^n = \hat{x}_{ii}|0\rangle^1|i\rangle^n + J_x|\perp_0\rangle^{n+1}, \quad (11)$$

where $|\perp_0\rangle^{n+1}$ is orthogonal to $|0\rangle^1|i\rangle^n$ for any $i = 0, 1, 2, \dots, 2^n - 1$.

We begin our algorithm from the discussion of the work in Ref. [33]. Given a unitary U_S ,

$$U_S|0\rangle^h = \sum_{j=0}^{2^h-1} \psi_j^S|j\rangle^h =: |\psi^S\rangle; \quad \psi_j^S \in \mathbb{C}; \quad \sum_{j=0}^{2^h-1} |\psi_j^S|^2 = 1; \quad (12)$$

the authors suggest an algorithm to build a $(1, h+2, 0)$ block-encoding [35] of the $A_{(p)}^S := \text{diag}(\text{Re}((-i)^p \psi_0^S), \text{Re}((-i)^p \psi_1^S), \dots, \text{Re}((-i)^p \psi_{2^h-1}^S))$ (see Lemma 6 from [33]). We show the direct view of the block-encoding and all the operators used in Eq. (13):

$$\begin{aligned} U_{(p)}^{(S)} &:= (XZX \otimes I^{\otimes 2h+1})(H_W \otimes W_p^\dagger)(|0\rangle^1|0\rangle^1 \\ &\quad \otimes G_p + |1\rangle^1\langle 1|^1 \otimes G_p^\dagger)(H_W \otimes W_p); \\ \tilde{Z} &:= I^{\otimes h} \otimes Z \otimes I^{\otimes h}; \quad \tilde{H}_W := I^{\otimes h} \otimes H_W \otimes I^{\otimes h}; \\ \tilde{S} &:= I^{\otimes h} \otimes S \otimes I^{\otimes h}; \\ R &:= (I^{\otimes h+1} - 2|0\rangle^{h+1}\langle 0|^{h+1}) \otimes I^{\otimes h}; \\ U_C^S &:= (U_S \otimes |0\rangle^1\langle 0|^1 + I^{\otimes h} \otimes |1\rangle^1\langle 1|^1) \otimes I^{\otimes h}; \\ U_{\text{Copy}} &:= I^{\otimes h} \otimes |0\rangle^1\langle 0|^1 \otimes I^{\otimes h} \\ &\quad + \sum_{k,j=0}^{2^h} |j \oplus k\rangle^h \langle j|^h \otimes |1\rangle^1\langle 1|^1 \otimes |k\rangle^h \langle k|^h; \\ W_p &:= \tilde{H}_W \tilde{S}^p U_{\text{Copy}} U_C^S \tilde{H}_W; \quad G_p := W_p R W_p^\dagger \tilde{Z}, \end{aligned} \quad (13)$$

where $S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$, $p = 1/0$ means that $\tilde{S}$ is applied/not applied. A construction of all those operators with one- and two-qubit operations is well described in the article. Let us now consider the following lemma.

Lemma 2 (Eigensystem of $-\frac{1}{2}W_p^\dagger(G_p + G_p^\dagger)W_p$, Lemma 5 from Ref.1 [33]). Let $p = \{0, 1\}$. The operator $-\frac{1}{2}W_p^\dagger(G_p + G_p^\dagger)W_p$ has eigenvectors $|0\rangle^h|0\rangle^1|k\rangle^h$, with associated eigenvalue $\text{Re}(\psi_k^S)$ if $p = 0$ and $\text{Im}(\psi_k^S)$ if $p = 1$.

Since the operation $-\frac{1}{2}W_p^\dagger(G_p + G_p^\dagger)W_p$ is supposed to act on superposition of states like $|0\rangle^h|0\rangle^1|k\rangle^h$, it is clear that at the end of operation the auxiliary registers are set to zero state. Consequently, these $h+1$ auxiliary qubits should be considered pure ancillas because we do not have to measure those qubits to get the desired action of $A_{(p)}^S$; on the contrary, we can use it further for some computations. The only qubit that is needed to be measured is that one that is used to create the linear combination of unitary $G_p + G_p^\dagger$. Thus, we state that $U_{(p)}^{(S)}$ is a $(1, 1, 0)$ block-encoding of $A_{(p)}^S$, which uses $h+1$ pure ancillas.

Theorem 6 (Diagonal block-encoding of amplitudes; modified Theorem 2 from Ref. [33]). Given an h -qubit quantum state specified by a state-preparation-unitary U_S , such that $|\psi^S\rangle^h = U_S|0\rangle^h = \sum_{j=0}^{2^h-1} \psi_j^S|j\rangle^h$ (with $\psi_j^S \in \mathbb{C}$), we can prepare a $(2, 2, 0)$ -block-encoding $U_{A'}$ of the $A' = \text{diag}(\psi_0^S, \dots, \psi_{2^h-1}^S)$ using:

- (1) 6 Applications of U_C^S ;
- (2) $528h - 463$ one-qubit operations;
- (3) $428h - 378$ C-NOTs;
- (4) $2h - 1$ pure ancillas.

Proof. Using the linear combination of unitaries $U_{(0)}^S + iU_{(1)}^S$ combining real and imaginary parts of ψ_j^S , we can construct the desired block-encoding of A' . The particular number of operations is computed from Eq. (13). Thus, we count that each control- U_C^S can be implemented using: (i) two Toffoli, (ii) one U_C^S , and (iii) one pure ancilla, where we assumed that each control one-qubit operation can be implemented using: (i) 2 one-qubit operations and (ii) 2 C-NOTs. ■

Corollary 1 (Amplitude-oracle for $\hat{x}$). Let $\hat{x}$ be a n -qubit matrix as in Eq. (4) with $\sum_{\kappa=0}^{2^n-1} (a + \kappa \Delta x)^2 = 1$. Then we can construct $(\sqrt{2}, 1, 0)$ -block-encoding $\hat{O}_x$

$$\hat{O}_x|0\rangle^1|i\rangle^n = \hat{x}_{ii}|0\rangle^1|i\rangle^n + J_x|\perp_0\rangle^{n+1},$$

with resources no greater than

- (1) $2304n^2 - 1064n - 109$ one-qubit operations;
- (2) $1864n^2 - 860n - 92$ C-NOTs;
- (3) $2n - 1$ pure ancillas.

Proof. We note that to build the controlled version of some unitary having its decomposition to one-qubit gates and C-NOTs, we turn all C-NOTs to Toffoli gates, one-qubit rotation to its controlled version (2 C-NOTs, 2 one-qubit rotations).

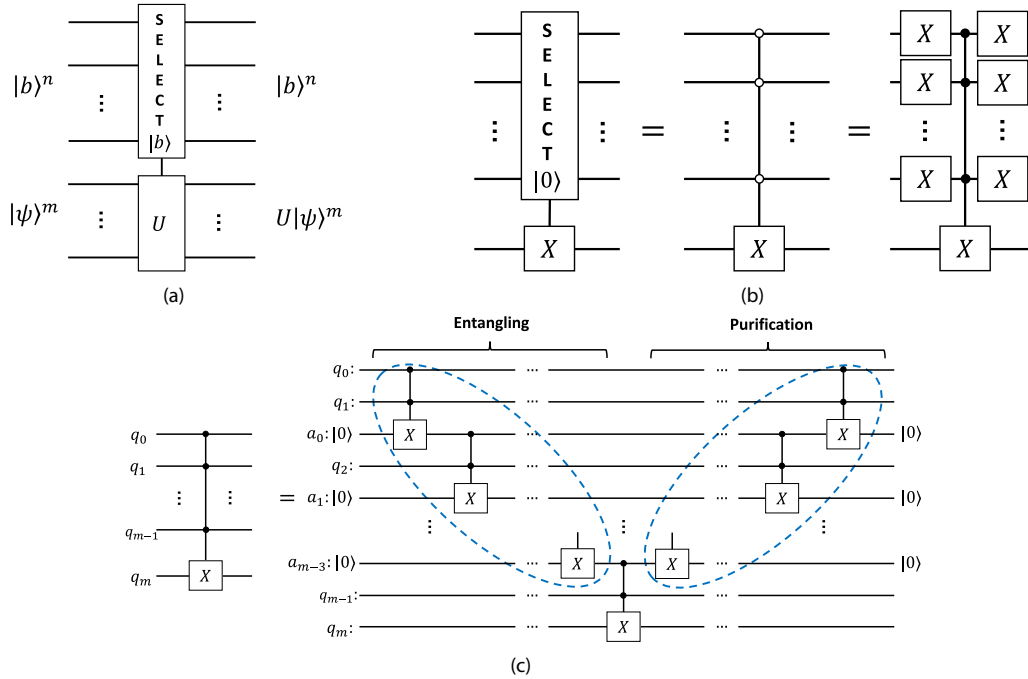


FIG. 3. (a) General view of multi-control operator C_U^b from Definition 2. Select operation means that the operator U is applied only if the state on the upper register is $|b\rangle^n$ (b) Explicit quantum circuit for implementation $C_X^{00...0}$. (c) Explicit quantum circuit for implementation $C_X^{11...1}$, which we use throughout this paper. We use $2m-3$ Toffoli application and $m-2$ pure ancillas.

For more details, see Ref. [25]. Hereafter, we always use this fact. ■

The Oracle $\hat{O}_x$ can be implemented using Theorem 6 applied to U_x from Eq. (10) and taking into account that x is real. That is why the number of additional qubits in the block-encoding is 1 against 2 in Theorem 6.

2. From x to a polynomial

In this subsection, we use the trick called qubitization [18] to transform $\hat{O}_x$ to $\hat{O}_{P_q}$. The main building block in this trick

is the so-called alternating phase modulation sequence. This unitary operation is a key instrument for building $\hat{O}_{P_q}$.

Definition 9 (Alternating phase modulation sequence).

Let U be a $(1, m, 0)$ -block-encoding of hermitian matrix A such that

$$A = (|0\rangle^m \langle 0|^m \otimes I^{\otimes n}) U (I^{\otimes n} \otimes |0\rangle^m \langle 0|^m);$$

let $\Phi \in \mathbb{R}^q$, then we define the q -phased alternating sequence U_Φ as follows:

$$U_\Phi := \begin{cases} e^{i\phi_1(2\Pi-I)} U \prod_{j=1}^{(q-1)/2} (e^{i\phi_{2j}(2\Pi-I)} U^\dagger e^{i\phi_{2j+1}(2\Pi-I)} U) & \text{if } q \text{ is odd, and} \\ \prod_{j=1}^{q/2} (e^{i\phi_{2j-1}(2\Pi-I)} U^\dagger e^{i\phi_{2j}(2\Pi-I)} U) & \text{if } q \text{ is even,} \end{cases}$$

where $2\Pi - I = (2|0\rangle^m \langle 0|^m - I^{\otimes m}) \otimes I^{\otimes n}$.

Now we apply the alternating phase modulation sequence on $\hat{O}_x$. In this case, $m = 1$, and $e^{i\phi(2\Pi-I)}$ turns to implementing the one-qubit rotation $R_z(-2\phi)$. The quantum circuit for U_Φ

for this case is depicted in Fig. 4.

$$R_z(\theta) = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix}. \quad (14)$$

Now we have all the ingredients to build the coordinate-polynomial-oracle $\hat{O}_{P_q}$ using singular value transformation.

Theorem 7 (Singular value transformation by real polynomials; Corollary 18 from Ref. [18]). Let $P_q(x)$ be a degree- q polynomial as in Definition 7 and U_Φ is the alternating phase modulation sequence as in Definition 9 based on amplitude-oracle for n -qubit matrix $\hat{x}$ as in Corollary 1. Then there exist $\Phi \in \mathbb{R}^n$, such that

$$\hat{O}_{P_q} = (H_W \otimes I^{\otimes n+1})(|0\rangle \langle 0| \otimes U_\Phi + |1\rangle \langle 1| \otimes U_{-\Phi})(H_W \otimes I^{\otimes n+1}),$$

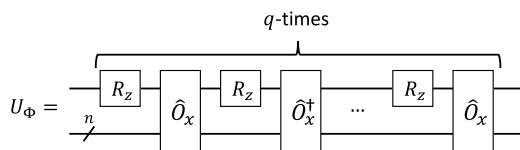


FIG. 4. Alternating phase modulation sequence U_Φ for $\hat{O}_x$ with odd q .

where $\hat{O}_{P_q}$ is coordinate-polynomial-oracle as in Definition 7

$$\hat{O}_{P_q}|0\rangle^2|i\rangle^n = (P_q(\hat{x}))_{ii}|0\rangle^2|i\rangle^n + J_{P_q}^{(i)}|\perp_0\rangle^{n+2}.$$

The total complexity of construction $\hat{O}_{P_q}$ does not exceed:

- (1) $2304qn^2 - 1064qn - 108q$ one-qubit operations;
- (2) $1864qn^2 - 860qn - 92q$ C-NOTs;
- (3) $2n - 1$ pure ancillas.

Moreover, given $P_q(x)$ and $\delta \geq 0$, we can find a $P'_q(x)$ and a corresponding Φ , such that $|P'_q(x) - P_q(x)| \leq \delta$ for all $x \in [-1, 1]$, using a classical computer in time $\mathcal{O}(\text{poly}(q, \log(1/\delta)))$.

Proof. The rule of choosing Φ and the general proof is given in Ref. [18]. We computed complexities using amplitude-oracle for $\hat{x}$ (Corollary 1) as an input for U_Φ depicted in Fig. 4. ■

B. Sparse-amplitude-oracle for momentum operator

In this subsection, we achieve another important milestone toward a block-encoding of Eq. (1), which is sparse-amplitude-oracle for the operator $\hat{p}^m$

$$\hat{O}_{p^m}^S|0\rangle^1|s\rangle^l := \frac{(\hat{p}^m)^{(s)}}{\sqrt{\mathcal{N}_{p^m}}} |0\rangle^1|s\rangle^l + \sqrt{1 - \frac{|(\hat{p}^m)^{(s)}|^2}{\mathcal{N}_{p^m}}} |1\rangle^1|s\rangle^l, \quad (15)$$

where l is logarithm of sparsity $l = \lceil \log_2(m+1) \rceil$ for $\hat{p}^m$ in the form (4); $\mathcal{N}_{p^m} \geq \|\hat{p}^m\|_{\max}^2$. We underline here that $(\hat{p}^m)^{(s)}$ does not depend on a row index since matrix $\hat{p}^m$ is sparse and banded.

Lemma 3. Let $\hat{p}$ be a $2^n \times 2^n$ matrix as in Eq. (4) then the sparse-oracle-access to the operator $\hat{p}^m$, where $2^l = m+1$ can be implemented (i) 2^l one-qubit operations, (ii) 2^l C-NOTs.

Proof. We design a quantum circuit for this operation and compute the complexity in Appendix B. We note that $\hat{p}_{ij}^m = (-1)^m \hat{p}_{ji}^m$, which means that a similar construction can be implemented for the column sparsity. ■

C. One-term block-encoding

In this section, we combine the coordinate-polynomial-oracle from Theorem 7 and sparse-amplitude-oracle for momentum operator from Lemma 3 into the auxiliary operators

$$\begin{aligned} A_{P_q P^m} |0\rangle^3 |0\rangle^n |\psi\rangle^n &= \frac{(-1)^m}{\sqrt{2^l \mathcal{N}_{p^m}}} \sum_{j=0, \dots, 2^n-1} \underbrace{P_q(\hat{x}_{ii})(\hat{p}^m)_{ij} \sigma_j |0\rangle^3 |i\rangle^n |j\rangle^n}_{(P_q(\hat{x}) \hat{p}^m)_{ij}} + J_{P_q P^m} |\perp_0\rangle^{2n+3}; \\ A_{P^m P_q} |0\rangle^3 |0\rangle^n |\psi\rangle^n &= \frac{(-1)^m}{\sqrt{2^l \mathcal{N}_{p^m}}} \sum_{j=0, \dots, 2^n-1} \underbrace{(\hat{p}^m)_{ij} P_q(\hat{x}_{jj}) \sigma_j |0\rangle^3 |i\rangle^n |j\rangle^n}_{(\hat{p}^m P_q(\hat{x}))_{ij}} + J_{P^m P_q} |\perp_0\rangle^{2n+3}; \\ |\psi\rangle^n &= \sum_{j=0}^{2^n-1} \sigma_j |j\rangle^n, \end{aligned} \quad (16)$$

where we associate the index j with columns and the index i with rows; $|\perp_0\rangle^{2n+3}$ is orthogonal to $|0\rangle^3 |\kappa\rangle^{2n}$ for any $\kappa = 0, 1, \dots, 2^{2n} - 1$; F_j^m means set of nonzero elements in j -th column of the $\hat{p}^m$. The usefulness of these operators can be seen if we apply the Hadamard operator $H_W^{\otimes n}$ on the last register, which effectively creates block-encodings of $P_q(\hat{x}) \hat{p}^m$ or $\hat{p}^m P_q(\hat{x})$.

Lemma 4 (1-term block-encoding auxiliary). Let P_q be a polynomial as in Definition 7, $\hat{x}$ and $\hat{p}$ be the n -qubit matrices as in Eq. (4), and let $m+1 = 2^l$ then operators $A_{P_q P^m}$ or $A_{P^m P_q}$ can be created using

- (1) $2304qn^2 - 1064qn + 32 \times 2^l n + 32n - 108q - 48 \times 2^l + 2^l - 48$ one-qubit operations;
- (2) $1864qn^2 - 860qn + 25 \times 2^l n + 32n - 92q - 17 \times 2^l - 48$ C-NOTs;
- (3) $2n - 1$ pure ancillas.

Proof. Figure 5 explicitly shows how to build the auxiliary unitaries. The number of operations is simply followed from the parts forming those auxiliary operations for the one-term: (i) coordinate-polynomial-oracle $\hat{O}_{P_q}$ (Theorem 7), (ii) sparse-amplitude-oracle for momentum operator $\hat{O}_{p^m}^S$ (Lemma 3), and (iii) banded-sparse-access for momentum operator $\hat{O}_{p^m}^{BS}$

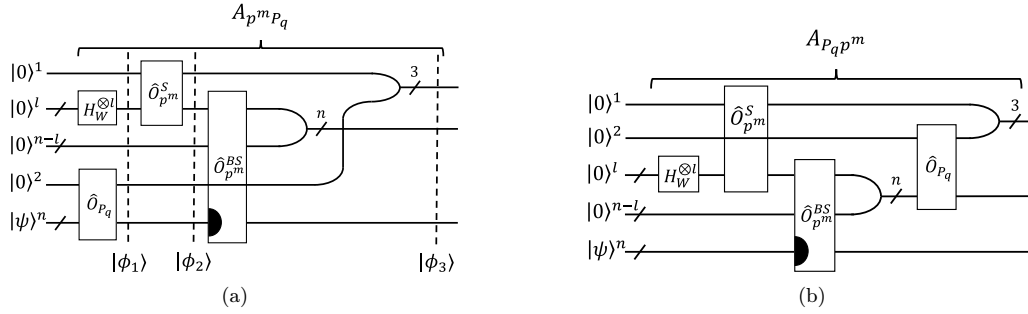


FIG. 5. Circuit design for implementing $A_{p^m p_q}$ (a) and $A_{p_q p^m}$ (b), where the last register is associated with column number; l - and $n-l$ -qubit registers are both correspond to the row number. The banded-sparse-access $\hat{O}_{p^m}^{BS}$ for p^m coincides to the one for $p_q(\hat{x})\hat{p}^m$ since the position of nonzero elements coincides. We underlined the register which does not change under action of $\hat{O}_{p^m}^{BS}$ by the filled semicircle. The Y-shaped frame means simple merging of registers without any operations. In scheme (a), $\hat{O}_{p^m}^{BS}$ does not act in the fourth register, and in scheme (b), $\hat{O}_{p^m}^{BS}$ does not act on the second one; we indicated this feature by plotting registers' lines above the boxes for operations. The fourth registers in scheme (a) change its place and merge with the first register at the end of the scheme. We clarify the circuit (a) in Eq. (17).

(Lemma 1). The following equation clarifies the circuits:

$$\begin{aligned}
 |\phi_1\rangle^{2n+3} &= \frac{1}{\sqrt{2^l}} \sum_{s=0}^{2^l-1} \sum_{j=0}^{2^n-1} \sigma_j P_q(\hat{x}_{jj}) |0\rangle^1 |s\rangle^l |0\rangle^{n-l} |0\rangle^2 |j\rangle^n + \dots \\
 |\phi_2\rangle^{2n+3} &= \frac{1}{\sqrt{2^l \mathcal{N}_{p^m}}} \sum_{s=0}^{2^l-1} \sum_{j=0}^{2^n-1} \sigma_j P_q(\hat{x}_{jj}) (\hat{p}^m)^{(s)} |0\rangle^1 |s\rangle^l |0\rangle^{n-l} |0\rangle^2 |j\rangle^n + \dots \\
 |\phi_3\rangle^{2n+3} &= \frac{1}{\sqrt{2^l \mathcal{N}_{p^m}}} \sum_{j=0, \dots, 2^n-1} \sum_{i \in F_j^{(m)}} \sigma_j P_q(\hat{x}_{jj}) (\hat{p}^m)_{ji} |0\rangle^3 |i\rangle^n |j\rangle^n + \dots \\
 &= \frac{(-1)^m}{\sqrt{2^l \mathcal{N}_{p^m}}} \sum_{j=0, \dots, 2^n-1} \sum_{i \in F_j^{(m)}} \sigma_j P_q(\hat{x}_{jj}) (\hat{p}^m)_{ij} |0\rangle^3 |i\rangle^n |j\rangle^n + \dots
 \end{aligned} \tag{17}$$

D. Block-encoding

In this section, we combine one-term block-encodings auxiliaries according the view of the Hamiltonian [Eq. (1) using the linear combination of unitary technique [35]; then we apply banded-sparse-access to the Hamiltonian to purify ancillas for row/column number. All those steps allow us to build an efficient block-encoding for the Hamiltonian. This process highlights the interplay between quantum algorithmic constructs and the physical representations of quantum systems, paving the way for advanced simulations and computations in quantum physics.

Corollary 2 (Block-encoding auxiliary). Let H be a Hamiltonian as in Eq. (1) with $\eta = 2^\gamma$, $q = \max_k q_k$, $2^{l_k} = m_k$, $l_{\max} = \max_k l_k$, and matrix representation as in Eq. (3), then the unitary operation A_H

$$\begin{aligned}
 A_H \sum_{j=0}^{2^n-1} \sigma_j |0\rangle^{\gamma+4} |0\rangle^n |j\rangle^n &= (U_\alpha^{H^T} \otimes I^{\otimes 2n+1}) \left(\prod_{k=0}^{\eta-1} C_{A_{p^{m_k} p_{q_k}}}^{k+\eta} \right) \left(\prod_{k=0}^{\eta-1} C_{A_{p_{q_k} p^{m_k}}}^k \right) [U_\alpha^H |0\rangle^{\gamma+1}] \sum_{j=0}^{2^n} \sigma_j |0\rangle^3 |0\rangle^n |j\rangle^n \\
 &= \frac{1}{\mathcal{N}_H} \sum_{i=0, \dots, 2^n-1} \sum_{j \in F_i^H} (\hat{H}_{ij} \sigma_j |0\rangle^{\gamma+4} |i\rangle^n |j\rangle^n + J_{\hat{H}_{ij}} |\perp_0\rangle^{\gamma+2n+4}); \quad \sigma_j \in \mathbb{C} \quad \forall j = 0, 1, \dots, 2^n-1 \\
 \mathcal{N}_H &= 2 \sum_{\kappa=0}^{2^\gamma-1} |\alpha_\kappa| \sqrt{2^{l_\kappa} \mathcal{N}_{p^{m_\kappa}}}
 \end{aligned}$$

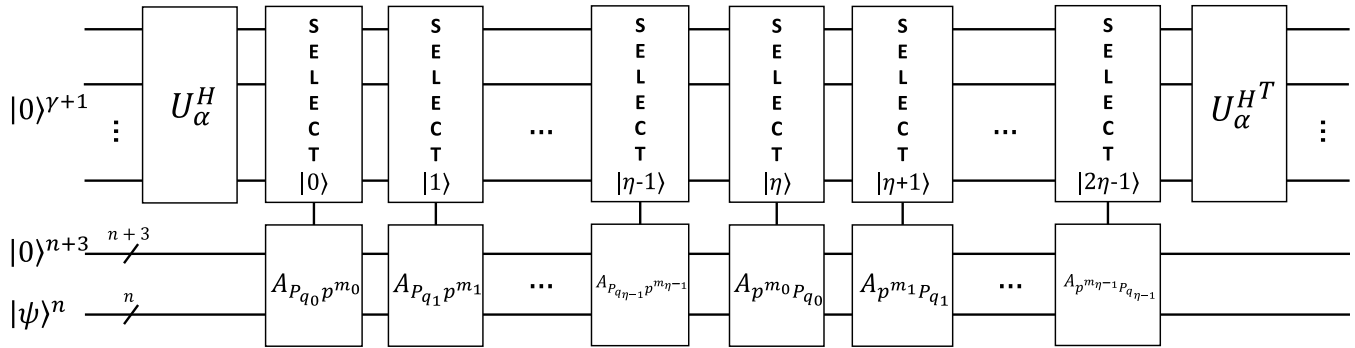


FIG. 6. Quantum circuit implementing A_H . The overall number of qubits is $2n + \gamma + 4$, where $\gamma + 1$ qubits are used to create linear superposition of unitaries. The symbol T in U_α^{HT} means the matrix transposition.

can be implemented with resources no greater than:

- (1) $2^{\gamma+2}[9760n^2q + 33 \times 2^{l_{\max}+2}n + 160n + 8\gamma - 4504nq - 476q - 115 \times 2^{l_{\max}} - 239]$ one-qubit operations;
- (2) $2^{\gamma+1}[15792n^2q + 107 \times 2^{l_{\max}+1} + 256n + 12\gamma - 7288nq - 768q - 49 \times 2^{l_{\max}+1} - 382]$ C-NOTs;
- (3) $2n + \gamma - 1$ pure ancillas.

The state $|\perp_0\rangle^{\gamma+2n+4}$ is orthogonal to $|0\rangle^{\gamma+4} \otimes I^{\otimes 2n}$; the symbol T in U_α^{HT} means the matrix transposition. The procedure exploits the linear combination theorem [18,35] applied on $A_{P_{q_0} p^{m_0}}, A_{P_{q_1} p^{m_1}}$ from Lemma 4. The quantum circuit for the A_H is depicted in Fig. 6. We suppose that a value $J_{\hat{H}_{ij}}$ and explicit view of $|\perp_0\rangle^{\gamma+2n+4}$ are unimportant for the reader, so we omit it; however, they can be retrieved from the Fig. 6 and explicit views of the operators.

Clarification. As a subroutine, the implementation of A_H uses U_α^H

$$U_\alpha^H |0\rangle^{\gamma+1} = \frac{1}{\sqrt{\mathcal{N}_H}} \sum_{k=0}^{2^\gamma-1} (2^{l_k} \mathcal{N}_p^{m_k})^{1/4} [\sqrt{i^{m_k} \alpha_k} |k\rangle^{\gamma+1} + \sqrt{i^{m_k} \alpha_k^*} |k + 2^\gamma\rangle^{\gamma+1}],$$

where i^{m_k} corresponds to the product of the omitted global phase $(-i)^{m_k}$ discussed in Appendix B and to $(-1)^{m_k}$ from the $A_{P_{q_0} p^{m_0}}, A_{P_{q_1} p^{m_1}}$; the $\sqrt{\alpha}$ is the principle square root. Since α_k can be any complex number in principle, the construction of U_α^H addresses the general problem of construction an arbitrary quantum state. The authors of Refs. [36,37] suggest an explicit implementation of U_α^H that consists of less than $\frac{23}{24} 2^{\gamma+1}$ C-NOTs for even numbers of qubits and $2^{\gamma+1}$ C-NOTs for odd numbers. Let us consider the number of one-qubit operations here. Naively, one can conclude that each C-NOT can be accompanied by six Euler rotations; however, a rotation around the z -axis commutes with control operation, and a rotation around an x one commutes with Pauli X operation. Thus, the number of one-qubit Euler rotations cannot exceed the number of C-NOTs by more than four times [38]. For simplicity, in this paper we address a general one-qubit operation, so we merge two Euler rotations acting on each two qubits where the C-NOT is applied. Thus, the complexity of the U_α^H implementation is less than: (i) $2^{\gamma+2}$ one-qubit rotations, (ii) $2^{\gamma+1}$ C-NOTs.

Now we estimate the sparsity of $\hat{H}$ from Eq. (1). Using binomial theorem [39] applied to $\hat{p}^m$, it can be proved

that the sparsity of the Hamiltonian matrix is no greater than $\max_k 2m_k + 1$; we assume that the sparsity value is 2^l . Thus, using the quantum circuits from Appendix A, we can efficiently implement the quantum circuit for the unitary Hamiltonian banded-sparse access $\hat{O}_H^{BS}$.

Theorem 8 (Hamiltonian Block-encoding). Let H be a Hamiltonian as in Eq. (1), with the number of terms $\eta = 2^\gamma$, $2^l = \max_k 2m_k + 1$, $q = \max_k q_k$, and with matrix representation $\hat{H}$ as in Eq. (3), with normalization constant $\mathcal{N}_H$ (see Corollary 2); then we can implement a $(\sqrt{2^l} \mathcal{N}_H, l + \gamma + 4, 0)$ -block-encoding of n -qubit matrix $\hat{H}$

$$U_H |0\rangle^{l+4} |\phi\rangle^n = \frac{1}{\sqrt{2^l \mathcal{N}_H}} \sum_{i=0, \dots, 2^n-1} \sum_{j \in F_i^H} \times (\hat{H}_{ij} \sigma_j |0\rangle^{l+4} |i\rangle^n + J_{\hat{H}_{ij}}^U |\perp_0\rangle^{n+l+4});$$

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

with resources no greater than:

- (1) $2^{\gamma+2}[9760n^2q + 33 \times 2^{l_{\max}+2}n + 160n + 8\gamma - 4504nq - 476q - 115 \times 2^{l_{\max}} - 239] + 2^{l+5}n - 3 \times 2^{l+4} + 32n + l - 48$ one-qubit operations;
- (2) $2^{\gamma+1}[15792n^2q + 107 \times 2^{l_{\max}+1} + 256n + 12\gamma - 7288nq - 768q - 49 \times 2^{l_{\max}+1} - 383] + 25 \times 2^l n - 36 \times 2^l + 32n - 48$ C-NOTs;
- (3) $3n + \gamma - l - 1$ pure ancillas.

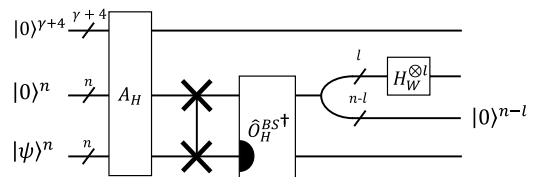


FIG. 7. The explicit design of quantum circuit for block-encoding of the Hamiltonian (3). The unitary $\hat{O}_H^{BS\dagger}$ is a Hermit conjugated $\hat{O}_H^{BS}$, see Appendix A. Again, the filled semicircle determines which register does not change under $\hat{O}_H^{BS}$ and $\hat{O}_H^{BS\dagger}$. Note that according to Definition 6, $\hat{O}_H^{BS} = \hat{O}_{H^T}^{BS}$, that is why we omitted the transpose notation for $\hat{O}_H^{BS\dagger}$. We indicated pure ancillas by showing that after the action the register's state becomes $|0\rangle^{n-l}$.

Proof. As a proof, we simply present Fig. 7, which explicitly shows how to construct the desired block-encoding using the auxiliary construction from Corollary 2. We exploit banded-sparse-access to the Hamiltonian in Fig. 7, which we built using the technique described in Appendix A. As an input for the algorithm we need a set of nonzero indexes in the first row of the Hamiltonian. This knowledge is given by degrees of the momentum operator contributing to Eq. (1). Using the periodic spatial boundaries [Eq. (4)] the sparsity of the Hamiltonian is no greater than $\max_k 2m_k + 1$. ■

E. Evolution operator creation

In this subsection, we present one of the promising ways to exploit the efficient block-encoding declared in Theorem 8. Now we apply Theorem 3 to the designed block-encoding.

Theorem 9 (Nearly-optimal Hamiltonian simulation).

Let $t \in \mathbb{R}/\{0\}$, $\epsilon \in (0, 1)$ and let U_H be an $(\sqrt{2}^l \mathcal{N}_H, l + \gamma + 4, 0)$ -block-encoding of the $2^n \times 2^n$ Hamiltonian H as in Theorem 8. Then we can implement an ϵ -precise Hamiltonian simulation unitary V , which is an $(2, l + \gamma + 6, \epsilon)$ -block-encoding of e^{-itH} , with resources no greater than:

$$(1) \ 305 \times 2^{7+\gamma} n^2 q \Omega(\alpha t, \epsilon) + 33 \times 2^{4+l+\gamma} n \Omega(\alpha t, \epsilon) + 5 \times 2^{7+\gamma} n \Omega(\alpha t, \epsilon) + 2^{5+l} n \Omega(\alpha t, \epsilon) + 32 n \Omega(\alpha t, \epsilon) + 17 l \Omega(\alpha t, \epsilon) + 66 \Omega(\alpha t, \epsilon) + 2^{5+\gamma} \gamma \Omega(\alpha t, \epsilon) + 35 \times 2^{7+l+\gamma} n + 21 \times 2^{8+\gamma} n + 33 \times 2^{3+l} n + 320 n + 323 \times 2^{10+\gamma} n^2 q + 2l - 563 \times 2^{5+\gamma} n q \Omega(\alpha t, \epsilon) - 119 \times$$

$$2^{4+\gamma} q \Omega(\alpha t, \epsilon) - 115 \times 2^{2+l+\gamma} \Omega(\alpha t, \epsilon) - 239 \times 2^{2+\gamma} \Omega(\alpha t, \epsilon) - 3 \times 2^{4+l} \Omega(\alpha t, \epsilon) - 2385 \times 2^{6+\gamma} n q - 503 \times 2^{5+\gamma} q - 507 \times 2^{3+l+\gamma} - 1005 \times 2^{3+\gamma} - 3 \times 2^{5+l} - 476$$

$$(2) \ 987 \times 2^{5+\gamma} n^2 q \Omega(\alpha t, \epsilon) + 107 \times 2^{2+l+\gamma} n \Omega(\alpha t, \epsilon) + 2^{9+\gamma} n \Omega(\alpha t, \epsilon) + 25 \times 2^l n \Omega(\alpha t, \epsilon) + 32 n \Omega(\alpha t, \epsilon) + 12 l \Omega(\alpha t, \epsilon) + 38 \Omega(\alpha t, \epsilon) + 3 \times 2^{3+\gamma} \gamma \Omega(\alpha t, \epsilon) + 453 \times 2^{3+l+\gamma} n + 17 \times 2^{8+\gamma} n + 107 \times 2^{1+l} n + 256 n + 4181 \times 2^{6+\gamma} n^2 q + 2l - 911 \times 2^{4+\gamma} n q \Omega(\alpha t, \epsilon) - 3 \times 2^{9+\gamma} q \Omega(\alpha t, \epsilon) - 49 \times 2^{3+l+\gamma} \Omega(\alpha t, \epsilon) - 383 \times 2^{1+\gamma} \Omega(\alpha t, \epsilon) - 3859 \times 2^{5+\gamma} n q - 407 \times 2^{5+\gamma} q - 409 \times 2^{3+l+\gamma} - 1627 \times 2^{2+\gamma} - 3 \times 2^{5+l} - 384 \text{ C-NOTs};$$

(3) $3n + \gamma - l - 1$ pure ancillas;

where we introduced $\alpha = 2^{l+1} \mathcal{N}_H$ for convenience. The coefficient $\Omega(\alpha t, \epsilon)$ is implicitly defined by the condition $g \geq \Omega(\alpha t, \epsilon)$ and can be retrieved from

$$\frac{1.07}{\sqrt{g}} \left(\frac{\alpha t \epsilon}{2g} \right)^g \leq \epsilon.$$

The parameter has scaling

$$\Omega(\alpha t, \epsilon) = \mathcal{O} \left(\alpha t + \frac{\ln(1/\epsilon)}{\ln(e + \frac{\ln(1/\epsilon)}{\alpha t})} \right).$$

Proof. The proof is well described in Ref. [18], where the authors provide a decomposition of the desired evolution operator to multicontrolled Pauli rotations R_z and applications of the block-encoding that is very similar to one described

TABLE III. Notations for a multidimensional case.

Notation	Name	Formula	First app.
$H^{(d)}$	Multidimensional Hamiltonian	$\sum_{k=0}^{n_d-1} \alpha_k \prod_{y=0}^{d-1} \otimes R(L_{ky}, P_{q_k^{(y)}}(x_y), P_{y_k^{(y)}})$	Eq. (18)
R	Auxiliary functions for permutations	$R(0, P_{q_k^{(y)}}(x_y), P_{y_k^{(y)}}) = P_{q_k^{(y)}}(x_y) P_{y_k^{(y)}};$ $R(1, P_{q_k^{(y)}}(x_y), P_{y_k^{(y)}}) = P_{y_k^{(y)}} P_{q_k^{(y)}}(x_y)$	Eq. (18)
$A_{q_k m_k}^{(L_k)}$	Multidimensional one-term block-encoding auxiliary	$\frac{A_{q_k m_k}^{(L_k)}}{\prod_{\sigma=0}^{d-1} \mathcal{N}_{m_k(\sigma)}} \sum_{j_0, \dots, j_{n_d-1}} \alpha_{j_0, \dots, j_{n_d-1}} (\prod_{y=0}^{d-1} \otimes 0\rangle^3 0\rangle^{n_0} j_0\rangle^{n_0}) =$ $\frac{1}{\prod_{\sigma=0}^{d-1} \mathcal{N}_{m_k(\sigma)}} \sum_{\substack{i_0=0, \dots, 2^{n_0}-1 \\ j_0 \in F_{i_0}^0}} \sum_{\substack{i_1=0, \dots, 2^{n_1}-1 \\ j_1 \in F_{i_1}^1}} \dots \sum_{\substack{i_{d-1}=0, \dots, 2^{n_{d-1}}-1 \\ j_{d-1} \in F_{i_{d-1}}^{d-1}}} \alpha_{j_0, \dots, j_{n_d-1}}$ $[\prod_{y=0}^{d-1} R(L_{ky}, P_{q_k^{(y)}}(\hat{x}_y), \hat{P}_{y_k^{(y)}})_{i_y j_y}] \prod_{y=0}^{d-1} \otimes 0\rangle^3 i_y\rangle^{n_y} j_y\rangle^{n_y} + \dots$	Eq. (19)
$\mathcal{N}_{m_k(\sigma)}$	Normalization constant for $A_{P_{q_k^{(y)}} P_{\sigma^k} m_k^{(\sigma)}}$	$\sqrt{2^{l_k(\sigma)} \mathcal{N}_{P_{m_k(\sigma)}}}$	Eq. (19)
F_i^y	Set of nonzero elements in the i -th row of $\hat{P}_{m_k^{(y)}}$	None	Eq. (19)
$A_H^{(d)}$	Multidimensional Hamiltonian block-encoding auxiliary	$A_H^{(d)} 0\rangle^y (\sum_{j_0, \dots, j_{d-1}} \phi_{j_0, \dots, j_{d-1}} \prod_{y=0}^{d-1} \otimes 0\rangle^3 0\rangle^{n_y} j_y\rangle^{n_y}) =$ $\frac{1}{\mathcal{N}_H^{(d)}} \sum_{i_0, j_0=0}^{2^{n_0}-1} \sum_{i_1, j_1=0}^{2^{n_1}-1} \dots \sum_{i_{d-1}, j_{d-1}=0}^{2^{n_{d-1}}-1} \hat{H}_{i_0 i_1 \dots i_{d-1} j_0 j_1 \dots j_{d-1}}^{(d)} 0\rangle^{y+3d}$ $\hat{H}_{i_0 i_1 \dots i_{d-1} j_0 j_1 \dots j_{d-1}}^{(d)} \prod_{y=0}^{d-1} \otimes i_y\rangle^{n_y} j_y\rangle^{n_y} + \dots$	Cor. 3
$\mathcal{N}_H^{(d)}$	Normalization constant for $A_H^{(d)}$	$\sum_{k=0}^{n_d-1} \alpha_k \prod_{\sigma=0}^{d-1} \mathcal{N}_{m_k(\sigma)}$	Cor. 3

TABLE IV. Circuit's complexity in multidimension. Here, n_y denotes the number of qubits register encoding row and column numbers in y -th dimension in Eq. (18). $q_k^{(y)}$ and $m_k^{(y)}$ are the exponent of coordinate operator and momentum operator in y -th dimension x_y and p_y in Eq. (18).

Name	Notation	Scaling	Ancillas	Main Thm.
Multidimensional Hamiltonian block-encoding auxiliary	$A_H^{(d)}$	$\mathcal{O}(\sum_{k=0}^{\eta_d-1} \sum_{y=0}^{d-1} (q_k^{(y)} n_y^2 + m_k^{(y)} n_y))$	$\mathcal{O}(\sum_{y=0}^d n_y)$	Cor. 3
Multidimensional block-encoding of Hamiltonian	$H^{(d)}$	$\mathcal{O}(\sum_{k=0}^{\eta_d-1} \sum_{y=0}^{d-1} (q_k^{(y)} n_y^2 + m_k^{(y)} n_y))$	$\mathcal{O}(\sum_{y=0}^d n_y + \log(\eta))$	Thm. 10

in Theorem 7. We provided a certain number of gates for our case based on the quantum schemes from the paper (Theorem 8).

Thus, we prove that the nearly optimal Hamiltonian simulation is achievable by providing a step-by-step guide for how to construct an efficient block-encoding and the evolution operator for the Hamiltonian with form (3). Moreover, we claim that the considered case is a basis for a more general algorithm. That is why, in next section, we generalize our approach to different boundaries and finite-difference schemes. In addition, we show how our approach can be straightforwardly extended to a multidimensional case (several qumodes). These techniques together allow us to build a general toolbox for constructing block-encodings.

IV. METHOD GENERALIZATION

The robustness of a proposed computational method is often gauged by its flexibility and the breadth of its applicability. In this section, we explore how the algorithm described in Sec. III can be straightforwardly extended to the multidimensional case. In this subsection, we clarify the implementation of each step from the road map (see Fig. 2) for the multidimensional case.

Suppose we have d -qumodes model:

$$H^{(d)} = \sum_{k=0}^{\eta_d-1} \alpha_k \prod_{y=0}^{d-1} \otimes R(L_{ky}, P_{q_k^{(y)}}(x_y), p_y^{m_k^{(y)}});$$

$$R(0, P_{q_k^{(y)}}(x_y), p_y^{m_k^{(y)}}) = P_{q_k^{(y)}}(x_y) p_y^{m_k^{(y)}};$$

$$R(1, P_{q_k^{(y)}}(x_y), p_y^{m_k^{(y)}}) = p_y^{m_k^{(y)}} P_{q_k^{(y)}}(x_y), \quad (18)$$

where x_y/p_y is the coordinate/momentum operator acting on y -qumode; we introduce set L and function R to go through possible permutations of P_q and p^m for each dimension. We assume that the coefficients α and L are such that $H^{(d)}$ is Hermitian. For each spatial dimension, we introduce two n_y -qubit register encoding row and column numbers. Similarly, we introduce finite-difference representation of coordinate and momentum operators $\hat{x}$, $\hat{p}$, as in Eq. (4).

Now let us consider each term of Eq. (18) independently, as in Sec. III. For an individual term, we can build a block-encoding auxiliary from Lemma 4 independently acting on each spatial dimension register:

$$A_{q_k m_k}^{(L_k)} \sum_{j_0, \dots, j_{n_d-1}} \sigma_{j_0, \dots, j_{n_d-1}} (|0\rangle^3 |0\rangle^{n_0} |j_0\rangle^{n_0} \dots |0\rangle^3 |0\rangle^{n_{d-1}} |j_{d-1}\rangle^{n_{d-1}})$$

$$= \sum_{j_0, \dots, j_{n_d-1}} \sigma_{j_0, \dots, j_{n_d-1}} \prod_{y=0}^{d-1} \otimes \left[A_{R(L_{ky}, P_{q_k^{(y)}}(x_y), p_y^{m_k^{(y)}})} |0\rangle^3 |0\rangle^{n_y} |j_y\rangle^{n_y} \right]$$

$$= \frac{1}{\prod_{\sigma=0}^{d-1} \mathcal{N}_{m_k(\sigma)}} \sum_{\substack{i_0=0, \dots, 2^{n_0}-1 \\ j_0 \in F_{i_0}^0}} \sum_{\substack{i_1=0, \dots, 2^{n_1}-1 \\ j_1 \in F_{i_1}^1}} \dots \sum_{\substack{i_{d-1}=0, \dots, 2^{n_{d-1}}-1 \\ j_{d-1} \in F_{i_{d-1}}^{d-1}}} \left[\prod_{y=0}^{d-1} R(L_{ky}, P_{q_k^{(y)}}(\hat{x}_y), \hat{p}_y^{m_k^{(y)}})_{i_y j_y} \right]$$

$$\times \sigma_{j_0, \dots, j_{n_d-1}} \prod_{y=0}^{d-1} \otimes |0\rangle^3 |i_y\rangle^{n_y} |j_y\rangle^{n_y} + \dots, \quad (19)$$

where F_i^y indicates a set of nonzero elements in the i -th row of $\hat{p}_k^{(y)}$; the normalization constant $\mathcal{N}_{m_k(\sigma)} = \sqrt{2^{l_k(\sigma)} \mathcal{N}_{p_{m_k(\sigma)}}}$ for certain and $m_k(\sigma)$ coincides to one in Eq. (16). In Eq. (19), we omitted the orthogonal vectors for simplicity, so the symbol "... " conceals some orthogonal states. Equation (19) shows that the terminal wave function is factorized. Similarly to the one-dimensional case, combining the superposition unitaries

$$\begin{aligned}
& A_H^{(d)} |0\rangle^\gamma \left(\sum_{j_0, \dots, j_{d-1}} \phi_{j_0, \dots, j_1} \prod_{y=0}^{d-1} \otimes |0\rangle^3 |0\rangle^{n_y} |j_y\rangle^{n_y} \right) \\
&= (U_\alpha^{H^{(d)T}} \otimes I^{\otimes \sum_{y=0}^{d-1} (2^{n_y+1})}) \left(\prod_{k=0}^{\eta_d-1} C_{A_{q_k m_k}}^{(L_k)} \right) [U_\alpha^{H^{(d)}} |0\rangle^\gamma] \left(\sum_{j_0, \dots, j_{d-1}} \phi_{j_0, \dots, j_1} \prod_{y=0}^{d-1} \otimes |0\rangle^3 |0\rangle^{n_y} |j_y\rangle^{n_y} \right) \\
&= \frac{1}{\mathcal{N}_H^{(d)}} \sum_{i_0=0}^{2^{n_0}-1} \sum_{i_1=0}^{2^{n_1}-1} \dots \sum_{i_{d-1}=0}^{2^{n_{d-1}}-1} \sum_{j_0, j_1, \dots, j_{d-1} \in F_{i_0, i_1, \dots, i_{d-1}}^H} \hat{H}_{i_0 i_1 \dots i_{d-1} j_0 j_1 \dots j_{d-1}}^{(d)} |0\rangle^\gamma \phi_{j_0, \dots, j_1} \prod_{y=0}^{d-1} \otimes |0\rangle^3 |i_y\rangle^{n_y} |j_y\rangle^{n_y} + \dots \\
&\rightarrow \frac{1}{\mathcal{N}_H^{(d)}} \sum_{i_0=0}^{2^{n_0}-1} \sum_{i_1=0}^{2^{n_1}-1} \dots \sum_{i_{d-1}=0}^{2^{n_{d-1}}-1} \sum_{j_0, j_1, \dots, j_{d-1} \in F_{i_0, i_1, \dots, i_{d-1}}^H} \hat{H}_{i_0 i_1 \dots i_{d-1} j_0 j_1 \dots j_{d-1}}^{(d)} |0\rangle^{\gamma+3d} \phi_{j_0, \dots, j_1} \prod_{y=0}^{d-1} \otimes |i_y\rangle^{n_y} |j_y\rangle^{n_y} + \dots \\
&= \frac{1}{\mathcal{N}_H^{(d)}} \sum_{i_0, j_0=0}^{2^{n_0}-1} \sum_{i_1, j_1=0}^{2^{n_1}-1} \dots \sum_{i_{d-1}, j_{d-1}=0}^{2^{n_{d-1}}-1} \hat{H}_{i_0 i_1 \dots i_{d-1} j_0 j_1 \dots j_{d-1}}^{(d)} |0\rangle^{\gamma+3d} \phi_{j_0, \dots, j_1} \prod_{y=0}^{d-1} \otimes |i_y\rangle^{n_y} |j_y\rangle^{n_y} + \dots; \quad \mathcal{N}_H^{(d)} = \sum_{k=0}^{\eta_d-1} |\alpha_k| \prod_{\sigma=0}^{d-1} \mathcal{N}_{m_k(\sigma)},
\end{aligned}$$

with $\mathcal{O}(\sum_{k=0}^{\eta_d-1} \sum_{y=0}^{d-1} (q_k^{(y)} n_y^2 + m_k^{(y)} n_y))$ gates and $\mathcal{O}(\sum_{y=0}^d n_y)$ pure ancillas. We use the notation $F_{i_0, i_1, \dots, i_{d-1}}^H$ for a set of nonzero indexes of nonzero elements in the row with index $i_0, i_1, \dots, i_{d-1}$ of the Hamiltonian $H^{(d)}$. This underlines the entanglement between spatial dimensions registers which arises here. However, we stress that the other elements of the Hamiltonian are still zeros.

The orthogonal state is omitted for simplicity; however, it can be accurately retrieved if it is needed. Similarly, the procedure exploits the linear combination theorem applied on $A_{q_k m_k}^{(d)}$. The quantum circuit for the $A_H^{(d)}$ is very similar to one depicted in Fig. 6. The auxiliary unitary $U_\alpha^{H^{(d)}}$ is given by

$$U_\alpha^{H^{(d)}} |0\rangle^\gamma = \frac{1}{\sqrt{\mathcal{N}_H^{(d)}}} \sum_{k=0}^{2^\gamma-1} \sqrt{i^{\sum_{\sigma} m_k(\sigma)} \alpha_k \left(\prod_{\sigma=0}^{d-1} \mathcal{N}_{m_k(\sigma)} \right)} |k\rangle^\gamma.$$

The next step from our road map (see Fig. 2) is to create the block-encoding for the Hamiltonian [Eq. (18)], which can be fulfilled in an analogous way to the Theorem 8.

The final step is to build a block-encoding for the multidimensional Hamiltonian $H^{(d)}$, which is a nontrivial step in this generalization to a multidimensional case since we need to consider the multidimensional sparsity encoded by $F_{i_0, i_1, \dots, i_{d-1}}^H$. In particular, to realize the scheme as in Fig. 7, we should build a multidimensional banded-sparse-access. The view of Eq. (18) and the matrix representation of momentum operator $\hat{p}_y$ allow us to accurately retrieve the set $F_{i_0, i_1, \dots, i_{d-1}}^H$. Thus, all the ingredients for the multidimensional block-encoding are prepared.

[Eq. (19)], we suggest a way to implement the multidimensional Hamiltonian block-encoding auxiliary (similar to Corollary 2).

Corollary 3 (Multi-dimensional Hamiltonian block-encoding auxiliary). Let $H^{(d)}$ be a Hamiltonian as in Eq. (18) with the number of terms $\eta_d = 2^\gamma$, matrix representation $\hat{H}^{(d)}$, and the set L then the unitary operation $A_H^{(d)}$:

Theorem 10 (Multi-dimensional block-encoding). Let $H^{(d)}$ be a Hamiltonian as in Eq. (5) with the number of terms $\eta_d = 2^\gamma$, sparsity 2^l , $n = \sum_y n_y$ -qubit matrix representation $\hat{H}^{(d)}$, and normalization constant $\mathcal{N}_H^{(d)}$ (see Corollary 3), then we can implement a $(\sqrt{2^l} \mathcal{N}_H^{(d)}, l + \gamma + 3d, 0)$ -block-encoding of $\hat{H}$ with $\mathcal{O}(\sum_{k=0}^{\eta_d-1} \sum_{y=0}^{d-1} (q_k^{(y)} n_y^2 + m_k^{(y)} n_y))$ gates and $\mathcal{O}(\sum_{y=0}^d n_y + \log(\eta))$ pure ancillas.

Proof. As a proof we address Fig. 7, which precisely describes the quantum circuit for the multidimensional case if $\hat{O}_H^{BS}$, A_H are substituted by the multidimensional analogues $\hat{O}_{H^{(d)}}^{BS}$, $A_H^{(d)}$. Both were described earlier in this subsection.

As demonstrated, the batch of techniques described in Sec. III combined together allows us to extend our approach for a wide class of boundary conditions, finite-difference schemes, and number of dimensions. Now, using the multidimensional block-encoding as an input for Theorem 3, we suggest constructing a straightforward quantum circuit for the evolution operator.

V. CONCLUSION

In this paper, we have advanced the methodology for explicit gate construction for block-encoding of Hamiltonians, crucial for quantum simulations of linear partial differential equations. Leveraging the foundational concept of block-encoding, our approach refines its application to enhance both theoretical and practical aspects of quantum simulations by providing an algorithm that is superior compared with classical state-of-the-art analogues for solving partial differential equations.

The core of our contribution is the development of an efficient gate construction strategy for block-encoding of

$$H^{(d)} = \sum_{k=0}^{\eta d-1} \alpha_k \prod_{y=0}^{d-1} \otimes R(L_{ky}, P_{q_k^{(y)}}(x_y), p_y^{m_k^{(y)}});$$

$$R(0, P_{q_k^{(y)}}(x_y), p_y^{m_k^{(y)}}) = P_{q_k^{(y)}}(x_y) p_y^{m_k^{(y)}};$$

$$R(1, P_{q_k^{(y)}}(x_y), p_y^{m_k^{(y)}}) = p_y^{m_k^{(y)}} P_{q_k^{(y)}}(x_y), \quad (20)$$

which demonstrates the potential for significant computational speedups over traditional classical finite-difference methods in terms of solving a wide class of linear partial differential equations with coefficients that are polynomial functions. By detailing the construction of these encodings with efficient one- and two-qubit operations, we provide a practical blueprint for implementing quantum simulations on currently emerging digital quantum hardware.

This work not only fills an important gap by providing an efficient way for practical implementations but also sets a foundation for further research into optimizing these methods for a wider range of Hamiltonians. As future work, we aim to extend our approach to a wider class of numerical schemes and extend the applicability to wider classes of boundary conditions. We will also examine important partial differential equations individually.

VI. TABLES

Here we present our table of notations for both the one-dimensional and multidimensional cases. For convenience, we also present a table summarizing the exact one- and two-qubit count for each part involved in our protocol.

ACKNOWLEDGMENTS

N.L. acknowledges funding from the Science and Technology Commission of Shanghai Municipality (STCSM) Grant No. 24LZ1401200 (21JC1402900), by NSFC Grants No. 12471411 and No. 12341104, the Shanghai Jiao Tong University 2030 Initiative, the Shanghai Science and Technology Innovation Action Plan (24LZ1401200), and the Fundamental Research Funds for the Central Universities. N.G. acknowledges funding from NSFC Grant No. W2442002.

APPENDIX A: EXPLICIT QUANTUM CIRCUIT CONSTRUCTION FOR $\hat{O}_A^{BS}$

Let a 2^l sparse matrix A be as in Definition 6; so, the nonzero elements follows

$$\hat{O}_A^{BS} |0\rangle^{n-l} |s\rangle^l |i\rangle^n = |r_{s0} + i\rangle^n |i\rangle^n, \quad (A1)$$

where sum $r_{s0} + i$ means an addition modulo 2^n . So, we decompose $\hat{O}_A^{BS}$ into a product of two unitary operators $\hat{O}_A^{BS} = U^{SUM} (U_A^{(l)} \otimes \hat{I}^n)$, where U^{SUM} performs addition modulo 2^n

$$U^{SUM} |i\rangle^n |j\rangle^n = |i + j \bmod 2^n\rangle^n |j\rangle^n, \quad (A2)$$

and operator $U_A^{(l)}$ transforming a banded sparse index into a column index

$$U_A^{(l)} |0\rangle^{n-l} |s\rangle^l = |r_{s0}\rangle^n. \quad (A3)$$

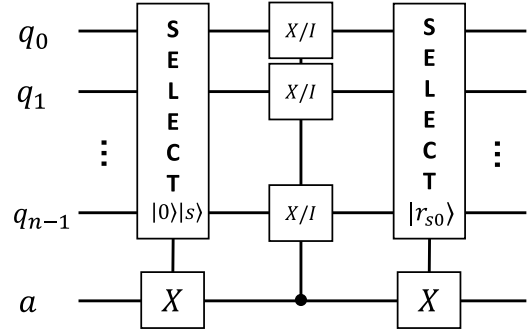


FIG. 8. Quantum circuit implementing $U_s^{r_{s0}}$. The first layer set the initialized with zero-state ancilla to $|1\rangle$ for state $|0\rangle^{n-l} \otimes |s\rangle^l$ in the first register. The second layer operation make state $|r_{s0}\rangle^n$ from $|0\rangle^{n-l} \otimes |s\rangle^l$ by choosing Pauli X or I according to bit-difference between s and r_{s0} . The third layer unentangle ancilla, making it pure.

We underline that the information about matrix A is contained only in the $U_A^{(l)}$; so we omitted the corresponding index for the U^{SUM} . Thus, we combine those two operators to build the banded-sparse-access

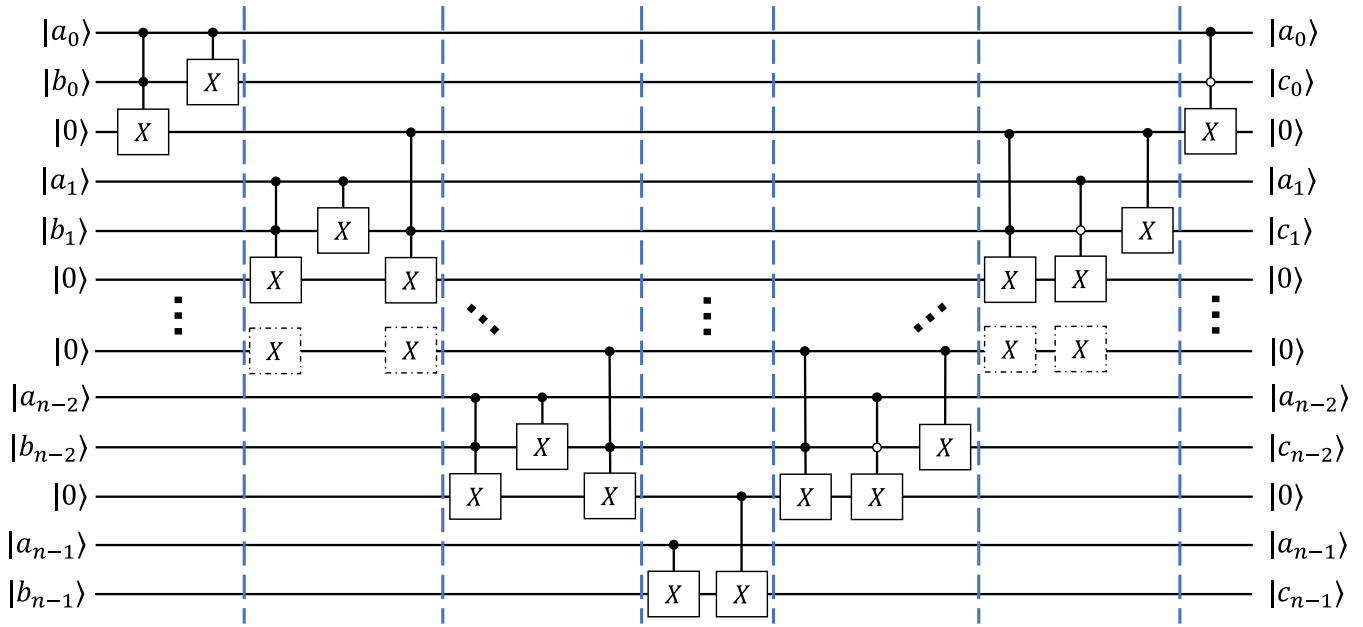
$$\begin{aligned} \hat{O}_A^{BS} |0\rangle^{n-l} |s\rangle^l |i\rangle^n &= U^{SUM} (U_A^{(l)} |0\rangle^{n-l} |s\rangle^l \otimes \hat{I}^n |i\rangle^n) \\ &= U^{SUM} |r_{s0}\rangle^n |i\rangle^n = |r_{s0} + i\rangle^n |i\rangle^n. \end{aligned} \quad (A4)$$

Now we demonstrate how $U_A^{(l)}$ can be constructed using one- and two-qubit basic quantum computer operations. First, we further decompose the operator $U_A^{(l)}$ into a product

$$U_A^{(l)} = \prod_{s=0}^{2^l-1} U_s^{r_{s0}}; \quad U_s^{r_{s0}} := |r_{s0}\rangle^n (\langle 0\rangle^{n-l} \otimes |s\rangle^l) + \sum_{j \neq s} |j\rangle^n \langle j|^n. \quad (A5)$$

The quantum circuit depicted in the Fig. 8 shows the explicit construction of $U_s^{r_{s0}}$, where each of the multicontrolled- X we implement as in Fig. 3. Thus, the overall resources for $U_A^{(l)}$ implementation no greater than: (i) $n - 1$ pure ancillas, (ii) $2^l(32n - 48)$ one-qubit operations, and (iii) $2^l(25n - 36)$ C-NOTs.

Now, let us discuss how to implement U^{SUM} , which adds the row number i to the r_{s0} . In Fig. 9, we explicitly demonstrate how such a circuit can be built. We slightly modify the adder from Ref. [40] to meet our needs. From the quantum scheme, it can be concluded that resources for the addition is no more than: (i) $n - 1$ pure ancillas, (ii) $32n - 48$ one-qubit operations, and (iii) $26n - 37$ C-NOTs. Although the final circuit depth is linear, which is not optimal, there exists a logarithmic-depth ($\mathcal{O}(\log n)$) algorithm according to Ref. [41], albeit requiring more gates (up to $\approx 124n$ C-NOTs). The choice of algorithm depends on hardware specifics. However, in this paper, we opt for the circuit in Fig. 9 due to its minimal gate and ancilla requirements.

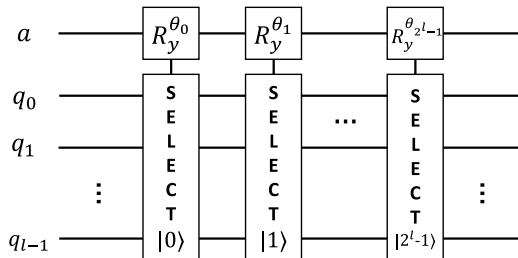
FIG. 9. Quantum circuit implementing U^{SUM} , which adds two binary strings: $a + b = c \pmod{2^n}$.

APPENDIX B: QUANTUM CIRCUIT DESIGN FOR $\hat{O}_{p^m}^S$

A first naive quantum circuit can be implemented in a straightforward way (see Fig. 10):

$$\hat{O}_{p^m}^S = e^{-i\pi m/2} \prod_{s=0}^{2^l-1} C_{R_y(\theta_s)}^S; \quad \theta_s = 2 \arccos \left(e^{i\pi m/2} \frac{(\hat{p}^m)^{(s)}}{\sqrt{\mathcal{N}_{p^m}}} \right), \quad (\text{B1})$$

where we hereafter omit the global phase $e^{-i\pi m/2}$; however, we take it into account in cases when it does matter. A more efficient approach can be constructed using Fable implementation of block-encoding [42,43]. A two-qubit toy example of the method that explains the general concept is depicted in Fig. 11. Basically, the authors use Gray code ordering to reduce the complexity of the $\hat{O}_{p^m}^S$ implementation.

FIG. 10. A naive design of the quantum circuit that implements $\hat{O}_{p^m}^S$. Each layer corresponds to the different nonzero elements of the first row of the matrix $\hat{p}^m$. The qubit a correspond to the first register in the definition of $\hat{O}_{p^m}^S$.

The algorithm is fully described in Ref. [42]; here, we just present the complexity: (i) 2^l one-qubit operations and (ii) 2^l C-NOTs.

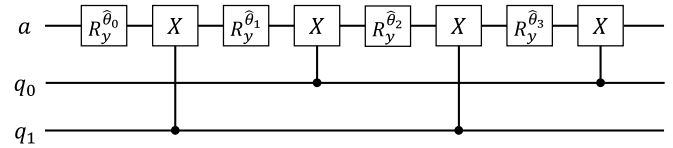
APPENDIX C: EXPLICIT STATE PREPARATION CIRCUIT DESIGN FOR FINITE BINARY NORM QUANTUM STATE

The circuit performs the following task. Assuming that the initial state of the circuit's n qubits is $|0\rangle^n$, we aim to transform it into

$$A_\psi |0\rangle^n = |\psi\rangle^n = \sum_{|i|_b \leq q} \beta_i |i\rangle^n,$$

where the coefficients $\beta_i \neq 0 (|i|_b \leq q)$ are all known. Let $m = \min\{q, n\}$. We first design a circuit, as shown in Eq. (12). Now, we derive the rotation angles for each R_y gate in the circuit.

By derivation, we know that the coefficients of the quantum state $|\psi\rangle^n$ satisfy the following relationship with the angles of

FIG. 11. Fable implementation of the query oracle $\hat{O}_{p^m}^S$. The particular one-qubit rotations angles are according to Ref. [42].

the R_y gates in the circuit.

$$\begin{aligned}
\beta_0 &= \prod_{j_0=0}^{n-1} \cos \frac{\theta_{j_0}^0}{2}, \\
\beta_{2^{p_1}} &= \left(\prod_{j_1=p_1+1}^{n-1} \cos \frac{\theta_{j_1}^{2^{p_1}}}{2} \right) \sin \frac{\theta_{p_1}^0}{2} \left(\prod_{j_0=0}^{p_1-1} \cos \frac{\theta_{j_0}^0}{2} \right) \quad (0 \leq p_1 < n-1), \\
\beta_{2^{n-1}} &= \sin \frac{\theta_{n-1}^0}{2} \left(\prod_{j_0=0}^{n-2} \cos \frac{\theta_{j_0}^0}{2} \right), \\
\beta_{2^{p_1}+2^{p_2}} &= \left(\prod_{j_2=p_2+1}^{n-1} \cos \frac{\theta_{j_2}^{2^{p_1}+2^{p_2}}}{2} \right) \sin \frac{\theta_{p_2}^{2^{p_1}}}{2} \left(\prod_{j_1=p_1+1}^{p_2-1} \cos \frac{\theta_{j_1}^{2^{p_1}}}{2} \right) \sin \frac{\theta_{p_1}^0}{2} \left(\prod_{j_0=0}^{p_1-1} \cos \frac{\theta_{j_0}^0}{2} \right) \quad (0 \leq p_1 < p_2 < n-1), \\
\beta_{2^{p_1}+2^{n-1}} &= \sin \frac{\theta_{n-1}^{2^{p_1}}}{2} \left(\prod_{j_1=p_1+1}^{n-2} \cos \frac{\theta_{j_1}^{2^{p_1}}}{2} \right) \sin \frac{\theta_{p_1}^0}{2} \left(\prod_{j_0=0}^{p_1-1} \cos \frac{\theta_{j_0}^0}{2} \right) \quad (0 \leq p_1 < n-1), \\
&\vdots \\
\beta_{2^{p_1}+\dots+2^{p_{m-2}}+2^{p_{m-1}}} &= \left(\prod_{j_{m-1}=p_{m-1}+1}^{n-1} \cos \frac{\theta_{j_{m-1}}^{2^{p_1}+\dots+2^{p_{m-1}}}}{2} \right) \sin \frac{\theta_{p_{m-1}}^{2^{p_1}+\dots+2^{p_{m-2}}}}{2} \left(\prod_{j_{m-2}=p_{m-2}+1}^{p_{m-1}-1} \cos \frac{\theta_{j_{m-2}}^{2^{p_1}+\dots+2^{p_{m-2}}}}{2} \right) \\
&\quad \times \sin \frac{\theta_{p_{m-2}}^{2^{p_1}+\dots+2^{p_{m-3}}}}{2} \left(\prod_{j_{m-3}=p_{m-3}+1}^{p_{m-2}-1} \cos \frac{\theta_{j_{m-3}}^{2^{p_1}+\dots+2^{p_{m-3}}}}{2} \right) \dots \left(\prod_{j_1=p_1+1}^{p_2-1} \cos \frac{\theta_{j_1}^{2^{p_1}}}{2} \right) \\
&\quad \times \sin \frac{\theta_{p_1}^0}{2} \left(\prod_{j_0=0}^{p_1-1} \cos \frac{\theta_{j_0}^0}{2} \right) \quad (0 \leq p_1 < p_2 < \dots < p_{m-1} < n-1), \\
\beta_{2^{p_1}+\dots+2^{p_{m-2}}+2^{n-1}} &= \sin \frac{\theta_{p_{m-1}}^{2^{p_1}+\dots+2^{p_{m-2}}}}{2} \left(\prod_{j_{m-2}=p_{m-2}+1}^{n-2} \cos \frac{\theta_{j_{m-2}}^{2^{p_1}+\dots+2^{p_{m-2}}}}{2} \right) \\
&\quad \times \sin \frac{\theta_{p_{m-2}}^{2^{p_1}+\dots+2^{p_{m-3}}}}{2} \left(\prod_{j_{m-3}=p_{m-3}+1}^{p_{m-2}-1} \cos \frac{\theta_{j_{m-3}}^{2^{p_1}+\dots+2^{p_{m-3}}}}{2} \right) \dots \left(\prod_{j_1=p_1+1}^{p_2-1} \cos \frac{\theta_{j_1}^{2^{p_1}}}{2} \right) \\
&\quad \times \sin \frac{\theta_{p_1}^0}{2} \left(\prod_{j_0=0}^{p_1-1} \cos \frac{\theta_{j_0}^0}{2} \right) \quad (0 \leq p_1 < p_2 < \dots < p_{m-2} < n-1), \\
\beta_{2^{p_1}+\dots+2^{p_{m-1}}+2^{p_m}} &= \sin \frac{\theta_{p_m}^{2^{p_1}+\dots+2^{p_{m-1}}}}{2} \left(\prod_{j_{m-1}=p_{m-1}+1}^{p_m-1} \cos \frac{\theta_{j_{m-1}}^{2^{p_1}+\dots+2^{p_{m-1}}}}{2} \right) \sin \frac{\theta_{p_{m-1}}^{2^{p_1}+\dots+2^{p_{m-2}}}}{2} \\
&\quad \times \left(\prod_{j_{m-2}=p_{m-2}+1}^{p_{m-1}-1} \cos \frac{\theta_{j_{m-2}}^{2^{p_1}+\dots+2^{p_{m-2}}}}{2} \right) \sin \frac{\theta_{p_{m-2}}^{2^{p_1}+\dots+2^{p_{m-3}}}}{2} \left(\prod_{j_{m-3}=p_{m-3}+1}^{p_{m-2}-1} \cos \frac{\theta_{j_{m-3}}^{2^{p_1}+\dots+2^{p_{m-3}}}}{2} \right) \\
&\quad \dots \left(\prod_{j_1=p_1+1}^{p_2-1} \cos \frac{\theta_{j_1}^{2^{p_1}}}{2} \right) \sin \frac{\theta_{p_1}^0}{2} \left(\prod_{j_0=0}^{p_1-1} \cos \frac{\theta_{j_0}^0}{2} \right) \quad (0 \leq p_1 < p_2 < \dots < p_{m-1} < p_m \leq n-1). \quad (C1)
\end{aligned}$$

Subsequently, we deduce the rotation angles for all R_y gates in the circuit from Eq. (C1). The process begins with determining θ_{n-1}^0 using the ratio

$$\frac{\beta_{2^{n-1}}}{\beta_0} = \tan \frac{\theta_{n-1}^0}{2},$$

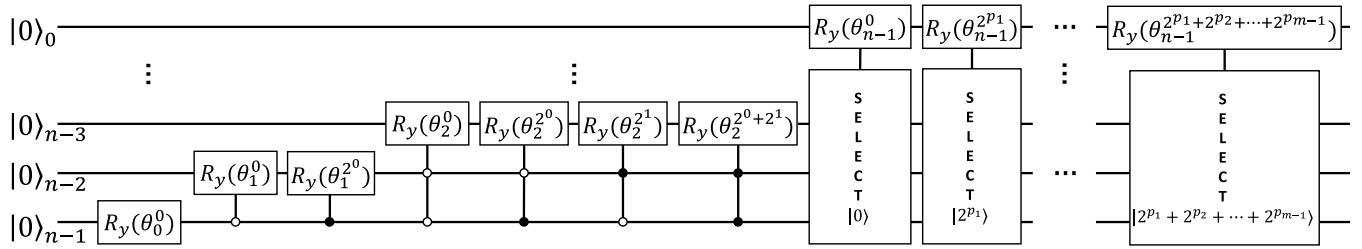


FIG. 12. The initial design of the quantum circuit that implements A_ψ . Angles of R_y gates are given by Eq.(C2)–(C6).

yielding

$$\theta_{n-1}^0 = 2\arctan\left(\frac{\beta_{2^{n-1}}}{\beta_0}\right). \quad (\text{C2})$$

Further, we find angles whose lower indices are all $n-1$

$$\theta_{n-1}^{2^{p_1}+\dots+2^{p_r}} = 2\arctan\left(\frac{\beta_{2^{p_1}+\dots+2^{p_r}+2^{n-1}}}{\beta_{2^{p_1}+\dots+2^{p_r}}}\right), \quad (0 \leq p_1 < \dots < p_r < n-1), \quad (\text{C3})$$

where $r = 1, 2, \dots, m-1$.

To compute the other θ values, for $0 \leq k < n-1$ we derive

$$\frac{\beta_{2^k}}{\beta_0} = \frac{\cos\frac{\theta_{n-1}^{2^k}}{2} \dots \cos\frac{\theta_{k+1}^{2^k}}{2}}{\cos\frac{\theta_{n-1}^0}{2} \dots \cos\frac{\theta_{k+1}^0}{2}} \tan\frac{\theta_k^0}{2},$$

from which we conclude

$$\theta_k^0 = 2\arctan\left(\frac{\cos\frac{\theta_{n-1}^0}{2} \dots \cos\frac{\theta_{k+1}^0}{2} \beta_{2^k}}{\cos\frac{\theta_{n-1}^{2^k}}{2} \dots \cos\frac{\theta_{k+1}^{2^k}}{2} \beta_0}\right). \quad (\text{C4})$$

Assume that $1 \leq k < n-1$. For $r = 1, \dots, \min\{k, m-2\}$ and $0 \leq p_1 < \dots < p_r < k$, we find

$$\frac{\beta_{2^{p_1}+\dots+2^{p_r}+2^k}}{\beta_{2^{p_1}+\dots+2^{p_r}}} = \frac{\cos\frac{\theta_{n-1}^{2^{p_1}+\dots+2^{p_r}+2^k}}{2} \dots \cos\frac{\theta_{k+1}^{2^{p_1}+\dots+2^{p_r}+2^k}}{2}}{\cos\frac{\theta_{n-1}^{2^{p_1}+\dots+2^{p_r}}}{2} \dots \cos\frac{\theta_{k+1}^{2^{p_1}+\dots+2^{p_r}}}{2}} \tan\frac{\theta_k^{2^{p_1}+\dots+2^{p_r}}}{2},$$

leading to

$$\theta_k^{2^{p_1}+\dots+2^{p_{r-1}}} = 2\arctan\left(\frac{\cos\frac{\theta_{n-1}^{2^{p_1}+\dots+2^{p_r}}}{2} \dots \cos\frac{\theta_{k+1}^{2^{p_1}+\dots+2^{p_r}}}{2} \beta_{2^{p_1}+\dots+2^{p_r}+2^k}}{\cos\frac{\theta_{n-1}^{2^{p_1}+\dots+2^{p_r}+2^k}}{2} \dots \cos\frac{\theta_{k+1}^{2^{p_1}+\dots+2^{p_r}+2^k}}{2} \beta_{2^{p_1}+\dots+2^{p_r}}}\right). \quad (\text{C5})$$

Additionally, if $k \geq m-1$, we find

$$\frac{\beta_{2^{p_1}+\dots+2^{p_{m-1}}+2^k}}{\beta_{2^{p_1}+\dots+2^{p_{m-1}}}} = \frac{1}{\cos\frac{\theta_{n-1}^{2^{p_1}+\dots+2^{p_{m-1}}}}{2} \dots \cos\frac{\theta_{k+1}^{2^{p_1}+\dots+2^{p_{m-1}}}}{2}} \tan\frac{\theta_k^{2^{p_1}+\dots+2^{p_{m-1}}}}{2},$$

where $0 \leq p_1 < \dots < p_{m-1} < k$, which gives

$$\theta_k^{2^{p_1}+\dots+2^{p_{m-1}}} = 2\arctan\left(\cos\frac{\theta_{n-1}^{2^{p_1}+\dots+2^{p_{m-1}}}}{2} \dots \cos\frac{\theta_{k+1}^{2^{p_1}+\dots+2^{p_{m-1}}}}{2} \frac{\beta_{2^{p_1}+\dots+2^{p_{m-1}}+2^k}}{\beta_{2^{p_1}+\dots+2^{p_{m-1}}}}\right). \quad (\text{C6})$$

From Eqs. (C2) and (C3) we can get the angles whose lower indices are $n-1$. Then from Eqs. (C4), (C5), and (C6) we get all the angles we need in the circuit.

Based on this circuit, we designed a more efficient circuit (see Fig. 13). It should only remove those “hollow dots” in the original circuit. Here, we need to use a property of R_y gate

$$R_y(\varphi_1 + \varphi_2) = R_y(\varphi_1) + R_y(\varphi_2).$$

Assuming $1 \leq k < n$ and $1 \leq r \leq \min\{k, m-1\}$ with $0 \leq p_1 < \dots < p_r < k$. When $r = \min\{k, m-1\}$

$$\theta_k^{2^{p_1}+\dots+2^{p_{\min\{k, m-1\}}}} = \omega_k^{2^{p_1}+\dots+2^{p_{\min\{k, m-1\}}}}, \quad (\text{C7})$$

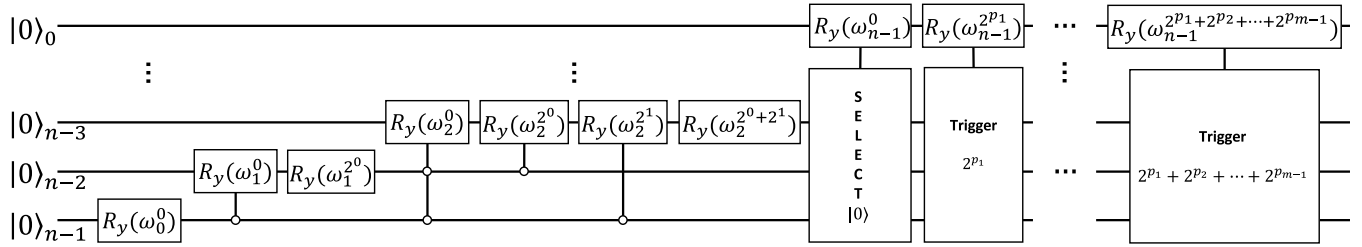


FIG. 13. A more efficient design of the quantum circuit that implements A_ψ . Angles of R_y gates are given by Eq. (C11)–(C14). Here, the term “trigger $2^{p_1} + \dots + 2^{p_r}$ ” refers to controlling only those qubits whose positions are represented by ‘0’s in the binary representation of $2^{p_1} + \dots + 2^{p_r}$.

and when $r < \min\{k, m-1\}$

$$\theta_k^{2^{p_1} + \dots + 2^{p_r}} = \omega_k^{2^{p_1} + \dots + 2^{p_r}} + \sum_{1 \leq j \leq \min\{k, m-1\} - r} \sum_{\substack{0 \leq l_1 < \dots < l_j < k, \\ \{l_1, \dots, l_j\} \cap \{p_1, \dots, p_r\} = \emptyset}} \omega_k^{(2^{p_1} + \dots + 2^{p_r}) + (2^{l_1} + \dots + 2^{l_j})}, \quad (\text{C8})$$

and

$$\theta_0^0 = \omega_0^0, \quad (\text{C9})$$

$$\theta_k^0 = \sum_{\substack{0 \leq l \leq 2^k - 1, \\ |l|_b \leq \min\{k, m-1\}}} \omega_k^l \quad (1 \leq k < n). \quad (\text{C10})$$

Equation (C10) means θ_k^0 is equal to the summation of all ω s whose lower index are k . Here, those θ s are angles of corresponding R_y gates in circuit 12.

The θ s are known from Eqs. (C2)–(C6). The next step involves solving Eqs. (C7)–(C9) to extract ω values from the θ values. Specifically, from Eq. (C7), we have

$$\omega_k^{2^{p_1} + \dots + 2^{p_{\min\{k, m-1\}}}} = \theta_k^{2^{p_1} + \dots + 2^{p_{\min\{k, m-1\}}}} \quad (1 \leq k < n, 0 \leq p_1 < \dots < p_{\min\{k, m-1\}} < k). \quad (\text{C11})$$

Further, from Eq. (C8), we obtain

$$\omega_k^{2^{p_1} + \dots + 2^{p_r}} = \theta_k^{2^{p_1} + \dots + 2^{p_r}} - \sum_{1 \leq j \leq \min\{k, m-1\} - r} \sum_{\substack{0 \leq l_1 < \dots < l_j < k, \\ \{l_1, \dots, l_j\} \cap \{p_1, \dots, p_r\} = \emptyset}} \omega_k^{(2^{p_1} + \dots + 2^{p_r}) + (2^{l_1} + \dots + 2^{l_j})}, \quad (\text{C12})$$

where $1 \leq k < n$, $1 \leq r \leq \min\{k, m-1\}$ and $0 \leq p_1 < \dots < p_r < k$. Additionally, from Eq. (C9)

$$\omega_0^0 = \theta_0^0. \quad (\text{C13})$$

And from Eq. (C10), we determine

$$\omega_k^0 = \theta_k^0 - \sum_{\substack{1 \leq l \leq 2^k - 1, \\ |l|_b \leq \min\{k, m-1\}}} \omega_k^l \quad (1 \leq k < n). \quad (\text{C14})$$

Thus, we have successfully calculated all necessary ω values. In summary, we initially designed Circuit 12 and subsequently simplified it to derive the more efficient Circuit 13. The rotation angles for the R_y gates in Circuit 13 are determined by Eqs. (C11)–(C14).

- [1] M. Pour-El and I. Richards, Simulating physics with computers, *Int. J. Theor. Phys.* **21**, 553 (1982).
 [2] P. W. Shor, Algorithms for quantum computation: Discrete logarithms and factoring, in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science (FOCS)* (IEEE, Santa Fe, NM, USA, 1994), pp. 124–134.

- [3] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM Rev.* **41**, 303 (1999).
 [4] L. K. Grover, A fast quantum mechanical algorithm for database search, in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*

- (Association for Computing Machinery, New York, 1996), pp. 212–219.
- [5] A. Ambainis, Quantum search algorithms, *ACM SIGACT News* **35**, 22 (2004).
 - [6] A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head-Gordon, Simulated quantum computation of molecular energies, *Science* **309**, 1704 (2005).
 - [7] M. Szegedy, Quantum speed-up of Markov chain based algorithms, in *45th Annual IEEE Symposium on Foundations of Computer Science* (IEEE, 2004), pp. 32–41.
 - [8] A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum algorithm for linear systems of equations, *Phys. Rev. Lett.* **103**, 150502 (2009).
 - [9] R. B. Laughlin and D. Pines, The theory of everything, *Proc. Natl. Acad. Sci. USA* **97**, 28 (2000).
 - [10] P. C. Costa, S. Jordan, and A. Ostrander, Quantum algorithm for simulating the wave equation, *Phys. Rev. A* **99**, 012323 (2019).
 - [11] F. Gaitan, Finding flows of a Navier–Stokes fluid through quantum computing, *npj Quantum Inf.* **6**, 61 (2020).
 - [12] 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).
 - [13] N. Linden, A. Montanaro, and C. Shao, Quantum vs. classical algorithms for solving the heat equation, *Commun. Math. Phys.* **395**, 601 (2022).
 - [14] S. Jin, N. Liu, and Y. Yu, Quantum simulation of partial differential equations: Applications and detailed analysis, *Phys. Rev. A* **108**, 032603 (2023).
 - [15] Y. Sato, R. Kondo, I. Hamamura, T. Onodera, and N. Yamamoto, Hamiltonian simulation for time-evolving partial differential equation by scalable quantum circuits, *Phys. Rev. Res.* **6**, 033246 (2024).
 - [16] N. Stamatopoulos, D. J. Egger, Y. Sun, C. Zoufal, R. Iten, N. Shen, and S. Woerner, Option pricing using quantum computers, *Quantum* **4**, 291 (2020).
 - [17] J. Gonzalez-Conde, Á. Rodríguez-Rozas, E. Solano, and M. Sanz, Efficient Hamiltonian simulation for solving option price dynamics, *Phys. Rev. Res.* **5**, 043220 (2023).
 - [18] 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* (Association for Computing Machinery, New York, 2019), pp. 193–204.
 - [19] 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).
 - [20] S. Jin and N. Liu, Analog quantum simulation of partial differential equations, *Quantum Sci. Technol.* **9**, 035047 (2024).
 - [21] Y. Cao, S. Jin, and N. Liu, Quantum simulation for time-dependent hamiltonians—with applications to non-autonomous ordinary and partial differential equations, *J. Phys. A: Math. Theor.* **58**, 155304 (2025).
 - [22] G. H. Low and I. L. Chuang, Optimal hamiltonian simulation by quantum signal processing, *Phys. Rev. Lett.* **118**, 010501 (2017).
 - [23] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, Exponential improvement in precision for simulating sparse Hamiltonians, in *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing* (Association for Computing Machinery, New York, 2014), pp. 283–292.
 - [24] J. Hu, S. Jin, N. Liu, and L. Zhang, Quantum circuits for partial differential equations via Schrödingerisation, *Quantum* **8**, 1563 (2024).
 - [25] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, Cambridge, 2011).
 - [26] C. M. Dawson and M. A. Nielsen, The Solovay-Kitaev algorithm, [arXiv:quant-ph/0505030](https://arxiv.org/abs/quant-ph/0505030).
 - [27] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu, *The Finite Element Method: Its Basis and Fundamentals* (Elsevier, Butterworth-Heinemann, 2005).
 - [28] M. N. Özişik, H. R. Orlande, M. J. Colaço, and R. M. Cotta, *Finite Difference Methods in Heat Transfer* (CRC Press, Boca Raton, Florida, 2017).
 - [29] S. Jin, N. Liu, and Y. Yu, Quantum simulation of partial differential equations via Schrodingerisation, *Phys. Rev. Lett.* **133**, 230602 (2024).
 - [30] S. Jin and N. Liu, Quantum algorithms for computing observables of nonlinear partial differential equations, *Bulletin des Sciences Mathématiques* **194**, 103457 (2024).
 - [31] N. Guseynov, A. Zhukov, W. Pogosev, and A. Lebedev, Depth analysis of variational quantum algorithms for the heat equation, *Phys. Rev. A* **107**, 052422 (2023).
 - [32] J. L. Walsh, A closed set of normal orthogonal functions, *Am. J. Math.* **45**, 5 (1923).
 - [33] A. G. Rattew and P. Rebentrost, Non-linear transformations of quantum amplitudes: Exponential improvement, generalization, and applications, [arXiv:2309.09839](https://arxiv.org/abs/2309.09839).
 - [34] N. Guo, K. Mitarai, and K. Fujii, Nonlinear transformation of complex amplitudes via quantum singular value transformation, *Phys. Rev. Res.* **6**, 043227 (2024).
 - [35] A. M. Dalzell, S. McArdle, M. Berta, P. Bienias, C.-F. Chen, A. Gilyén, C. T. Hann, M. J. Kastoryano, E. T. Khabiboulline, A. Kubica *et al.*, Quantum algorithms: A survey of applications and end-to-end complexities, [arXiv:2310.03011](https://arxiv.org/abs/2310.03011).
 - [36] M. Plesch and Č. Brukner, Quantum-state preparation with universal gate decompositions, *Phys. Rev. A* **83**, 032302 (2011).
 - [37] V. Bergholm, J. J. Vartiainen, M. Möttönen, and M. M. Salomaa, Quantum circuits with uniformly controlled one-qubit gates, *Phys. Rev. A* **71**, 052330 (2005).
 - [38] V. V. Shende, I. L. Markov, and S. S. Bullock, Minimal universal two-qubit controlled-not-based circuits, *Phys. Rev. A* **69**, 062321 (2004).
 - [39] J. L. Coolidge, The story of the binomial theorem, *Am. Math. Mon.* **56**, 147 (1949).
 - [40] T. G. Draper, Addition on a quantum computer, [arXiv:quant-ph/0008033](https://arxiv.org/abs/quant-ph/0008033).
 - [41] T. G. Draper, S. A. Kutin, E. M. Rains, and K. M. Svore, A logarithmic-depth quantum carry-lookahead adder, *Quantum Info. Comput.* **6**, 351 (2006).
 - [42] D. Camps and R. Van Beeumen, Fable: Fast approximate quantum circuits for block-encodings, in *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE, 2022), pp. 104–113.
 - [43] M. Möttönen, J. J. Vartiainen, V. Bergholm, and M. M. Salomaa, Quantum circuits for general multiqubit gates, *Phys. Rev. Lett.* **93**, 130502 (2004).