Title

Improving Schrödinger Equation Implementations with Gray Code for Adiabatic Quantum Computers

Permalink
https://escholarship.org/uc/item/4s66x708

Journal
PRX Quantum, 3(2)

ISSN
2691-3399

Authors
Chang, Chia Cheng
McElvain, Kenneth S
Rrapaj, Ermal
et al.

Publication Date
2022-06-01

DOI
10.1103/prxquantum.3.020356

Peer reviewed

Improving Schrödinger Equation Implementations with Gray Code for Adiabatic Quantum Computers

Chia Cheng Chang,[1,2,3,4,*] Kenneth S. McElvain,[2,3] Ermal Rrapaj,[2] and Yantao Wu[1,2,5]

[1] *RIKEN iTHEMS, Wako, Saitama 351-0198, Japan*
[2] *Department of Physics, University of California, Berkeley, California 94720, USA*
[3] *Nuclear Science Division, Lawrence Berkeley National Laboratory, Berkeley, California 94720, USA*
[4] *LinkedIn Corporation, Sunnyvale, California 94085, USA*
[5] *Department of Physics, Princeton University, Princeton, New Jersey 08544, USA*

(Received 14 April 2021; revised 7 October 2021; accepted 26 May 2022; published 15 June 2022)

We reformulate the continuous-space Schrödinger equation in terms of spin Hamiltonians. For the kinetic energy operator, the critical concept facilitating the reduction in model complexity is the idea of position encoding. A binary encoding of position produces a spin-1/2 Heisenberg-like model and yields exponential improvement in space complexity when compared to classical computing. Encoding with a binary reflected Gray code (BRGC), and a Hamming-distance-2 Gray code (H2GC) reduces the model complexity down to the XZ and transverse Ising model, respectively. For A qubits BRGC yields 2^A positions and is reduced to its 2-local form with $\mathrm{O}(A)$ ancillary qubits. H2GC yields $2^{A/2+1}$ positions with $\mathrm{O}(A^2)$ three-local penalty terms. We also identify the bijective mapping between diagonal unitaries and the Walsh series, producing the mapping of any real potential to a series of k-local Ising models through the fast Walsh transform. Finally, in a finite volume, we provide some numerical evidence to support the claim that the total time needed for adiabatic evolution is protected by the infrared cutoff of the system. As a result, initial state preparation from a free-field wave function to an interacting system is expected to exhibit polynomial time complexity with volume and constant scaling with respect to lattice discretization for all encodings. For H2GC, if the evolution starts with the transverse Hamiltonian due to hardware restrictions, then penalties are dynamically introduced such that the low-lying spectrum reproduces the energy levels of the Laplacian. The adiabatic evolution of the penalty Hamiltonian is therefore sensitive to the ultraviolet scale. It is expected to exhibit polynomial time complexity with lattice discretization, or exponential time complexity with respect to the number of qubits given a fixed volume.

DOI: 10.1103/PRXQuantum.3.020356

I. INTRODUCTION

The understanding of many physical problems requires obtaining Schrödinger equation solutions for the system under study. In this work, we develop techniques to solve it with adiabatic quantum computing. A typical classical computing choice for numerically solving the Schrödinger equation is to pick a discrete basis in which to express the Hamiltonian. Then, one diagonalizes the resulting matrix, either completely, or for very large problems, uses techniques such as the Lanczos algorithm to find low-lying eigenstates and eigenvalues. The discrete basis can, for example, be comprised of the states of a harmonic oscillator, or some other exactly solvable Hamiltonian. Other useful basis choices are a discrete position, or momentum basis. As a first step, we focus on a simple version of the problem: a one-body system with a local potential in a D-dimension periodic position basis.

$$-\frac{\hbar^2}{2M}\nabla^2\psi(x)+V(x)\psi(x)=E\psi(x). \tag{1}$$

For simplicity of notation, throughout this paper, we work in natural units, $\hbar=c=1$. We discretize the equation on a lattice with spacing a and N positions in each of the D directions. Then, up to a discretization error proportional to a, the Laplacian becomes an $N\times N$ matrix, which acts

[*] chiachang@berkeley.edu

on the discretized wave function $\psi(a\mathbf{m})$:

$$\begin{aligned}\nabla^2\psi(x) &\approx \frac{1}{a^2}(L\psi)(a\mathbf{n})\\ &= \frac{1}{a^2}\left[\left(\sum_{\mathbf{m}\in\mathcal{N}(n)}\psi(a\mathbf{m})\right)-2D\psi(a\mathbf{n})\right],\end{aligned} \tag{2}$$

with $\mathcal{N}(\mathbf{n})$ indicating the set of immediate neighbors of the discrete point $\mathbf{n}$. Here, we use L to denote the dimensionless part of the Laplacian. The discrete Schrödinger equation then follows as

$$(H\psi)(a\mathbf{n}) = -\frac{1}{2Ma^2}(L\psi)(a\mathbf{n}) + V(a\mathbf{n})\psi(a\mathbf{n}) = E\psi(a\mathbf{n}). \tag{3}$$

The next step is to encode the positions in states of qubits (spins). One choice of encoding in use is to associate position i with a set of qubits [1,2]. In each position, the value of the function is given by a fixed-point representation of the qubits. The advantage of such an encoding is that the solution is diagonal in the computational basis, and is implementable with quantum annealers available today. However, the number of qubits is comparable to the number of classical bits required to solve the same problem.

Alternatively, one can associate positions with A-body qubit states in the computational basis and identify each basis state's amplitude with the wave function at the corresponding point. Such an association produces a lattice with 2^A sites, yielding an exponential improvement in space complexity. This approach appears in circuit-based quantum algorithms [3], specifically associating the state of qubit i with the value of bit i of the position index. Bit i of a number is the coefficient of 2^i in the base-2 representation of the number.

In this work, we explore the advantages of other encoding possibilities which yield simpler spin Hamiltonians. A first encoding choice uses the binary reflected Gray code (BRGC) to represent the sequence of positions, with the bits of the code having the same connection to the qubit states as before. This option requires only the $\sigma_i^x\sigma_j^z$ operators, in addition to the σ_i^x and $\sigma_i^z\sigma_j^z$ operators in the transverse-field Ising model, and has the key advantage of allowing the Laplacian matrix be reduced to a 2-local form with O(A) number of auxillary qubits. The BRGC encoding preserves the maximum 2^A lattice sites that can be generated from A qubits. Gray codes are proposed for encoding ladder states in d-level systems to simplify raising and lowering operators in gate-based quantum computing [4]. In Ref. [5] this idea is applied to finding the ground-state energy of a deuteron in a harmonic oscillator basis with a simulated variational quantum eigensolver (VQE). Here, we extend the application of BRGC to map the Schrödinger equation, in any dimension, to the XZ model.

A second Gray code, which we call a Hamming-distance-2 Gray code (H2GC), introduces an alternative mapping of the Schrödinger equation requiring only the transverse-field Ising model, i.e., containing only two-body $\sigma_i^z\sigma_j^z$ and one-body σ_i^x couplings. The mapping retains an exponential number of valid lattice sites associated with A bit codes in the sequence, while the invalid codes are nulled using an O(A^2) number of 3-local penalty terms. As a result, the H2GC formulation is polynomially equivalent to BRGC while reducing the complexity of the spin model.

For the examples in this work we consider a single particle moving in D dimensions. In general, the extension to m particles moving in d spatial dimensions is equivalent to $D = md$. If each of the D dimensions is discretized into N lattice points, one needs $\log_2(N^D) = D\log_2(N)$ qubits to represent the entire discretized lattice.

In this work, we treat all particles as being distinguishable, i.e., particles with Boltzmann statistics. In quantum chemistry, one often starts from a potential energy surface of atoms and studies the atoms directly without tracking the dynamics of the electrons [6,7]. In this case, Boltzmann statistics gives an accurate description of molecules when identical particles, e.g., the hydrogen nuclei in the molecule malonaldehyde $C_3H_4O_2$, are not too close in the ground-state wave function. For example, the tunneling splitting energy and the quantum momentum distribution can be computed accurately with sampling techniques assuming Boltzmann statistics [8,9]. In problems like these, our method offers a direct way to efficiently compute the ground-state wavefunction of molecules with adiabatic quantum computing. Particles with bosonic or fermionic statistics are important and natural extensions, but are beyond the scope of this work.

For the convenience of the reader, the notation used throughout this work is defined in Sec. I A. In Sec. II we provide the mapping of the discretized Laplacian to a k-local Hamiltonian, in binary, BRGC, and H2GC codes in Secs. II A, II B, and II C, respectively. Then, we proceed to describe the mapping of the local potential to any Gray code in Sec. III. Having provided all the necessary steps for encoding the Hamiltonian, in Sec. IV we provide various simulations of quantum adiabatic computation of the ground state. Specifically, in Sec. IV A we study a BRGC encoded S-wave nucleon potential that reproduces the deuteron binding energy. In Sec. IV B we focus on a two-dimensional quartic and quadratic set of potentials activated in different time intervals to study both initial state preparation and time evolution of the system. We also provide an example of the H2GC code with a harmonic oscillator potential in Sec. IV C. We conclude with a summary of our results in Sec. V.

A. Notation and definitions

Before we begin our discussion, the notation used throughout the paper is defined here for clarity.

First, we define a bijection between binary bits and qubit states. Spin up will be associated with a bit value of 0 or $|0\rangle$, and spin down with a bit value of 1 or $|1\rangle$. Basis states of an A-body qubit system are therefore associated with an A-bit binary string with the usual interpretation as an integer in a base-2 representation. For matrices and vectors over the basis, we order the entries according to the integer value of the corresponding state's bit string. Let $\mathbb{1}$ be the 2×2 identity matrix, and σ^x, σ^y, and σ^z be the Pauli matrices:

$$\sigma^x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \sigma^y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \sigma^z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{4}$$

The set $\{\mathbb{1}, \sigma^x, i\sigma^y, \sigma^z\}$ forms a basis of 2×2 real matrices. Thus, any real matrix of size $2^A \times 2^A$ has a unique tensor product decomposition with these four matrices.

Throughout the paper, for a matrix M of size $2^A \times 2^A$, or an array V of size 2^A, the indices will be denoted by square brackets: $M[k,m]$ is the (k,m) element of M and $V[k]$ is the kth element of V. Subscripts of operators denote the qubit index. When we index qubits, we start from 0 and count from the right. For example,

$$\sigma_1^z = \mathbb{1} \otimes \mathbb{1} \otimes \sigma^z \otimes \mathbb{1} \tag{5}$$

means that σ^z is acting on qubit 1, while the tensor product of the three identity operators explicitly states that we are working in a Hilbert-space of four qubits. In the subscript notation, the dimension of the Hilbert space is unspecified, and is explicitly stated if necessary (e.g., when we provide explicit examples).

For multiqubit operators such as the Laplacian, we explicitly list all indices in the subscript. For example, we label a three-qubit Laplacian operator acting on qubits 0, 1, 2 as

$$L_{0\ldots2}^{(3,\text{bin})}. \tag{6}$$

Additionally, the superscript "bin" denotes that the Laplacian is expressed in binary order. In this work, we also derive the Laplacian in "BRGC" and "H2GC" forms for the binary reflected Gray code and the Hamming-distance-2 Gray code.

For convenience in what follows we define qubit (spin) projection operators

$$P^0 = \frac{\mathbb{1} + \sigma^z}{2} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad P^1 = \frac{\mathbb{1} - \sigma^z}{2} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \tag{7}$$

where P^0 projects onto $|0\rangle$ (spin up) and P^1 onto $|1\rangle$ (spin down) for a single qubit. Raising and lowering operators on a spin are defined as

$$\begin{aligned} \sigma^+ &= \frac{\sigma^x + i\sigma^y}{2} = \sigma^x P^1 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \\ \sigma^- &= \frac{\sigma^x - i\sigma^y}{2} = \sigma^x P^0 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}. \end{aligned} \tag{8}$$

The variable A indicates the number of qubits in the system.

Readers who do not speak binary as a first or second language are highly encouraged to read Appendix B, which summarizes the various binary representations used in this work and their related Walsh functions, which are the foundation of our construction of arbitrary real potentials. We make substantial use of Karnaugh maps [10] in describing the construction of the H2GC Laplacian. Karnaugh maps are used in boolean circuit minimization and we include a brief introduction biased towards our application in Appendix D. Additionally, for readers who would enjoy a more in-depth overview of orthogonal functions and Gray codes, there are many textbooks available in the literature (e.g., Ref. [11]).

II. CONSTRUCTION OF THE LAPLACIAN

In this section, we present the mapping of the discrete Laplacian to k-local Hamiltonians. The simplest form of the discrete Laplacian is given by the nearest-neighbor finite-difference method,

$$\frac{\partial^2}{\partial x^2} f(x) = \frac{1}{a^2} \left[f(x+a) + f(x-a) - 2f(x) \right], \tag{9}$$

where a is the lattice spacing. In operator form, the dimensionless part of the discrete Laplacian is a tridiagonal matrix with additional nonzero entries in the ends of the antidiagonal due to periodic boundary conditions. For example, a one-dimensional (1D) lattice with 2^3 lattice sites has a binary encoded Laplacian operator given by

$$L_{0\ldots2}^{(3,\text{bin})} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \tag{10}$$

where we drop the -2 down the main diagonal. In the context of Hamiltonian evolution, the main diagonal contributes a global time-dependent phase and a shift of the eigenvalues by a constant while leaving the eigenvectors unchanged. Note that the full Laplacian operator is $(1/a^2)L$.

The results can be generalized to the multidimensional case since contributions in different dimensions are independent. For example, in two dimensions with A_x qubits in the x direction, A_y qubits in the y direction, and independent of the position encoding, we have

$$L^{(A_x,A_y)}_{0\ldots(A_x-1),A_x\ldots(A_x+A_y-1)} = L^{(A_x)}_{0\ldots(A_x-1)} + L^{(A_y)}_{A_x\ldots(A_x+A_y-1)}. \tag{11}$$

The inexpensive addition of multiple dimensions means that the Laplacian, with Boltzmann statistics, scales linearly with the number of particles. In general the encoding for a D-dimensional lattice is given by layers of one-dimensional codes. An explicit example of two dimensions with four lattice positions in x and eight in y (32 sites in all), encoded in binary with the leading two qubits for x and last three qubits for y is

y x	0	1	2	3	4	5	6	7
0	00000	00001	00010	00011	00100	00101	00110	00111
1	01000	01001	01010	01011	01100	01101	01110	01111
2	10000	10001	10010	10011	10100	10101	10110	10111
3	11000	11001	11010	11011	11100	11101	11110	11111

A two-dimensional example on a larger lattice is implemented and analyzed in Sec. IV B.

In the following sections, we first present the mapping in binary encoding in Sec II A, which requires the full Pauli basis, the derivation of the BRGC Laplacian in Sec. II B, which maps to the *XZ* model, and the H2GC Laplacian in Sec. II C, which maps to the transverse Ising model.

A. The Laplacian matrix in the binary encoding

Let $L^{(A,\text{bin})}$ be the Laplacian matrix of 2^A lattice points with periodic boundary condition in one dimension. When $A = 3$, for example, $L^{(3,\text{bin})}$ is given by Eq. (10). We define the operator

$$C^{(A)}_{0\ldots A-1} = \prod_{i=0}^{A-1}\sigma^+_i + \prod_{i=0}^{A-1}\sigma^-_i .$$

Then one obtains a recursive formula for $L^{(A,\text{bin})}$:

$$\begin{aligned} L^{(A,\text{bin})}_{0\ldots A-1} &= L^{(A-1,\text{bin})}_{0\ldots A-2} - C^{(A-1)}_{0\ldots A-2} + \sigma^x_{A-1} C^{(A-1)}_{0\ldots A-2} \\ &= L^{(A-1,\text{bin})}_{0\ldots A-2} + (\sigma^x_{A-1} - 1)\left(\prod_{i=0}^{A-2}\sigma^x_i P^0_i + \prod_{i=0}^{A-2}\sigma^x_i P^1_i\right) \end{aligned} \tag{12}$$

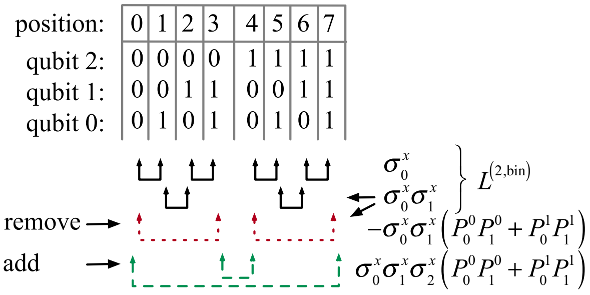
starting from the two-site Laplacian with periodic boundary conditions

$$L^{(1)}_0 = 2\sigma^x_0. \tag{13}$$

We emphasize that $L^{(1)}_0$ is the same for all codes and is the starting condition for all recursive formulas presented in this work. As a reminder, the first term in Eq. (12) does not include the index $A - 1$ and, therefore, there is an implied identity operator on this index, as shown in Eq. (5). The same convention is applied throughout the paper.

Figure 1 gives a graphical derivation of Eq. (12) for $A = 3$. The sum of projection operator products can be seen to be picking out the ends of the 2^{A-1} position subregions. Then, the σ^x operators add the new green dashed contributions in, and subtract the old red dotted contributions out.

The σ^x_i part of the correction generates many copies of the product $\sigma^x_0\sigma^z_0 = -i\sigma^y_0$, so the Laplacian includes all



position:	0	1	2	3	4	5	6	7
qubit 2:	0	0	0	0	1	1	1	1
qubit 1:	0	0	1	1	0	0	1	1
qubit 0:	0	1	0	1	0	1	0	1

σ^x_0
$\sigma^x_0\sigma^x_1$
$L^{(2,\text{bin})}$
remove
add
$-\sigma^x_0\sigma^x_1\left(P^0_0P^0_1 + P^1_0P^1_1\right)$
$\sigma^x_0\sigma^x_1\sigma^x_2\left(P^0_0P^0_1 + P^1_0P^1_1\right)$

FIG. 1. The three-qubit binary encoded Laplacian is constructed from the two-qubit Laplacian and two corrections. The columns under the position index line are the position encoding in qubit states. For example, position state 4 in the second subblock is encoded as 100 in qubit states specified below position index 4. The horseshoe shaped lines with arrows indicate the symmetric contributions between neighboring positions. The first three row of lines show the contributions inherited from $L^{(2,\text{bin})}$, including contributions shown as red (dotted) lines that are excess in the $A = 3$ context. The excess contribution is removed by term $-\sigma^x_0\sigma^x_1\left(P^0_0P^0_1 + P^1_0P^1_1\right)$. The green (dashed) lines indicate the two contributions not provided by $L^{(2,\text{bin})}$ that are added by the term to the right. The corrections are shown next to the red and green lines.

three Pauli matrices if expanded. In conjunction with the potential, this mapping uses the entire Pauli basis. Current quantum annealers can only evolve qubit systems from the transverse-field Hamiltonian to the classical Ising model [12]. While hardware improvements may be developed to handle couplings of other Pauli products, such as XZ, in the near future, one expects that it will still handle only Hamiltonians composed of terms with small Pauli support [13].

In adiabatic quantum computing, if the target Hamiltonian is classical, i.e., diagonal in the computational basis, the reduction of multiqubit to two-qubit interactions is well understood [14]. In Eq. (12), however, the operator is nondiagonal in the computational basis on most of the sites, making the method in Ref. [14] not applicable. In Sec. II B, we introduce the BRGC encoding of the Laplacian, which allows for reduction with $\mathrm{O}(A)$ qubits to a 2-local form. This is an important advantage of a BRGC encoding over the binary one.

B. The Laplacian matrix in the binary reflected Gray encoding

As explained in the previous section, the tensor-product decomposition of the Laplacian matrix in the binary encoding has the undesirable σ^y terms. Thus, it is natural to ask whether it is possible to find a position encoding so that the tensor-product decomposition of the Laplacian is simpler. We show that the BRGC encoding of position, matching position x to the qubit state specified by the xth member of the BRGC, achieves a dramatic simplification.

An implementable qubit or spin Hamiltonian in current quantum annealers is the sum of transverse fields:

$$H^x = \sum_{i=0}^{A-1} \sigma_i^x. \tag{14}$$

For a system with A qubits, H^x contains $2^{A-1}A$ symmetric couplings between qubit states differing in one bit. By symmetric, we mean that H^x couples, for example, qubit states 000 to 001 and 001 to 000. Each of these symmetric couplings will give rise to two 1s in H^x expressed in the tensor products of the A qubits. We call the symmetric coupling a connection between the qubit states. Note that the Laplacian matrix couples consecutive positions. Thus, if the A-body qubit states are ordered so that the consecutive states differ only in the state of one qubit, then the representation of H^x in the ordering contains all the nonzero elements of the Laplacian matrix [15]. Gray codes, well-known in signal processing, have this property [16–18]. We give a short review below.

1. Gray encoding

Any Gray encoding (reflection-based or not) of the positions guarantees that neighboring bit strings differ in exactly one bit. Gray code is an alternate compact binary encoding of integers 0 to 2^A-1 into A bits. For example, the standard binary encoding of the numbers from 0 to 3 is 00, 01, 10, 11. The binary encoding is convenient for arithmetic, but neighboring numbers have varying numbers of bit differences. For neighborhood operators like the Laplacian we would like to minimize the bit difference between nearby points. Gray code does this, resulting in neighboring points differing in only one bit in their code. If we are working in a periodic space, this property is preserved with a one bit difference between the first and last point. The most common Gray code is the binary reflected Gray code, which can be constructed by induction on A. For the base case, $A=1$, we encode $\{0,1\}$ as $\{0,1\}$. For larger A we concatenate the codes for the $A-1$ case to the reflected (or reversed order) codes for the $A-1$ case. We then add a most significant bit of 0 to the first half and a 1 to the second half. For $A=2$, this procedure yields $\{00,01,11,10\}$, and for $A=3$ it yields $\{000,001,011,010,110,111,101,100\}$.

In general, for an encoding G, let the *encoding function* $G(n)$ denote the integer that the nth bit string of G represents in binary. When we have a specific encoding function, we explicitly use its code name to denote it. For example,

$$\mathrm{BRGC}(0)=0,\ \mathrm{BRGC}(1)=1,\ \mathrm{BRGC}(2)=3,\ \mathrm{BRGC}(3)=2,$$
$$\mathrm{BRGC}(4)=6,\ \mathrm{BRGC}(5)=7,\ \mathrm{BRGC}(6)=5,\ \mathrm{BRGC}(7)=4.$$

To transform a matrix M between different encodings, we view the matrix as intrinsically defined with respect to the A-body qubit states. Because the qubit states are ordered differently in different encodings, the matrix transformation is induced by the encoding function G. The encoding function for the binary encoding is the identity map, and therefore M is just M. For the M in encoding G, we define

$$M^{(A,\mathrm{G})}[G(k), G(m)] \equiv M^{(A,\mathrm{bin})}[k,m]. \tag{15}$$

For simplicity, we also denote the nth bit string of G with $G(n)$.

2. The recursive formula for the Laplacian matrix

In the case of the Laplacian, $L^{(\mathrm{bin})}$ is tridiagonal, but $L^{(A,\mathrm{BRGC})}$ is not. As explained above, $L^{(A,\mathrm{BRGC})}$ should be closer to H^x than $L^{(A,\mathrm{bin})}$. For example, when $A=2$,

$$L^{(2,\mathrm{bin})} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array}\!\!\begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \end{array} \xrightarrow[\text{to BRGC}]{\text{bin}} \begin{array}{c} \\ 0 \\ 1 \\ 3 \\ 2 \end{array}\!\!\begin{array}{c} \begin{array}{cccc} 0 & 1 & 3 & 2 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

$$= \sum_{i=0}^{1} \sigma_i^x = H^x = L^{(2,\mathrm{BRGC})}.$$

In the Gray encoding, the 2^A connections between adjacent positions in the Laplacian matrix are generated by H^x, but there are $2^{A-1}A - 2^A = 2^{A-1}(A-2)$ extra connections that must be suppressed. We suppress the extra contributions by multiplying by a sum of projection operators on one or more bits. The projection operators are the simplest if they represent a projection onto a lower dimension subspace of the 2^A binary hypercube.

With a BRGC encoding of the positions, the Laplacian has a recursive decomposition that follows the recursive definition of the BRGC itself. Because the two sub-blocks are reflections of each other with leading 0 and 1 bits added, we know precisely the codes at the sub-block boundaries. The outer codes are all zero except for the leading bit. The inner codes are zero except for the two leading bits being 01 and 11. All corrections required to construct the larger A-bit Laplacian from the $A-1$ bit Laplacian take place in the subspace of the four codes and are restricted to that subspace by a simple projection operator.

The base case is the one-qubit Laplacian of Eq. (13). We emphasize, however, for $A = 2$, the neighbor contributions to the Laplacian operator are

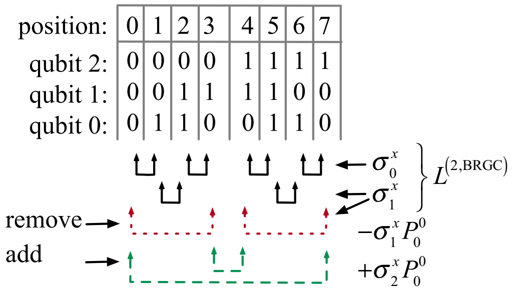
$$L^{(2,\mathrm{BRGC})}_{0,1} = \sigma^x_0 + \sigma^x_1, \tag{16}$$

which we again recognize as the transverse Hamiltonian acting on qubits 0 and 1. For larger A we have

$$L^{(A,\mathrm{BRGC})}_{0\ldots A-1} = L^{(A-1,\mathrm{BRGC})}_{0\ldots A-2} + \left(\sigma^x_{A-1} - \sigma^x_{A-2}\right)\prod_{i=0}^{A-3} P^0_i. \tag{17}$$

The reduction to a 2-local form of the projector product is addressed in Sec. II B 3. For $L^{(A,\mathrm{BRGC})}$, the decomposition of all projectors result in an additional O(A) ancillary qubits and 2-local terms. A graphical derivation of Eq. (17) is given in Fig. 2. A more detailed derivation is given in Appendix C. In this construction, it is clear that since projection operators contain only constants and σ^z operators, and that no products of Pauli matrices are taken that act on the same qubits, that each iteration will not introduce either σ^y or products of σ^x. Since the base case Eq. (16) has no σ^y operators or σ^x products, this property is preserved for all A.

Figure 2 shows how the added and subtracted terms correct the 1D Laplacian $L^{(2,\mathrm{BRGC})}$ to make $L^{(3,\mathrm{BRGC})}$. All the required corrections take place in the subspace defined by the lower $A-2$ qubits being zero. The trailing product of projection operators in Eq. (17) implements the projection onto that subspace. In a system with A qubits, $L^{(A-1)}$ makes extra 2^{A-1} periodic contributions, shown as red dotted lines, which must be removed, as well as missing contributions connecting the ends of the two 2^{A-1} length sub-blocks, shown as dashed green lines, that must be added.



position:	0	1	2	3	4	5	6	7
qubit 2:	0	0	0	0	1	1	1	1
qubit 1:	0	0	1	1	1	1	0	0
qubit 0:	0	1	1	0	0	1	1	0

FIG. 2. The three-qubit BRGC Laplacian is constructed from the two-qubit Laplacian and two corrections. The columns under the position index line are the position encoding in qubit states. For example, position state 4 in the second sub-block is encoded as 110 in qubit states. The horseshoe-shaped lines with arrows indicate the symmetric contributions between adjacent positions with σ^x_0 generating the first row and σ^x_1 the second and third rows. The third row, drawn as a red (dotted) line, is an excess contribution generated by σ^x_1 in the $A = 3$ context and is removed by $-\sigma^x_1 P^0_0$. The last row, drawn with green (dashed) lines, shows the missing sub-block end contributions between positions pairs $\{0, 7\}$ and $\{3, 4\}$, which are added with term $\sigma^x_2 P^0_0$.

3. *Reduction to the 2-local form*

One can immediately see that the correction terms in each iteration of Eq. (17) are simple, containing none of the XX couplings appearing in $L^{(\mathrm{bin})}$. In particular, in Eq. (17), the nondiagonal operator σ^x_i appears just once each term. Thus, one can reduce the shared product of $A-2$ projection operators P^0_i into a single qubit, for which the classical method in Ref. [14] applies. This gives the 2-local Laplacian containing XZ couplings. It is important to note that the number of ancillary qubits required is a linear function of A, preserving the exponential improvement over classical computing. See Appendix A for the reduction of $L^{(A,\mathrm{BRGC})}$ to a 2-local Hamiltonian.

C. The Laplacian matrix in the Hamming-distance-2 Gray encoding

The advantage of a Gray encoding of position is that neighboring positions always differ in a single bit flip, meaning that H^x automatically generates the neighbor contributions to the Laplacian. The Hamming distance between two codes is defined to be the number of bit differences between them. The existence of pairs of nonadjacent positions of Hamming distance 1 requires the use of projection operators to eliminate the associated unwanted contributions. This motivates a second Gray code sequence that we call a Hamming-distance-2 Gray code sequence. With H2GC any two codes in the sequence that are not sequential neighbors are at least Hamming distance 2 from each other. Finding the longest such sequence is known as the snake-in-a-box problem; optimal sequences are unknown for large A. However, lower-bound constructions

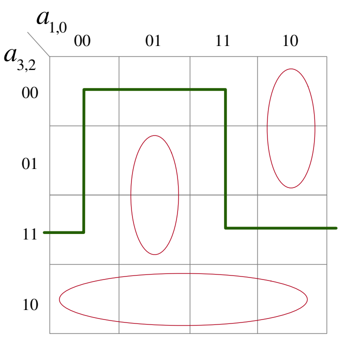


FIG. 3. The Karnaugh map of the $A = 4$ base case for recursive H2GC construction. The sequence is illustrated with a solid line and passes through eight codes. The remaining eight unused codes are covered by a 2-local term on the bottom, and a pair of 3-local terms indicated by vertical ovals.

show that the sequence length grows exponentially [19], as does the number of omitted codes. These longest-length H2GCs are quite irregular and would have large and complex penalty Hamiltonian contributions for the excluded codes.

We now demonstrate a recursive regular construction of an H2GC sequence in which the length grows slightly slower, as $n = 2^{A/2+1}$, but with an efficient implementation of the penalty Hamiltonian. We begin with an $A = 4$ construction with sequence length 8 as a base case, illustrated in Fig. 3 on the Karnaugh map.

The recursive step builds the $A + 2$ qubit sequence from two copies of the A-qubit sequence. The first step is the removal of a link in the cycle for A, exposing two ends that will be used to join the two copies in a larger cycle via codes in the upper-right subspace of Fig. 4. For $A = 4$ we choose code $a_{3,2,1,0} = 0001$ to remove, as the required penalty term is only 2-local. For larger A we choose one of the codes in the upper-right subspace previously used to join two subsequences. A 3-local penalty term suffices to cover the opened code without overlapping the H2GC sequence. For $A = 4$, this is obvious, and for larger A, the links between the $A - 2$ sequence copies are isolated in a subspace encoded by $a_{A+1,A} = 01$ ($a_{5,4} = 01$ in the figure) and a single variable suffices to distinguish the two link codes in the subspace, which can be seen in the upper right $a_{5,4} = 01$ subspace of Fig. 4.

We then make two copies of the A sequence associating one with a code of $a_{A+1,A} = 00$ and the other with a code of $a_{A+1,A} = 11$. In doing so, all existing penalty terms extend over all four values of the new qubits, and are therefore preserved unchanged for the $A + 2$ sequence.

The penalty cover for the bottom-left subspace, which the sequence does not enter, is completed by a single 2-local term that depends only on the two new qubits. The upper-right subspace has two drill-through sequence members added that complete the new H2GC cycle. The penalty cover for the unused part of the upper-right subspace is completed by $(A - 3)$ 3-local terms, each including one subspace qubit having the same value for both drill-through codes, with the penalty term carrying the the opposite value for the qubit, and a code of 01 for the new qubits. It would be $(A - 2)$ 3-local terms, but the inherited penalty terms will already cover $1/4$ of the subspace, seen in the bottom row of the subspace. At larger A, the preexisting coverage comes from the lower-left subspace of the $A - 2$ sequence.

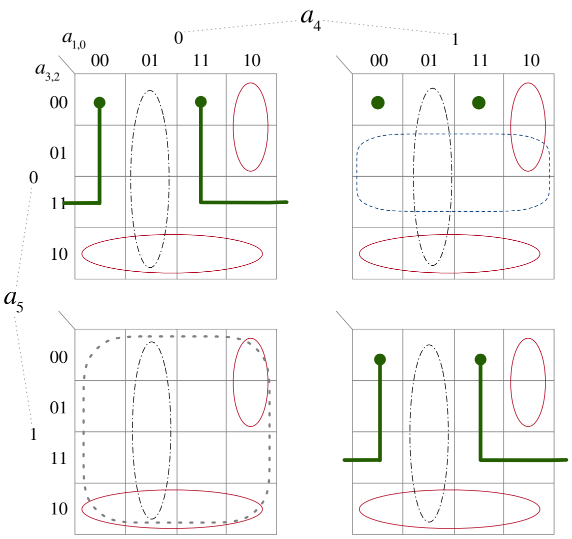


FIG. 4. The Karnaugh map for the recursive construction of the cycle for $(A + 2)$ qubits creates two copies of the A qubit cycle with a gap, isolated by a difference of both new qubits. Solid ovals are the original penalty cover extended over the two new qubits. The upper right subspace contains the two codes used to connect the copies into a new cycle. They are referred to as drill-through codes because they connect the subsequent copies through the isolating layer represented by the upper-right subspace. The additional penalty cover for the upper right subspace is a set of large subcubes, each covering half the subspace. Code values for $a_{3,2}$ and $a_{1,0}$ should be applied across the entire figure.

The upper-left and lower-right subspaces are themselves distance-2 sequences because they are derived from the known good sequence on A qubits. All members of the upper left and lower right differ in both of the new qubits. The drill-through codes in the upper right differ in one bit with the adjacent sequence member in the upper-left and lower-right subspace, and must differ in an additional bit with any other sequence member in the upper left or lower right. And last, the two drill-through codes differ by two bits from each other by construction. We conclude therefore that the constructed sequence is a H2GC sequence.

The construction of the $(A + 2)$ sequence, adds $(A - 2)$ 3-local terms to the penalty Hamiltonian. Therefore, the number of 3-local terms grows quadratically with A. A reduction to 2-local requires at most (some ancillary qubits may be shared) $(A-2)$ additional ancillary qubits.

A key observation is that the Hamiltonian for the system now takes the form

$$\sum_{i=0}^{A-1}\sigma_i^x + A(t)\,Q \sum_{p\,\in\,\text{penalty terms}} p + B(t)V.$$

With the penalty strength, Q, set to $O(1/a)$, the spurious degrees of freedom decouple from the theory. For the decomposition of the potential, the prohibited basis states are then “don’t cares,” a term from Boolean logic minimization, meaning that the value of $V(c)$ for the prohibited codes may be adjusted to a value that simplifies the decomposition of V.

In this Hamiltonian, all terms are either just σ^x or products of σ^z (from the expansion of the projection operators), and therefore map to the k-local transverse Ising model. The set of projection operators can be decomposed to a 2-local form using intermediate qubits with penalty terms as is discussed in Appendix A, which then further maps the Schrödinger equation to a 2-local transverse Ising model. The simple form of the H2GC Laplacian raises the possibility of applying this technique with existing hardware systems.

One strategy for evolving to the H2GC Laplacian is to first turn on the penalty via $A(t)$ and later to turn on V via $B(t)$. We provide a demonstration of the time evolution of the H2GC Hamiltonian in Sec. IV C.

Alternatively, since the ground state of the H2GC Laplacian operator is known, one can imagine initializing a future quantum computer with this *a priori* known (and therefore “trivial” in the context of the quantum adiabatic theorem) ground state. Specifically, at first approximation, the n valid sites the ground-state wave function are given by the normalization coefficient, while the invalid sites are then set to zero such that

$$\Psi(t=0,x) = 1/2^{(A+2)/4} \qquad \forall x \in \{\text{H2GC}(i)\},$$
$$\Psi(t=0,x) = 0 \qquad \forall x \notin \{\text{H2GC}(i)\},$$

where $\{i \in \mathbb{Z} | 0 \le i < 2^{A/2+1}\}$. Corrections to this ideal ground state are polynomially suppressed by the strength of the penalty coefficient. The exact form of higher-order correction terms will depend on the penalty cover as shown in Fig. 4 and is beyond the scope of the current work. The time evolution would then depend only on the intermediate ground-state gap induced by V as its contribution is turned on, and importantly, be protected by the IR cutoff of the system, which we discuss in more detail in Sec. IV.

III. POTENTIAL DECOMPOSITION

We demonstrate the method of mapping any local, real potentials sampled at $N = 2^A$ lattice sites in D dimensions. The potential matrix is diagonal and spanned by the product of A-body σ_i^z interactions yielding the complete Walsh basis in encoding G:

$$W_n^G = (\sigma^z)^{\text{bit}(G(n),A-1)} \otimes \cdots \otimes (\sigma^z)^{\text{bit}(G(n),0)}$$
$$= \bigotimes_{i=A-1}^{0} (\sigma^z)^{\text{bit}(G(n),i)}, \tag{18}$$

where $G(n)$ is the nth bit string in G. Note that $\text{bit}(s, A-1)$ is the most significant, i.e., the leftmost, bit of s, and $\text{bit}(s,0)$ is the rightmost bit. Independent of the encoding, the Walsh functions are bijectively mapped to the set of k-local Ising models with $k \le A$, and, indeed, any local potential can be represented in this fashion. This conclusion can be reached by recognizing the Walsh basis as the digitized version of the Fourier series. While the focus of this work are quantum adiabatic compuations, we would like to emphasize that this bijection reduces the problem of constructing the minimal depth quantum circuit for a given error tolerance, for an arbitrary diagonal unitary operator $e^{if(\hat{x})}$, to that of finding the minimal length Walsh-series approximation of the exponent $f(x)$ [20]. Details of the Walsh basis and their relationship to the Ising model and Fourier series are provided in Appendix B.

Typically, given a local potential $V(x)$ in the continuum, its discrete version $V^{\text{bin}}[m]$ is an array obtained through sampling the continuum at successive lattice spacings a:

$$V^{\text{bin}}[m] = V^{\text{cont.}}(ma). \tag{19}$$

In order to correctly evaluate the Schrödinger equation however, one must encode the position of the discretized potential in the same encoding G as the Laplacian operator $L^{(A,G)}$,

$$V^{G}[G(m)] = V^{\text{bin}}[m], \tag{20}$$

For example, if we want to solve the Schrödinger equation in BRGC with $L^{(A,\text{BRGC})}$, then a two-qubit potential will take the form

$$V^{\text{BRGC}}[0] = V^{\text{BRGC}}[\text{BRGC}(0)] = V^{\text{bin}}[0],$$
$$V^{\text{BRGC}}[1] = V^{\text{BRGC}}[\text{BRGC}(1)] = V^{\text{bin}}[1],$$
$$V^{\text{BRGC}}[2] = V^{\text{BRGC}}[\text{BRGC}(3)] = V^{\text{bin}}[3],$$
$$V^{\text{BRGC}}[3] = V^{\text{BRGC}}[\text{BRGC}(2)] = V^{\text{bin}}[2],$$

because the positions $[0,1,2,3]$ are encoded in BRGC as $[00,01,11,10]$ and reinterpreted as binary numbers to $[0,1,3,2]$ in the same way the Laplacian is encoded.

After being encoded, the potential can then be expanded in W_n^G. Typically, the inner product with each basis element is taken, resulting in a series expansion. However, in practice, this is computationally expensive as, for $N = 2^A$, $\mathrm{O}(N^2)$ operations are required. To speed up the decomposition, the fast Wash-Hadamard transform (FWHT) is employed to reduce the complexity to $\mathrm{O}(N\log(N))$ operations. This transform expands any real potential in W_n^G to a given order.

As a consequence of the various codes available, the FWHT is also representation dependent. However, the chosen representation in this case is immaterial and yields only a remapping of the Walsh functions. In particular, given a system of A qubits, one obtains the same set of basis operators. For example, if one works in the binary representation, what is labeled W_2^{bin} will simply be bijectively remapped to W_3^{BRGC} as discussed in Appendix B, while the resulting k-local Ising model stays unchanged. In this paper, we use the sequency ordered transformation $\text{FWHT}^{\text{seq}_A}$ [21,22] because the subscript label in $W_n^{\text{seq}_A}$ can be interpreted as the sequency [23] of the basis function and is therefore the choice that mimics the Fourier-series mode expansion.

In summary, the steps of mapping the potential to the qubit or spin Hamiltonian are (1) discretize the potential to a given lattice, (2) map the potential array to the same code as the Laplacian, (3) decompose the mapped potential using FWHT, (4) map the resulting series expansion to the k-local Ising model.

A. Potential coarse graining

While the FWHT reduces the complexity of decomposition, the cost still scales exponentially with respect to the number of qubits. To further reduce the setup cost, we opt to employ coarse-graining methods. If the features of the potential are on a scale that is much larger than the lattice spacing, then one expects a low-mode expansion to be a sufficient representation of the potential. As a result, given a coarse-graining scale $a^{\text{CG}} \geq a$, the complexity of the FWHT becomes $\mathrm{O}(N^{\text{CG}}\log(N^{\text{CG}}))$ where $N^{\text{CG}} = (L/a^{\text{CG}})^D$ in D dimensions can be exponentially smaller than the original lattice. Such a strategy allows one to decouple the setup cost from the lattice size for a suitable potential.

In this work we explore two coarse-graining strategies: averaging and decimation. In both cases, we define the coarse-grained lattice, $N^{\text{CG}} = 2^{A^{\text{CG}}}$ where $A^{\text{CG}} < A$.

In averaging, we block average the potential between a given interval. This approach has the benefit of obtaining exactly the same coefficients as in the complete expansion with the high-sequency modes $A^{\text{CG}} < r \leq A$ set to zero, serving as a low-pass filter. Therefore, for suitable potentials, averaging introduces only a series truncation error that is well behaved, in the sense that the coefficients of higher-sequency contributions are at least polynomially suppressed. If we have the functional form of the potential, one can analytically compute the indefinite integral and construct averages for the result, and would be computationally cheap to carry out. If the potential does not have an analytic form, one would require sampling at all grid locations making the computational complexity of averaging $\mathrm{O}(2^A)$, and can become prohibitively expensive.

One approach to coarse-grain potentials without analytic forms is to sample only N^{CG} equidistant points through decimation. Unlike averaging however, decimation introduces an uncontrolled error in the values of the coefficients in addition to the series truncation error. Multiple coarse-graining scales will need to be studied to numerically demonstrate that decimation is under control. Nevertheless, the computational cost of decimation can be made negligibly small for any potential.

B. Example: *S*-wave deuteron potential

As an illustration, in Fig. 5(a), we plot both the potential and its low mode expansions from both strategies. We construct a simple S-channel smooth hard core plus well nucleon-nucleon potential that roughly mimics the form of the well-known Argonne v_{18} potential [24]. The height of the hard core and the depth of the well are tuned to reproduce the deuteron binding energy in infinite volume. The potential has the functional form

$$V_{NN}(r) = E_{\text{core}}e^{-(r/R_{\text{core}})^4} - E_{\text{well}}e^{-(r/R_{\text{well}})^4}, \tag{21}$$

where E and R are free parameters tuned to experimental data parameterizing the height and radius of the core ($r \lesssim 0.3$ fm) and well ($r \gtrsim 0.3$ fm). Details of the free parameters are given in Fig. 5(a).

As demonstrated, the Walsh expansion is a very effective representation in both approaches. The difference between averaging and decimation decreases rapidly as A^{CG} increases. The difference from the continuum potential is summarized in Fig. 5(b), where the L_1-normed error per degree-of-freedom is shown to decrease exponentially given a linear increase in A^{CG}.

The potential is encoded in the BRGC representation for this example. Once encoded, it is then expanded by the aforementioned $\text{FWHT}^{\text{seq}_A}$. As an illustration, we provide the Walsh expansion for the averaging scheme depicted in Fig. 5 for $A^{\text{CG}} = 4$,

$$\begin{aligned} V_{NN} &\xrightarrow{\text{discretize}} V_{NN}^{\text{bin}} \xrightarrow{\text{BRGC}} V_{NN}^{\text{BRGC}} \\ &= 129\sum_{i=0}^{3} W_i^{\text{seq}_4} + 129\sum_{i=4}^{7} W_i^{\text{seq}_4} \\ &\quad + 135\sum_{i=8}^{10} W_i^{\text{seq}_4} + 135\sum_{i=11}^{13} W_i^{\text{seq}_4}, \end{aligned}$$

where, for brevity, we round to integer values in the decomposition. Since we have the functional form of

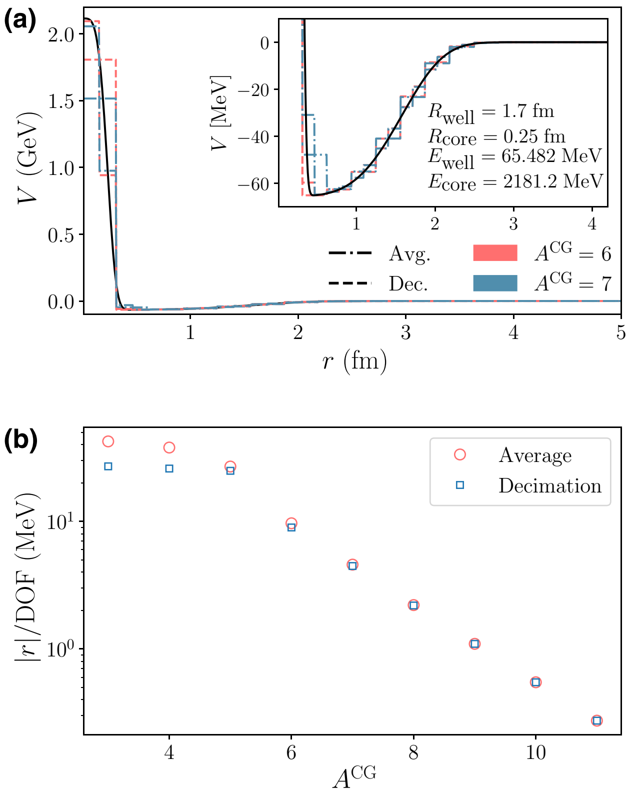


FIG. 5. (a) The S-wave potential Eq. (21), with parameters in the inset tuned to reproduce the deuteron binding energy (solid black line), and the respective discretized potentials on coarse-grained lattices. The inset graph enlarges into the bottom of the well near $r = 0.5$ fm to show how the discretization approximates the potential there. (b) The L_1-normed error per degree of freedom (DOF) of the coarse-grained potential as a function of the number of qubits used to represent the coarse-grained lattice. As A^{CG} increases, the L_1-normed difference from the continuum potential exponentially decreases.

the potential given by, Eq. (21), the averaging can be performed analytically.

The resulting potential decomposition to the k-local Ising model is

$$\begin{aligned} H_V^{\mathrm{BRGC}} &= 129\left(I + \sigma_3^z + \sigma_3^z\sigma_2^z + \sigma_2^z\right)\\ &\quad + 129\left(\sigma_2^z\sigma_1^z + \sigma_3^z\sigma_2^z\sigma_1^z + \sigma_3^z\sigma_1^z + \sigma_1^z\right)\\ &\quad + 135\left(\sigma_1^z\sigma_0^z + \sigma_3^z\sigma_1^z\sigma_0^z + \sigma_3^z\sigma_2^z\sigma_1^z\sigma_0^z + \sigma_2^z\sigma_1^z\sigma_0^z\right)\\ &\quad + 135\left(\sigma_2^z\sigma_0^z + \sigma_3^z\sigma_2^z\sigma_0^z + \sigma_3^z\sigma_0^z + \sigma_0^z\right), \end{aligned}$$

which can be obtained by inspection from the bijective map between the Walsh functions and the k-local Ising model. The k-local terms can be decomposed to a binary tree of ancillary qubits constructed with 2-local terms [14,20] and in Appendix A. We comment that the deuteron potential requires a relatively large basis to describe because the 2 GeV hard core is δ-function-like, and poses a challenge for series-expansion methods.

IV. ADIABATIC QUANTUM-COMPUTING SIMULATIONS

Adiabatic quantum computation (AQC) solves for the ground state of a complex Hamiltonian by starting from the known ground state of a trivial Hamiltonian and adiabatically evolving the initial Hamiltonian to the final target [25–27]. AQC is an alternative paradigm for realizing universal quantum computation. Quantum annealing hardware is the closest to an implementation of AQC. It solves problems where the initial Hamiltonian is the transverse field, and the final Hamiltonian is restricted to be a 2-local Ising model. One goal of this work is to tailor our algorithm to be implemented with as few extensions to existing hardware as possible in the hope that new generations of hardware will incorporate them. In particular, the application of BRGC eliminates the necessity of σ^y. It requires only the addition of a 2-local XZ coupling. The application of H2GC allows simulations to proceed via the transverse Ising model and in principle, could be implemented today if the transverse field were allowed to persist throughout the evolution.

In this section we simulate the following time-dependent Hamiltonian:

$$H(s) = L + B(s)V, \tag{22}$$

$$\Psi(0) = (|\uparrow\rangle + |\downarrow\rangle)^{\otimes A}, \tag{23}$$

where L is the Laplacian defined in Sec. II. Here, $a = L/2^A$ is the lattice spacing and V is the potential from Sec. III.

The time dependence comes from $B(s)$, whose argument s is a dimensionless coefficient with normalized evolution time $s = t/T$ with total evolution time T such that $s \in [0, 1]$. The initial wave function $\Psi(0)$ is an equal superposition state in any encoding of the Laplacian. In particular, the ground state of L for binary encoding and BRGC is exactly the same as the transverse Hamiltonian $H^x = \sum_i \sigma_i^x$, which is seen to be the zero-frequency plane-wave solution. It follows that adiabatic evolution to $H(1)$ prepares the qubits into the ground state of a given quantum system, which becomes the starting point for a time-dependent Schrödinger simulation.

In Ref. [25] the authors compare quantum and statistical annealing for the transverse Ising model using three annealing schedules; linear, square root, and the logarithmic form. They found that the logarithmic annealing schedule keeps the wave function closest to the instantaneous ground state (with the largest overlap). In this work, we chose the schedule based on recent developments in understanding adiabatic schedules [28,29]. Following Ref. [29], we employ a schedule with vanishing gradients at the boundary,

$$B(s) = \frac{\int_0^s \exp\left(\frac{-1}{s'(1-s')}\right) ds'}{\int_0^1 \exp\left(\frac{-1}{s'(1-s')}\right) ds'} \tag{24}$$

for all simulations presented in this work. Additional optimizations to the schedule warrant further investigation [30–32], but are beyond the scope of this work.

The total evolution time T can be roughly estimated from the IR cutoff of the theory given by the box size. In particular, the energy gap between the ground state and first excited state of the free field equation with periodic boundary conditions is

$$\delta E = \frac{(2\pi)^2}{2mL^2} \gg 1/T, \tag{25}$$

where m is the (reduced) particle mass, and L is the length of the finite box. In the examples below, we find setting T to be approximately 2 orders of magnitude longer than $1/\delta E$ is sufficient to evolve the system adiabatically. We note that there exist proofs of rigorous bounds for the quantum adiabatic theorem [28,33], but when applied to examples discussed later in this section, the rigorous bounds overestimate the required time by several orders of magnitude when compared to both Eq. (25) and observations from the corresponding numerical simulations. Additional investigation of tighter theoretical adiabatic bounds for Hamiltonian simulation is important but beyond this work's scope.

In the other extreme, the UV cutoff is regularized by the lattice spacing and given by $1/a$. The critical role of the UV cutoff in the application of the H2GC Laplacian is discussed in Sec. IV C.

All simulations have been performed with `QbSim`, a quantum bit simulator. `QbSim` performs real or imaginary time simulation of qubit systems where the Hamiltonian is expressed as

$$H(t) = \sum_i B_i(t) H_i. \tag{26}$$

The $B_i(t)$ functions are scalar weight functions implementing time dependence. The H_i components are expressed as sums of products of Pauli matrix operators and composites like projection operators P_i^v. A Python integration is used to configure the simulation and access the system's evolving state as time is advanced. A higher-order Dyson series expansion with automatic step size control generates the state evolution. Calculations take place in a fully parallel way. With GPU acceleration, runtimes are reasonable for 20+ qubit systems. We intend to write a separate document describing `QbSim`, and make it available for broader use.

In the sections that follow, we continue the discussion of the S-wave deuteron potential mapped using BRGC as a time-independent application, followed by an example of a time-dependent calculation in two dimensions for a quartic potential in BRGC, and conclude with a simple harmonic oscillator mapped to H2GC.

A. Example: *S*-wave nucleon potential with BRGC

We simulate quantum adiabatic evolution for the potential discussed in Sec. III. The grid size is chosen to be $N = 2^7$ based on the results displayed in Fig. 5. The S-wave potential is fitted to reproduce the deuteron binding energy with a reduced mass of 469.14 MeV, and vanishes at a distance $r \approx 5$ fm. Because the system is weakly bound it extends much further in radius. The box size is set to $L = 20$ fm to properly represent the interacting system's ground state.

In Fig. 6 we show the evolution of the system as a function of total evolution times ranging from $(1\ \text{MeV})^{-1}$ to $(50\ \text{keV})^{-1}$. Given a lattice box size of 20 fm, the lowest

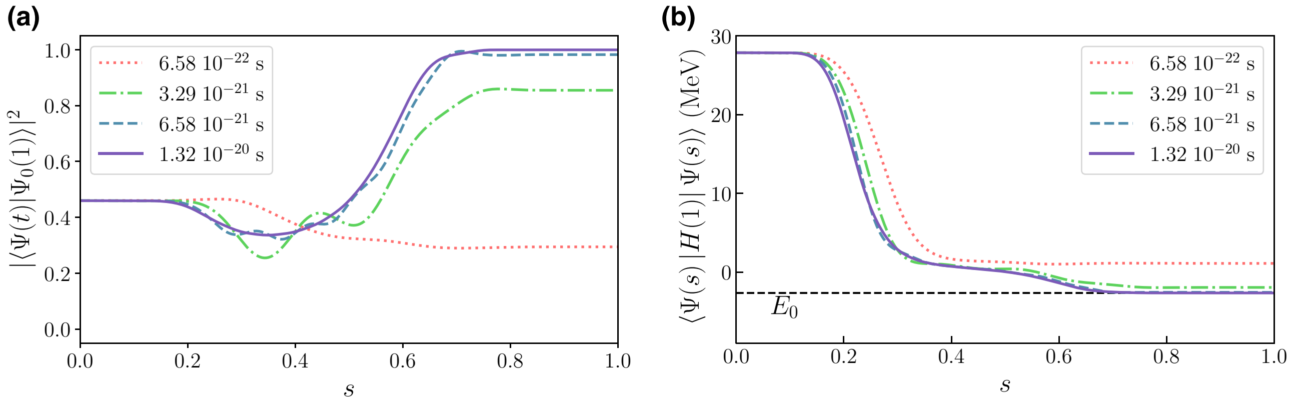


FIG. 6. Adiabatic evolution from a free particle to the interacting system with the S-wave potential of Sec. III. (a) Overlap with the true ground state and expectation of the interacting Hamiltonian as a function of total evolution time. (b) The ground-state energy as a function of total evolution time. The dashed black line marks the physical deuteron binding energy of −2.2 MeV.

nonzero momentum state is approximately a 4-MeV excitation above the ground state. We observe that at a total evolution time of $(100\ \mathrm{keV})^{-1}$ recovers the ground state at the end of the evolution with 98% probability, and at $(50\ \mathrm{keV})^{-1}$ the probability increases to 99.97%. These values are roughly 2 orders of magnitude longer than the estimate of Eq. (25).

While changes in the schedule will affect the result, we observe numerical evidence for physical systems that the IR cutoff of the theory sets the scale for adiabatic evolution. As a result, for a physical system, the total evolution time required for adiabatic state preparation is expected to scale polynomially with the box size, while exhibiting constant scaling with respect to the lattice discretization, which governs the ultraviolet cutoff. This claim is further supported by studying the instantaneous energy spectrum during the evolution as shown in Fig. 7. We observe throughout the entire evolution that the ground-state to the first excited-state energy gap remains of the same scale as the IR cutoff, only subject to small changes even when the system is undergoing the nontrivial change of introducing a 2-GeV hard-core potential. The fundamental reason why the energy gap is so well protected, even against significant changes in the potential, is that the kinetic energy is quantized within a finite box. This is a significantly different situation than the typical quantum annealing application, in which the transverse field is progressively switched off during the evolution.

A classical determination of the ground state is more challenging than one might expect. Because of the large difference in scale between the hard-core height and the binding energy, numerical differential equation solvers are unstable, requiring extra precision and care to find the ground states. A more straightforward technique is to pick a large discrete basis such as more than O(100) states in a harmonic oscillator basis or a similar number of points in a discrete position basis. One then takes matrix elements in that basis and diagonalizes. The large basis is required to simultaneously resolve the spatially tiny hard core and represent the wave function at the long-range associated with the small binding energy. The runtime of partial diagonalization with techniques like the Lanczos algorithm is a function of N, the number of basis states, and the number c of matrix-vector product iterations required, taking $O(cN^3)$ for a dense matrix. In contrast, with the position encoding here, $A = 7$ qubits yield $N = 128$ basis states.

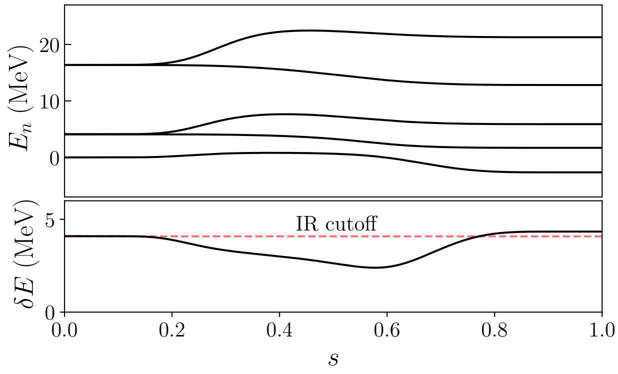


FIG. 7. (Top) The instantaneous spectrum during the adiabatic evolution for the first five states. At $s = 0$, the free-field Laplacian has a twofold degeneracy for periodic boundary conditions. The NN potential lifts the degeneracy afterwards. (Bottom) The energy gap between the first excited state and ground state as a function of evolution time s. The dashed red line is calculated from Eq. (25), yielding the infrared cutoff.

B. Example: two-dimensional quartic potential with BRGC

In the section we demonstrate the ability to evaluate potentials beyond one dimension. The example performs adiabatic evolutions in two stages. Starting from the free particle Hamiltonian, we first evolve the system into a quartic potential as an example of initial state preparation, followed by the introduction of an additional quadratic potential,

$$\begin{aligned} V^{(\mathrm{quartic})}(x,y) &= V_4(x^2+y^2)^2, \\ V^{(\mathrm{quadratic})}(x,y) &= -V_2(x^2+y^2), \end{aligned} \tag{27}$$

where, $V_4 = 10$ MeV and $V_2 = 100$ MeV. As a consequence of turning on the quadratic potential, the ground-state wave function, which was originally centered around the origin deforms into a ring.

The effective mass is set to $m = 1$ GeV in a box that is $L = 10$ fm long in each direction. We opt for $N = 2^6$ lattice points per dimension (for a total of 12 qubits) yielding a lattice spacing of $a \sim 0.15$ fm resulting in a UV cutoff of approximately 1.3 GeV. Correspondingly, the IR cutoff is approximately 8 MeV given the box size and particle mass. As a result, we set the evolution time of going from free field to the quartic potential as $T_1 = (50\ \mathrm{keV})^{-1}$, while $T_2 = (100\ \mathrm{keV})^{-1}$ is used as the evolution time for ramping up the quadratic potential. For purposes of separating the two parts of the evolution, we normalize the evolution time of the first half to $s_1 \in [0, 1]$, which governs the initial state preparation of the interacting system with a quartic potential. In the second part of the simulation, $s_2 \in [1, 1.5]$ the quadratic potential is turned on gradually.

In Fig. 8(a), we show the time-dependent energy of the system. Due to the long evolution time, we observe that the system reaches the correct ground state for the quartic potential at $s = 1$. The system then proceeds with the addition of a quadratic potential and reaches its new ground-state energy. Figure 8(b) shows the evolution of the low-lying spectrum of the system. Similar to the deuteron example in Sec. IV A, we observe that the minimum energy gap stays well protected by the IR cutoff.

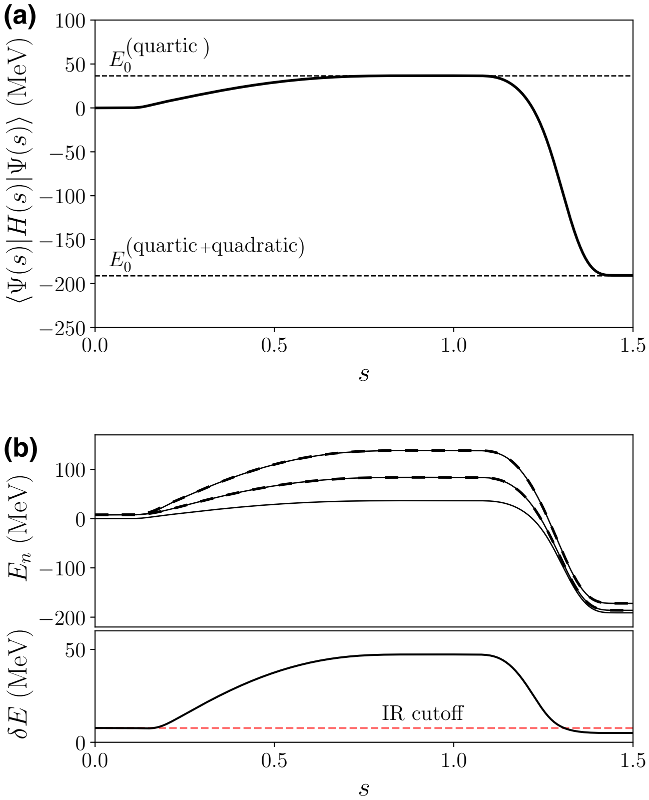


FIG. 8. (a) Instantaneous energy of the time-evolved wave function as a function of evolution time s. The dashed line labeled by $E_0^{(\text{quartic})}$ shows where the ground-state energy of the system with the quartic potential lies. The dashed line labeled by $E_0^{(\text{quartic+quadratic})}$ shows where the ground-state energy lies with the addition of the quadratic potential. (b) (Top) The energy spectrum of the first five eigenstates of $H(s)$, and (bottom) energy gap between the ground state and the first excited state as functions of evolution time. The dashed red line is calculated from Eq. (25), yielding the infrared cutoff.

Additionally, in Fig. 9 we show snapshots of the wave function at $s = 1$ and $s = 1.5$, illustrating the phase transition effects. The wave function can, in principle, be obtained through repeated measurements of the qubits at the end of the evolution.

C. Example: harmonic oscillator with H2GC

In this section, we perform a calculation using the H2GC Laplacian. The final target potential is that of the simple harmonic oscillator

$$V(x) = \frac{1}{2}mx^2. \tag{28}$$

In this example, we work in dimensionless units for simplicity, setting the particle mass m to 10 to confine low-lying states to the box and avoid finite volume effects. We work in a symmetric box ranging from $L = [-1, 1]$.

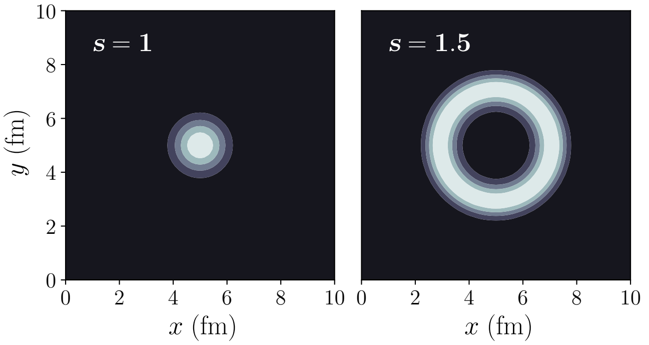


FIG. 9. The probability density from the instantaneous wave function at $s = 1$ of the quartic potential, and $s = 1.5$ with the addition of a quadratic potential.

With $A = 8$, H2GC has 32 valid codes, yielding a lattice spacing of $a = 2/32 = 0.0625$. As a result, the IR cutoff is $(2\pi)^2/2mL^2 = 0.4934$, and the UV cutoff is 16. Because the Laplacian matrix is multiplied by $1/Ma^2$ in the Schrödinger equation, we use the scale $U \equiv 1/Ma^2$ to measure the penalty strength Q needed to suppress the invalid codes in H2GC.

Our time-evolution strategy keeps the transverse-field constant during evolution using the schedule function defined in Eq. (24) to introduce the penalty Hamiltonian, followed by a second delayed schedule to introduce the harmonic oscillator potential. Figure 10(a) shows the time-dependent weight $B(s)$ for the three different contributions to the Hamiltonian. The total time required for adiabatic evolution can be roughly estimated by considering the dynamics of two stages: (1) transverse Hamiltonian to H2GC Laplacian and (2) H2GC Laplacian to the harmonic oscillator system.

Given a system of qubits in the ground state of the Laplacian operator, the evolution time required to turn on the harmonic oscillator potential follows the reasoning from previous sections and is some multiple (e.g., 100×) the IR scale, which in this example still holds. However, the physics governing the adiabatic evolution to the H2GC Laplacian from the transverse Hamiltonian is dominated by the UV cutoff, and must therefore scale as a function of U. This is because the penalty Hamiltonian coefficient needs to be many orders of magnitude above the UV scale to preserve the Laplacian's eigenspectrum. As a result, the penalty Hamiltonian significantly ramps up as a function of s and requires a commensurate amount of evolution time. In simulations, we set the evolution time of the first stage to equal the penalty Hamiltonian coefficient. Applying this logic, the time complexity of adiabatically evolving from the transverse Hamiltonian to the H2GC Laplacian scales exponentially poorly with the number of qubits, since the lattice spacing approaches the continuum exponentially quickly due to the exponential growth of the length of the H2GC code. Nevertheless, the H2GC Laplacian provides

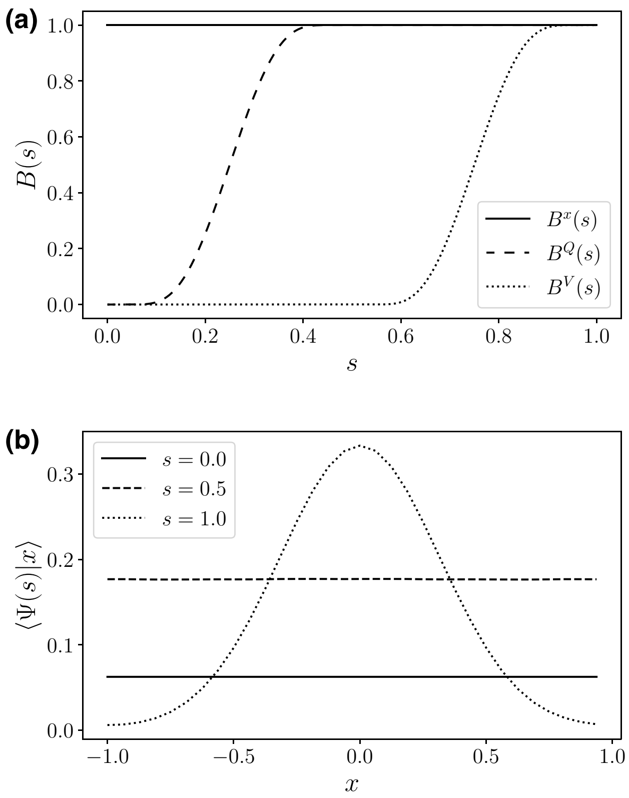


FIG. 10. (a) The time-dependent coefficient $B^x(s)$ for the transverse Hamiltonian H^x, $B^Q(s)$ for the penalty Hamiltonian $Q\sum_c P(c)$, and $B^V(s)$ for the potential. (b) The ground-state eigenvector of $H(s)$ for when (solid line) $s = 0$ and the system is governed by H^x, (dashed line) $s = 0.5$ when the system is the H2GC Laplacian, and (dotted line) $s = 1$ where the harmonic oscillator potential is present.

a way to implement the Schrödinger equation on hardware very similar to systems available today, with the additional requirement to retain the contribution of the transverse field throughout the whole evolution.

For a better understanding of what is happening to the wave function, we provide the ground-state eigenvectors obtained from direct diagonalization in Fig. 10(b). We observe that when $H(s)$ is the transverse Hamiltonian, the ground state is the properly normalized superposition state. With eight qubits, the normalization factor is $\sqrt{1/2^8} = 0.0625$, as indicated by the solid black line. When the penalty Hamiltonian is fully engaged, the H2GC Laplacian will be in its zero-energy ground state. When properly normalized over 32 positions, we see a constant wave function at $\sqrt{1/32} = 0.177$ as shown by the dashed line. After introducing the harmonic oscillator potential, the wave function becomes the expected Gaussian, as demonstrated by the dotted line.

One can further infer the dynamics of the system by studying the time-dependent spectrum of the system shown in Fig. 11(a). In a system of eight qubits, we observe that the system exhibits an eightfold degeneracy in the first excited state, as is expected from the transverse Hamiltonian. We plot the ninth (dashed odd state) excited state to confirm there are no additional degeneracies. When the penalty Hamiltonian is introduced, the eightfold degeneracy evolves into the expected tower of twofold degeneracies for the Laplacian operator in a periodic box.

Figure 11(b) further demonstrates why the prohibited codes must be cleanly gapped from the rest of the system. In this plot, we show the time-dependent energy gap between the ground state and first excited state. We expect that the free-field Laplacian has a gap given by the IR cutoff, while deviations from the red line arise only from interactions with the potential. After increasing the penalty coefficient to $100U$, the H2GC Laplacian starts to reproduce the expected gap within 1% [Fig. 11(b)].

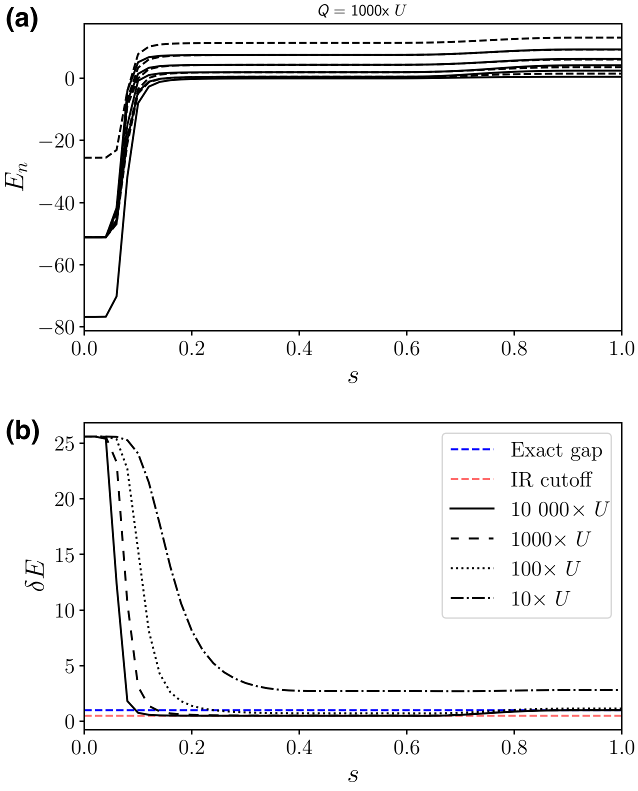


FIG. 11. (a) The time-dependent energy spectrum of $H(s)$ of the first ten states for a system of eight qubits. The solid lines denote even-numbered states (0, 2, 4, 6, 8), and dashed lines label odd-numbered states (1, 3, 5, 7, 9). (b) The time-dependent energy gap between the first excited state and ground state as a function of evolution time s and strength of the penalty coefficient Q. The dotted, dashed, and solid lines denote progressively stronger penalty coefficients, set relative to the U scale. The dashed red line is calculated from Eq. (25), yielding the infrared cutoff.

D. Example: He atom with two electrons

For a final demonstration we include an example with two independent particles. We work in units of electronvolts (eV) and nanometers (nm). We model two electrons, distinguished by spin, around a helium nucleus at the origin in a periodic three-dimensional volume $b = 0.128$ nm on each side. The volume size is approximately 4 times the Bohr radius $r_b = 0.031$ nm found from an effective central charge of approximately 1.69 due to partial shielding by the other electron. Each electron spatial direction is BRGC encoded with three qubits for a total of 18 qubits (262144 basis states) and the the multiparticle Laplacian is implemented following Eq. (11), summing the Laplacian on the first nine with that of the last nine qubits. A bare Coulomb potential would yield an infinite sum over periodic images, so a Yukawa potential (in natural units)

$$V(r) = Z_1 Z_2 \alpha \frac{e^{-r/b}}{r + r_E} \tag{29}$$

is used for the interactions between the electrons and between the nucleus and electrons. The parameters Z_1 and Z_2 are the charges for particle 1 and 2, respectively, where $Z = -1$ for the electrons and $Z = +2$ for the helium nucleus. The parameter $\alpha \sim 1/137$ is the fine-structure constant, and $r_E = 1.6\ fm$ is the helium-nucleus charge radius. Contributions from neighboring periodic images are included but suppressed by the exponential in the Yukawa potential.

We first use Lanczos diagonalization to solve for the ground state, yielding a total binding energy of 74.5 eV versus the experimental value of approximately 79 eV. Given the small number of qubits, the eigenvalue is sensitive to the volume size as well as the inclusion of an exponential factor in the potential and a close match is not expected. We then evolve with AQC from the initial transverse state with $H_0 = T$ to $H = T + V$.

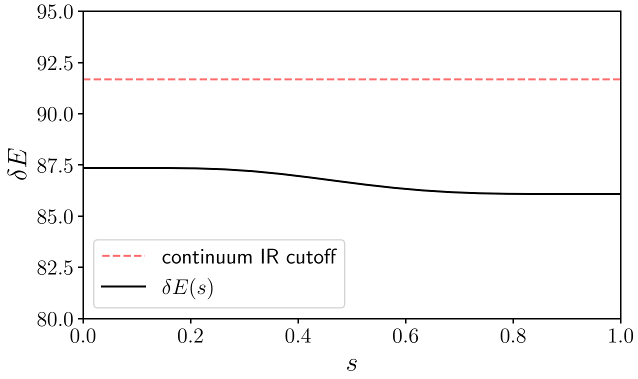


FIG. 12. The time-dependent energy gap for $H(s)$ given by Eq. (22) (black solid line), where the two-particle Yukawa potential in three dimensions Eq. (29) is used. The corresponding energy gap of the free-field system given by Eq. (25) (dotted red line) differs from the gap at $H(0)$ with a correction of O(a^2) at leading order, and is visible due to the relatively coarse lattice spacing used in this example.

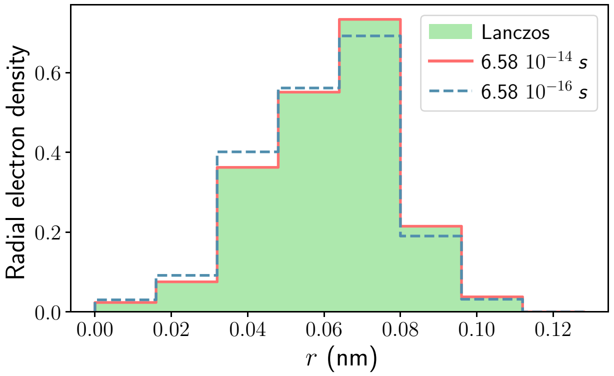


FIG. 13. The radial electron density obtained with AQC for $T \sim 1/\delta E$ (dotted blue) and $T \sim 100/\delta E$ (solid red), compared to the density obtained by diagonalization (green histogram) for the He atom, where δE is the energy gap between the ground state and first excited states of the free-field solution.

Figure 12 shows the time-dependent energy gap between the ground state and first excited state. The dotted red line is the IR cutoff predicted by Eq. (25) given by a 0.128-fm box. Due to the coarse discretization used in this example, the IR cutoff predicted in the continuum limit differs (inconsequentially for our purposes) by approximately 5% from the gap of the discretized Laplacian. More importantly, we observe that the time-dependent energy gap is again, protected by the IR cutoff and therefore is expected to retain polynomial time complexity with respect to increasing system size.

In Fig. 13 we compare the density determined by AQC to that determined from diagonalization. We observe that when a total evolution time of 100× the IR cutoff is used for AQC, the electron density reproduces the result from exact diagonalization within a fraction of a percent. We highlight here the observation that the adiabatic bound of O$(100/\delta E)$ is consistent with all other examples presented in this work.

V. SUMMARY AND CONCLUSION

The Schrödinger equation remains one of the foundational blocks of our understanding of quantum systems. One common method of solving this equation is by discretization in a selected basis and it has found wide applications in classical computations. We introduced the concept of encoding in the association of positions with A-body qubit states in the computational basis. Such an association provides an exponential improvement in space complexity, and encoding further reduces the Hamiltonian complexity. With the limitations of current adiabatic quantum computers in mind, we demonstrate the power of Gray encoding in simplifying the spin Hamiltonian representation. With a BRGC encoding of positions, we mapped the Schrödinger equation with a real, local potential to the *XZ* model, and through H2GC we further simplified it to a transverse Ising model while maintaining an exponential improvement in space complexity and a quadratic count of 3-local penalty terms for unused codes. These advantages also apply to the implementation of ladder operators. More specifically, with H2GC, the generic Schrödinger equation with a local potential discretized on $2^{A/2+1}$ points is equivalent to a transverse Ising model on A qubits and a quadratic in A count of 3-local penalty terms.

In both BRGC and H2GC cases, we employed the FWHT to efficiently encode the potential as an Ising Hamiltonian and showed that coarse-graining techniques could further reduce the computational cost of encoding.

Through numerical simulations, we discovered that the system's adiabatic evolution is stable due to the infrared cutoff associated with finite volume. By borrowing techniques successfully used in lattice QCD computations with finite range interactions, e.g., Luscher's method [34], for extracting infinite volume results from finite volume ones, we can envisage performing quantum calculations in finite volume and benefiting from the polynomial time scaling associated with finite volume for computing observables.

For all codes, evolution from a free field to an interacting system exhibits polynomial time complexity with volume and constant scaling with respect to lattice discretization. For H2GC, if the evolution begins with the transverse Hamiltonian followed by the introduction of penalties to keep the low-lying spectrum of the Laplacian intact, the time evolution will initially be sensitive to the ultraviolet scale. This sensitivity will give rise to polynomial time complexity with lattice discretization.

ACKNOWLEDGMENTS

We thank Dong An and Alessandro Roggero for useful discussions and suggestions. Lawrence Berkeley National Laboratory (LBNL) is operated by The Regents of the University of California (UC) for the U.S. Department of Energy (DOE) under Federal Prime Agreement DE-AC02-05CH11231. This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Nuclear Physics, Quantum Horizons: QIS Research and Innovation for Nuclear Science under Award Number FWP-NQISCCAWL (CCC, KSM, YW). E.R. acknowledges the NSF N3AS Physics Frontier Center, NSF Grant No. PHY-2020275, and the Heising-Simons Foundation (2017-228). Y.W. is grateful for mentorship from his advisor Roberto Car, and acknowledges support from the DOE Award DE-SC0017865.

APPENDIX A: PROJECTION OPERATOR PRODUCT REDUCTION

Current hardware topology requires the qubit or spin Hamiltonian to have 2-local interactions. For Eq. (17), fortunately, there is a known method [14] to replace a product of P_i^0s to the projection operator of a single qubit while keeping the low-lying spectrum of the Hamiltonian, at the expense of adding ancillary qubits [35]. Because each iteration of the recursion formula shares all but one projection operator with its predecessor, the ancillary bit cost is linear in A. We include a brief discussion of the construction of the reduction.

To reduce the correction terms to 2-local, it is sufficient to reduce the product of projection operators to the projection operator of a single ancillary qubit. If we can reduce two qubits to one, then a tree or chain of such reductions will suffice. We construct the 2-to-1 reduction by adding qubit a along with a penalty contribution to the Hamiltonian. To replace $P_i^0P_j^0$, for example, a penalty is added when $P_a^0 \neq P_i^0P_j^0$. The penalty is shown for all states of qubits a, i, and j in Table I. We use a Karnaugh map, shown in Fig. 14, to visualize the adjacencies and assist in minimizing the implementation of the penalty Hamiltonian.

The resulting penalty Hamiltonian is

$$H_{\text{pen}} = Q\left[P_a^1 + \sigma_a^z\left(P_i^1 + P_j^1\right) + P_i^1P_j^1\right]. \tag{A1}$$

All pieces of this penalty contribution are 2-local.

To reduce a collection of projector products to 2-local, an efficient heuristic is to rank projector pairs by the number of existing products in which they appear. Then, the highest-ranked such pair is processed, producing a new qubit, and the projector on the new qubit replaces the pair in every product in which it appears, repeating the process until completion. This process is a well-known heuristic for reducing collections of multi-input boolean and gates. The pair's tree height can also be included as a negative

TABLE I. A diagonal penalty Hamiltonian for reducing the product of two-qubit projection operators p_i^0 and p_j^0 to a single qubit labeled q that is 0 when the original qubits are both 0. The Q indicates a sufficient penalty to remove violators from the low-lying states, not a specific value.

a	P_a^0	i	j	$P_i^0P_j^0$	H_{pen}
0	1	0	0	1	0
0	1	0	1	0	Q
0	1	1	0	0	Q
0	1	1	1	0	Q
1	0	0	0	1	Q
1	0	0	1	0	0
1	0	1	0	0	0
1	0	1	1	0	0

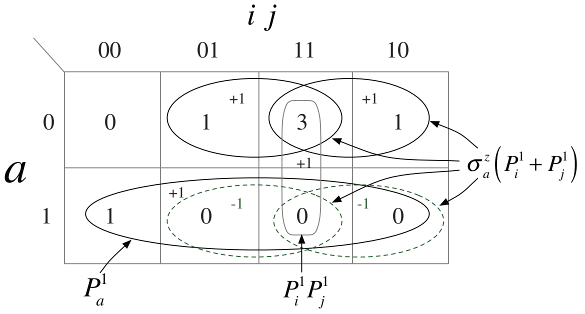


FIG. 14. Penalties from Table I appear here as nonzero values in the center of the cells, which will be multiplied by Q. Contributions are shown as ovals with $+1$ or -1 indicating the coefficient for the contribution. The sum of such values for all ovals covering the center of a cell totals to the penalty value of the cell. When restricted to the first row, $P^1_i + P^1_j$ generates the two ovals in the row. The same contribution with the opposite phase is needed in the bottom row, shown with green dashed ovals. $\sigma^z_a\left(P^1_i + P^1_j\right)$ generates all four contributions. A value of 1 is added in the bottom row with P^1_a, leaving only cell 1/11 needing a correction, which is provided by $P^1_i P^1_j$. This process leaves cell 0/11 with a larger than needed but acceptable penalty of $3Q$.

contribution in the ranking to avoid long chains of ancillary qubits.

APPENDIX B: REVIEW OF THE ORTHOGONAL FUNCTIONS

1. Walsh series

The tensor product space of Pauli σ^z matrices form a complete basis for real-valued functions and is analogous to a digitized Fourier series expansion. This can be understood by recognizing that there is a one-to-one mapping of the σ^z tensor products to the Walsh series, which we discuss in this section.

a. Walsh and Rademacher functions

Before defining the Walsh functions, let us first introduce the Rademacher functions R_n, which are the basic building blocks of the basis states. The functions R_n are defined as

$$R_n(t) = \operatorname{sign}\sin(2^{n+1}\pi t), \tag{B1}$$

where t spans the unit interval, and n is the set of natural numbers starting from zero. We can immediately interpret R_n as the set of periodic step functions with integer frequencies as enforced by periodic boundary conditions.

In an encoding G, the Walsh functions are constructed from the Rademacher functions such that

$$W^G_0 \equiv 1, \tag{B2}$$

$$W^G_n = \Pi_{n_i} R_{n_i}, \tag{B3}$$

where the set $\{n_i\}$ is composed of the positional indices of the nonzero bits in $G(n)$. The bit-string representation of n is read from left to right. For example, if $G(n) = 110$, then $\{n_i\} = \{0, 1\}$, and the corresponding Walsh function is $R_0(t)R_1(t)$. By definition, W^G_0 corresponds to the zero-frequency mode in all encodings. For a more concrete discussion, we list the first 2^3 integers in binary, BRGC, sequency, and H2GC encoding

$$\begin{array}{ccccccc} \begin{pmatrix} 0\\1\\2\\3\\4\\5\\6\\7 \end{pmatrix} & \xrightarrow[\text{rep}]{\text{bin}} & \begin{pmatrix} 000\\001\\010\\011\\100\\101\\110\\111 \end{pmatrix} & \begin{pmatrix} 000\\001\\011\\010\\110\\111\\101\\100 \end{pmatrix} & \begin{pmatrix} 000\\100\\110\\010\\011\\111\\101\\001 \end{pmatrix} & \begin{pmatrix} 0000\\0001\\0011\\0111\\1111\\1110\\1100\\0100 \end{pmatrix} \\ \text{int} & & \text{bin} & \text{BRGC} & \text{seq}_3 & \text{H2GC} \end{array} \tag{B4}$$

The binary order is also called the Hadamard order in the literature, the BRGC order follows from Gray code discussed in Sec. II B 1, sequency order is also called the Walsh order in the literature and is just the reflection of the BRGC order for a given number of bits, and finally the H2GC sequence is discussed in Sec. II C and is used to encode the Laplacian with the transverse Ising model Hamiltonian.

b. Binary order

The Walsh functions W^{bin}_n in binary order are denoted by a superscript bin. Following Eq. (B4), we give the first three Walsh functions in binary order to illustrate the construction

$$\begin{aligned} 1 \rightarrow 001 \qquad & \therefore W^{\text{bin}}_1 = R_2, \\ 2 \rightarrow 010 \qquad & \therefore W^{\text{bin}}_2 = R_1, \\ 3 \rightarrow 011 \qquad & \therefore W^{\text{bin}}_3 = R_1 R_2. \end{aligned}$$

c. Binary reflected Gray order

An alternative way to order the Walsh functions is to map the sequence to BRGC, and is the computational ordering for the XZ-model mapping of the Schrödinger equation. We use the notation W^{BRGC}_n to denote the Gray-ordered Walsh function. Following Eq. (B4), the first three Walsh functions in Gray order are

$$\begin{aligned} 1 \rightarrow 001 \qquad & \therefore W^{\text{BRGC}}_1 = R_2, \\ 2 \rightarrow 011 \qquad & \therefore W^{\text{BRGC}}_2 = R_1 R_2, \\ 3 \rightarrow 010 \qquad & \therefore W^{\text{BRGC}}_3 = R_1. \end{aligned}$$

d. Sequency order

The sequency order is analogous to the Fourier-series mode expansion, and was the version originally employed by Walsh [36]. In this order, each function has one more zero crossing than the previous function and the set alternates between even and odd functions sequentially. From this perspective, it is very similar to the Fourier series and the concept of frequency is replaced by sequency. The list of sequency bit strings are obtained by performing a bit reversal on the BRGC bit strings. Due to bit reversal, the sequency order mapping is dependent on the total size of the system A. We use the notation $W_n^{\text{seq}_A}$ to denote the sequency-ordered Walsh functions for an A (qu)bit system. Following Eq. (B4), the first three Walsh functions in sequency order for a 3-bit system are

$$
\begin{aligned}
1 &\rightarrow 100 \qquad \therefore W_1^{\text{seq}_3} = R_0,\\
2 &\rightarrow 110 \qquad \therefore W_2^{\text{seq}_3} = R_0 R_1,\\
3 &\rightarrow 010 \qquad \therefore W_3^{\text{seq}_3} = R_1.
\end{aligned}
$$

As a result, low-mode expansions can be computed successively one contribution at a time given the above sequency order. In Sec. III we suggest using a combination of coarse graining and the fast Walsh transform (similar to the Fourier version) in order to gain a substantial computational speed up when series expanding arbitrary real functions. Therefore, this discussion of the sequency ordering is meant to give better intuition for the Walsh series, and are important when discussing the series expansion for potentials. The Walsh functions in the sequency order are also given by the rows of the Hadamard matrix,

$$
\begin{aligned}
H(2^k) &= \begin{pmatrix} H(2^{k-1}) & H(2^{k-1}) \\ H(2^{k-1}) & -H(2^{k-1}) \end{pmatrix},\\
H(2^1) &= \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.
\end{aligned}
\tag{B5}
$$

Then, $W_n^{\text{bin}} = n^{\text{th}}$ row of $H(2^k)$.

e. Mapping to the Pauli basis

For a system of A qubits, the first 2^A Rademacher functions have an exact mapping to the diagonal of the 1-local σ^z Hamiltonian, $R_n \rightarrow \sigma^z_{A-1-n}$. The first three Rademacher functions are shown in Fig. 15 along with the corresponding 1-local Hamiltonian for a system of three qubits.

It follows immediately that given a system of A qubits, the set of 2^A, k-local Ising-like Hamiltonians are bijectively mapped to the first 2^A Walsh functions. For example, in a system of three qubits, the $n = 4$ Walsh function in binary order is given by

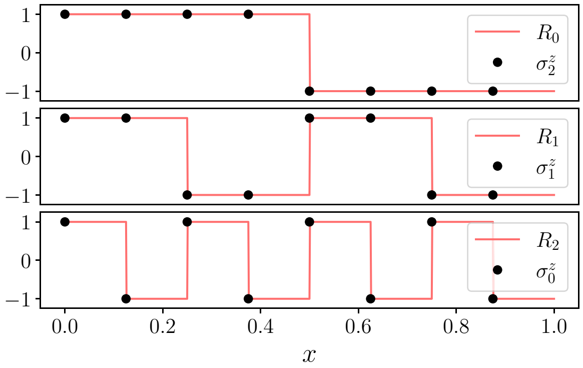


FIG. 15. The first three Rademacher functions.

$$
W_4^{\text{bin}} = R_2 \rightarrow \sigma_0^z = \mathbb{1} \otimes \mathbb{1} \otimes \sigma^z \tag{B6}
$$

and in Gray order as

$$
W_4^{\text{BRGC}} = R_0 R_1 \rightarrow \sigma_2^z \sigma_1^z = \sigma^z \otimes \sigma^z \otimes \mathbb{1} \tag{B7}
$$

and in Walsh order as

$$
W_4^{\text{seq}_3} = R_1 R_2 \rightarrow \sigma_1^z \sigma_0^z = \mathbb{1} \otimes \sigma^z \otimes \sigma^z. \tag{B8}
$$

In general, given a binary representation for an integer n, the 1s and 0s map, respectively, to tensor products of σ^z and $\mathbb{1}$. Figure 16 shows the first seven Walsh functions in sequency order and highlights the connection to sine and cosine functions with increasing frequency.

2. Fast Walsh-Hadamard transform

In analogy with the Fourier series, the Walsh functions in a given order form an orthonormal basis for the vector space of functions defined on $[0, 1]$. As already touched upon, differently from the Fourier series, there are many versions of the Walsh series depending on how the functions are ordered and sequency most closely resembles the concept of frequency with each subsequent element in the series increasing the number of zero crossings by one. The expansion can be performed through the inner product,

$$
\begin{aligned}
f(x) &= \sum_n a_n W_n(x),\\
a_n &= \int_0^1 f(x) W_n(x).
\end{aligned}
\tag{B9}
$$

On a discretized domain of $N = 2^A$ equally spaced grid points, the Walsh-Hadamard transform can be easily realized from Eq. (B5),

$$
\boldsymbol{f}^{(W)} = \frac{1}{2^A} H\left(2^A\right) \boldsymbol{f}, \tag{B10}
$$

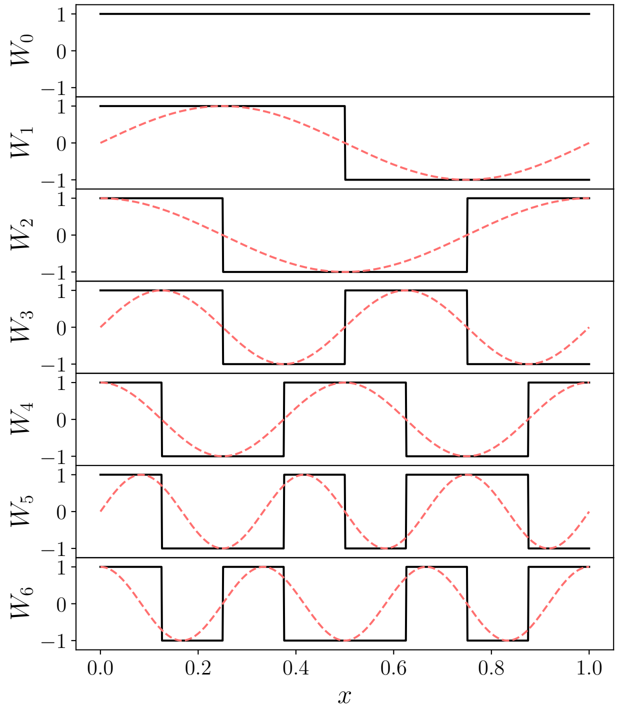


FIG. 16. The first seven Walsh functions in sequency order.

where the real function, $\boldsymbol{f} = \{f(x_i)\}_{i=0}^{2^A-1}$, has been evaluated at the grid points. This transformation requires N^2 operations, just like the discrete Fourier transform (DFT), and indeed is equivalent to a multidimensional DFT of size 2^A [37]. In practice, one opts for an efficient implementation like the fast Fourier transform (FFT) [38]. This is achieved by the FWHT, which requires $N\log(N)$ operations. Through the decades, various fast algorithms have been developed, which automatically return the expansion in a given order. As an illustration, here we provide the decompostion in binary order for a sequence of four grid points by matrix partitioning techniques [39], where

$$4\begin{pmatrix} f^{(W,b)}(x_0)\\ f^{(W,b)}(x_1)\\ f^{(W,b)}(x_2)\\ f^{(W,b)}(x_3)\end{pmatrix} = H(4)\begin{pmatrix} f(x_0)\\ f(x_1)\\ f(x_2)\\ f(x_3)\end{pmatrix}$$

$$= \begin{pmatrix} H(2) & H(2)\\ H(2) & -H(2)\end{pmatrix}\begin{pmatrix} f(x_0)\\ f(x_1)\\ f(x_2)\\ f(x_3)\end{pmatrix}$$

partitions to

$$4\begin{pmatrix} f^{(W,b)}(x_0)\\ f^{(W,b)}(x_1)\end{pmatrix} = H(2)\begin{pmatrix} f_1(x_0)\\ f_1(x_1)\end{pmatrix}$$
$$= H(2)\begin{pmatrix} f(x_0)+f(x_2)\\ f(x_1)+f(x_3)\end{pmatrix},$$

$$4\begin{pmatrix} f^{(W,b)}(x_2)\\ f^{(W,b)}(x_3)\end{pmatrix} = H(2)\begin{pmatrix} f_1(x_2)\\ f_1(x_3)\end{pmatrix}$$
$$= H(2)\begin{pmatrix} f(x_0)-f(x_2)\\ f(x_1)-f(x_3)\end{pmatrix}$$

which can be further partitioned into

$$\begin{aligned} 4f^{(W,b)}(x_0) &= f_2(x_0) = (f_1(x_0)+f_2(x_1)),\\ 4f^{(W,b)}(x_1) &= f_2(x_1) = (f_1(x_0)-f_2(x_1)),\\ 4f^{(W,b)}(x_2) &= f_2(x_2) = (f_1(x_2)+f_2(x_3)),\\ 4f^{(W,b)}(x_3) &= f_2(x_3) = (f_1(x_2)-f_2(x_3)). \end{aligned}$$

APPENDIX C: DERIVATION OF EQ. (17) IN TENSOR PRODUCT NOTATION

We start from the recursion formula of $L^{(A,\text{bin})}$ in Eq. (12):

$$L^{(A,\text{bin})} = \mathbb{1}\otimes(L^{(A-1,\text{bin})} - C_{A-1}) + \sigma^x \otimes C_{A-1}. \quad \text{(C1)}$$

We follow the notation in Sec. II B 1. For an integer $N = 2^A$, the encoding function, G_A, of BRGC is a permutation of $(0,1,2,\ldots,N-1)$. It is defined inductively $G_1 = (0,1)$. For $A > 1$, the first half of G_A is G_{A-1}, and the second half of G_A is G_{A-1} reversed in order and then added by 2^{A-1}. For example, G_2 is $(0,1)$ concatenated with $(1+2, 0+2)$, which is $(0,1,3,2)$. In particular,

$$\begin{aligned} &G_A(0) = 0\\ &G_A(2^A-1) = 2^{A-1} + G_A(0) = 2^{A-1}\\ &G_A(2^{A-1}-1) = 2^{A-2}\\ &G_A(2^{A-1}) = 2^{A-1} + G_A(2^{A-1}-1) = 2^{A-1} + 2^{A-2} \end{aligned} \quad . \quad \text{(C2)}$$

Let the matrix transformation in Eq. (15) be denoted by $\mathscr{G}_A$:

$$L^{(A,\text{BRGC})} \equiv \mathscr{G}_A(L^{(A,\text{bin})}). \quad \text{(C3)}$$

To derive a formula for $L^{(A,\text{BRGC})}$, we first note that $\mathscr{G}_A(\mathbb{1}\otimes M) = \mathbb{1}\otimes\mathscr{G}_{A-1}(M)$ if M is invariant under the reflection permutation, $(A-2, A-3, \ldots, 0)$, and that $L^{(A-1,\text{bin})}$ and C^{A-1} both enjoy this invariance. Thus,

$$\begin{aligned} L^{(A,\text{BRGC})} &= \mathbb{1}\otimes L^{(A-1,\text{BRGC})} - \mathbb{1}\otimes\mathscr{G}_{A-1}(C_{A-1})\\ &\quad + \mathscr{G}_A(\sigma^x\otimes C_{A-1}). \end{aligned} \quad \text{(C4)}$$

To compute $\mathscr{G}_A(C_A)$, note that $(C_A)_{ij}$ is nonzero if $(i,j) = (0, 2^A-1)$ or $(2^A-1, 0)$. Thus, $\mathscr{G}_A(C_A)$ is nonzero at

$(0, 2^{A-1})$ and $(2^{A-1}, 0)$. For example, for $A = 2$,

$$\mathscr{G}_2(C_2) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \sigma^x \otimes P^0, \tag{C5}$$

where $P^0 = (1 - \sigma^z)/2$ is a z-projection matrix. It is easy to see that, for general A,

$$\mathscr{G}_A(C_A) = \sigma^x \otimes (P^0)^{\otimes(A-1)}. \tag{C6}$$

To compute $\mathscr{G}_A(\sigma^x \otimes C_{A-1})$, we note that $(\sigma^x \otimes C_{A-1})_{ij}$ is nonzero at $(i,j) = (2^A - 1, 0)$, $(0, 2^A - 1)$, $(2^{A-1}, 2^{A-1} - 1)$, and $(2^{A-1} - 1, 2^{A-1})$. According to Eq. (C2), this means that $\mathscr{G}_A(\sigma^x \otimes C_{A-1})$ is nonzero at $(2^{A-1}, 0)$, $(0, 2^{A-1})$, $(2^{A-1} + 2^{A-2}, 2^{A-2})$, and $(2^{A-2}, 2^{A-1} + 2^{A-2})$. For example, for $A = 3$,

$$\begin{aligned} \mathscr{G}_3(\sigma^x \otimes C_2) &= \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\ &= \sigma^x \otimes \mathbb{1} \otimes P^0. \end{aligned} \tag{C7}$$

For general n, we see that

$$\mathscr{G}_A(\sigma^x \otimes C_{A-1}) = \sigma^x \otimes \mathbb{1} \otimes (P^0)^{\otimes(A-2)}. \tag{C8}$$

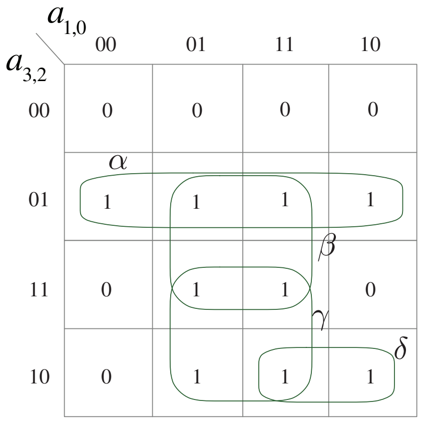
Thus, we obtain

$$\begin{aligned} L^{(A,\mathrm{BRGC})} &= \mathbb{1} \otimes L^{(A-1,\mathrm{BRGC})} + (\sigma^x \otimes \mathbb{1} - \mathbb{1} \otimes \sigma^x) \\ &\quad \times \otimes (P^0)^{\otimes(A-2)}, \end{aligned} \tag{C9}$$

where the base case is $L_2 = \mathbb{1} \otimes \sigma^x + \sigma^x \otimes \mathbb{1}$.

APPENDIX D: INTRODUCTION TO KARNAUGH MAPS

Karnaugh maps [10] are a tool for visualizing binary hypercubes with dimension ≥ 3. A common use of them in classical boolean circuit design is as an aid in minimization of boolean functions as a sum of products or a product of sums. The cells in a Karnaugh map represent the corners of a binary hypercube in a way that makes it easy to visually identify subcubes of the complete hypercube. Figure 17 shows a Karnaugh map for a function of four variables. Subcubes are important because they can also be specified as a boolean product of boolean literals (a literal is a boolean variable or its complement) for variables that do not change in the subcube. In the quantum-computing context the parallel specification is a product of projection operators on the qubits whose values are constant in the subcube. In Fig. 17 there are four groups indicated with labels α, β, γ, and δ. Each of these groups is a subcube of the full four-variable hypercube. We can write a simple product expression for each group. Note that the bar over variables is a standard way to indicate a boolean complement



$a_{3,2}$ \ $a_{1,0}$	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	0	1	1	0
10	0	1	1	1

FIG. 17. A Karnaugh map on four input variables. The 16 cells correspond to the corners of a dimension-four binary hypercube, each of which is associated with specific values of the input variables. The input values can be read from the left of the row and the opt of the column the cell is in. Each cell contains the boolean function value for the input state of the cell. The greek letter labels have been added for the discussion but are not normally part of a Karnaugh map. The leftmost 1 under the label α is in a cell with input values 0100. Note that the column and row input values follow the sequence 00, 01, 11, 10, which has the property that visually adjacent cells differ in the value of only one input variable. We also consider the right-hand edge to be adjacent to the left-hand edge and the bottom edge to be adjacent to the top edge.

$$\alpha = \bar{a}_3 a_2, \ \beta = a_2 a_0, \ \gamma = a_3 a_0, \ \delta = a_3 \bar{a}_2 a_1. \tag{D1}$$

In using a Karnaugh map for optimization of a sum of products one first finds the set of maximal subcubes containing only 1s. The indicated groups are maximial because removal of any variable from their product expression would enlarge them to include a cell with a 0. Next, a subset of the subcubes are selected such that every 1 is inside one of the selected subcubes. In this example subcube β is unnecessary because all the contained 1s are covered by subcubes α and γ, which is easily checked by examination of the Karnaugh map. The complete boolean function f is then

$$f \ = \bar{a}_3 a_2 + a_3 a_0 + a_3 \bar{a}_2 a_1. \tag{D2}$$

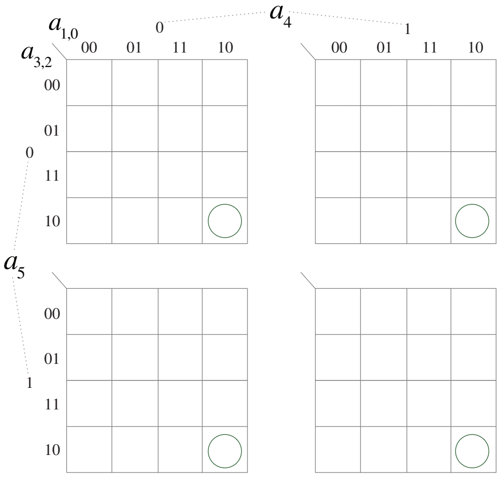


FIG. 18. A Karnaugh map on six input variables. One should regard the submaps as separate pieces of paper stacked up in the order $a_{5,4} = 00, 01, 11, 10$, in clockwise order. The top and bottom sheets are adjacent because of periodicity. A cell is considered adjacent to the cell in the sheet above or below it in addition to adjacencies in the four-variable map. As an example, a two-variable subcube with four cells can be formed from the collection of the lower-right cells of all four submaps, with inputs $a_{3,2,1,0} = 1010$ or $a_3\bar{a}_2a_1\bar{a}_0$.

In our quantum-computing application to penalties the sum operation is a numeric sum instead of a boolean sum (also known as a boolean or). For penalties however, we do not care about the exact penalty value and two terms of the same sign may both contribute without harm, giving the same character as the boolean sum.

The simple Karnaugh map in Fig. 17 can be extended to more variables in a hierarchical way. Figure 18 shows a six-input map, which can be thought of as a two-variable map with each cell containing a four-variable map. An eight-input map can be constructed as a four-variable map with each cell being in turn a four-variable map.

[1] S. Abel, N. Chancellor, and M. Spannowsky, Quantum computing for quantum tunneling, Phys. Rev. D **103**, 016008 (2021).

[2] G. Pilon, N. Gugole, and N. Massarenti, Aircraft loading optimization – qubo models under multiple constraints, arXiv:2102.09621 [quant-ph] (2021).

[3] P. Mocz and A. Szasz, Towards cosmological simulations of dark matter on quantum computers, arXiv:2101.05821 (2021).

[4] N. P. D. Sawaya, T. Menke, T. H. Kyaw, S. Johri, A. Aspuru-Guzik, and G. G. Guerreschi, Resource-efficient digital quantum simulation of d-level systems for photonic, vibrational, and spin-s Hamiltonians, Npj Quantum Inf. **6**, 49 (2020).

[5] O. Di Matteo, A. McCoy, P. Gysbers, T. Miyagi, R. M. Woloshyn, and P. Navrátil, Improving Hamiltonian encodings with the Gray code, arXiv:2008.05012 [quant-ph] (2020).

[6] M. Born and R. Oppenheimer, Zur Quantentheorie der Molekeln, Ann. Phys. **389**, 457 (1927).

[7] R. Car and M. Parrinello, Unified Approach for Molecular Dynamics and Density-Functional Theory, Phys. Rev. Lett. **55**, 2471 (1985).

[8] E. Mátyus, D. J. Wales, and S. C. Althorpe, Quantum tunneling splittings from path-integral molecular dynamics, J. Chem. Phys. **144**, 114108 (2016).

[9] Y. Wu and R. Car, Quantum momentum distribution and quantum entanglement in the deep tunneling regime, J. Chem. Phys. **152**, 024106 (2020).

[10] M. Karnaugh, The map method for synthesis of combinational logic circuits, Trans. Am. Inst. Electr. Eng. Part I: Commun. Electron. **72**, 593 (1953).

[11] K. Rao and N. Ahmed, in *ICASSP '76. IEEE International Conference on Acoustics, Speech, and Signal Processing* (IEEE, New York, NY, USA, 1976), Vol. 1, p. 136.

[12] K. Boothby *et al.*, Architectural considerations in the design of a third-generation superconducting quantum annealing processor, arXiv:2108.02322 [quant-ph] (2021).

[13] The support of an operator is the number of nonidentity Pauli matrices.

[14] A. Perdomo, C. Truncik, I. Tubert-Brohman, G. Rose, and A. Aspuru-Guzik, Construction of model Hamiltonians for adiabatic quantum computation and its application to finding low-energy conformations of lattice protein models, Phys. Rev. A **78**, 012320 (2008).

[15] Note that binary encoding does not meet this condition.

[16] W. H. Kautz, Unit-distance error-checking codes, IRE Trans. Electron. Comput. **EC-7**, 179 (1958).

[17] H. E. Tompkins, Unit-distance binary-decimal codes for two-track commutation, IRE Trans. Electron. Comput. **EC-5**, 139 (1956).

[18] J. Chinal, in *Design Methods for Digital Systems* (Springer Berlin Heidelberg, Berlin, Heidelberg, 1973), p. 44.

[19] H. Abbott and M. Katchalski, Further results on snakes in powers of complete graphs, Discrete Math. **91**, 111 (1991).

[20] J. Welch, D. Greenbaum, S. Mostame, and A. Aspuru-Guzik, Efficient quantum circuits for diagonal unitaries without ancillas, New J. Phys. **16**, 033040 (2014).

[21] W. K. Pratt, J. Kane, and H. C. Andrews, Hadamard transform image coding, Proc. IEEE **57**, 58 (1969).

[22] J. Manz, A sequency-ordered fast Walsh transform, IEEE Trans. Audio Electroacoust. **20**, 204 (1972).

[23] The sequency of a Walsh function is the number of positive zero crossings of that function.

[24] R. B. Wiringa, V. G. J. Stoks, and R. Schiavilla, Accurate nucleon-nucleon potential with charge-independence breaking, Phys. Rev. C **51**, 38 (1995).

[25] T. Kadowaki and H. Nishimori, Quantum annealing in the transverse Ising model, Phys. Rev. E **58**, 5355 (1998).

[26] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, Quantum Computation by Adiabatic Evolution, eprint arXiv:quant-ph/0001106 (2000).

[27] A. Das and B. K. Chakrabarti, Colloquium: Quantum annealing and analog quantum computation, Rev. Mod. Phys. **80**, 1061 (2008).

[28] T. Albash and D. A. Lidar, Adiabatic quantum computation, Rev. Mod. Phys. **90**, 015002 (2018).

[29] D. An and L. Lin, Quantum linear system solver based on time-optimal adiabatic quantum computing and quantum approximate optimization algorithm, arXiv:1909.05500 [quant-ph] (2020).

[30] T. Albash and D. A. Lidar, Demonstration of a Scaling Advantage for a Quantum Annealer over Simulated Annealing, Phys. Rev. X **8**, 031016 (2018).

[31] P. Hauke, H. G. Katzgraber, W. Lechner, H. Nishimori, and W. D. Oliver, Perspectives of quantum annealing: Methods and implementations, Rep. Prog. Phys. **83**, 054401 (2020).

[32] K. Takada, Y. Yamashiro, and H. Nishimori, Mean-field solution of the weak-strong cluster problem for quantum annealing with stoquastic and non-stoquastic catalysts, J. Phys. Soc. Jpn. **89**, 044001 (2020).

[33] S. Jansen, M.-B. Ruskai, and R. Seiler, Bounds for the adiabatic approximation with applications to quantum computation, J. Math. Phys. **48**, 102111 (2007).

[34] M. Lüscher, Volume dependence of the energy spectrum in massive quantum field theories, Commun. Math. Phys. **105**, 153 (1986).

[35] The $\hat{q}_i$ in [14] is P_i^1 here.

[36] J. L. Walsh, A closed set of normal orthogonal functions, Am. J. Math. **45**, 5 (1923).

[37] Kunz, On the equivalence between one-dimensional discrete Walsh-Hadamard and multidimensional discrete fourier transforms, IEEE Trans. Comput. **C-28**, 267 (1979).

[38] J. Cooley and J. Tukey, An algorithm for the machine calculation of complex Fourier series, Math. Comput. **19**, 297 (1965).

[39] N. Ahmed and S. M. Cheng, On matrix partitioning and a class of algorithms, IEEE Trans. Educ. **13**, 103 (1970).