Quantum Algorithms to Simulate Many-Body Physics of Correlated Fermions

Zhang Jiang,[1,2,*] Kevin J. Sung,[3] Kostyantyn Kechedzhi,[1,4] Vadim N. Smelyanskiy,[5] and Sergio Boixo[5]

[1]*NASA Ames Research Center Quantum Artificial Intelligence Laboratory (QuAIL),*
Moffett Field, California 94035, USA
[2]*Stinger Ghaffarian Technologies, Inc., 7701 Greenbelt Road,*
Suite 400, Greenbelt, Maryland 20770, USA
[3]*Department of Electrical Engineering and Computer Science,*
University of Michigan, Ann Arbor, Michigan 48109, USA
[4]*USRA, NASA Ames Research Center, Moffett Field, California 94035, USA*
[5]*Google, Venice, California 90291, USA*

(Received 26 November 2017; revised manuscript received 2 March 2018; published 26 April 2018)

Simulating strongly correlated fermionic systems is notoriously hard on classical computers. An alternative approach, as proposed by Feynman, is to use a quantum computer. We discuss simulating strongly correlated fermionic systems using near-term quantum devices. We focus specifically on two-dimensional (2D) or linear geometry with nearest-neighbor qubit-qubit couplings, typical for superconducting transmon qubit arrays. We improve an existing algorithm to prepare an arbitrary Slater determinant by exploiting a unitary symmetry. We also present a quantum algorithm to prepare an arbitrary fermionic Gaussian state with $O(N^2)$ gates and $O(N)$ circuit depth. Both algorithms are optimal in the sense that the numbers of parameters in the quantum circuits are equal to those describing the quantum states. Furthermore, we propose an algorithm to implement the 2D fermionic Fourier transformation on a 2D qubit array with only $O(N^{1.5})$ gates and $O(\sqrt{N})$ circuit depth, which is the minimum depth required for quantum information to travel across the qubit array. We also present methods to simulate each time step in the evolution of the 2D Fermi-Hubbard model—again on a 2D qubit array—with $O(N)$ gates and $O(\sqrt{N})$ circuit depth. Finally, we discuss how these algorithms can be used to determine the ground-state properties and phase diagrams of strongly correlated quantum systems using the Hubbard model as an example.

DOI: 10.1103/PhysRevApplied.9.044036

I. INTRODUCTION

A large class of materials evade description by density-functional theory [1] due to the effects of strong electron-electron correlations [2,3]. A simulation of strongly correlated electronic structure and dynamical effects in such materials would allow a quantitative prediction of their physical characteristics before fabricating them, which is often costly. This approach would open the route for designing materials with application-specific characteristics. The properties of strongly correlated fermionic systems remain elusive after many years of intensive research. Indeed, solving a general quantum many-body problem without using approximations on a classical computer takes exponential time in the size of the problem.

One way to avoid this difficulty, as envisioned by Feynman, is to use a quantum computer to simulate quantum systems [4,5]. In experiments with real materials, one rarely knows for sure the initial states and the underlying Hamiltonians. A quantum computer, however, enables one to prepare the initial state with confidence and to have full control over the Hamiltonian under which the state evolves. We are closer to Feynman's vision with recent advances in quantum-computing hardware. It is important to develop simulation algorithms that are optimized given the limitations of current and near-term quantum hardware, such as the locality of qubit-qubit couplings, the gate sequences, and the mitigation of errors and noise. Significant progress has been made in this direction [6–9].

We propose several quantum algorithms to simulate correlated fermions on 2D and linear qubit arrays with nearest-neighbor couplings which are typical for superconducting transmon qubits. We use the Fermi-Hubbard model as an example to demonstrate these algorithms. In this simple model, the superconductor–to–Mott-insulator transition is solely determined by the competition between the hopping and the interaction terms. Solving the Hubbard model allows one to single out this mechanism from other effects such as disorder and long-range interactions in real materials.

[*] Present address: Google, Venice, California 90291, USA.
Corresponding author.
qzj@google.com

The Hubbard model [10] approximates the long-range Coulomb interaction of electrons in a crystal with a local on-site interaction. This locality reduces the resources required for simulating the model and makes it a prime candidate for the early applications of quantum simulations [11]. The single-band Fermi-Hubbard model is described by the Hamiltonian

$$\begin{aligned}\mathcal{H}_{\mathrm{FH}} = &-\sum_{\langle j,k\rangle,\sigma} t_{jk}(c^\dagger_{j,\sigma}c_{k,\sigma} + \mathrm{H.c.}) + U\sum_j n_{j,\uparrow}n_{j,\downarrow}\\ &+\sum_{j,\sigma}(\epsilon_j - \mu)n_{j,\sigma} - \sum_j h_j(n_{j,\uparrow} - n_{j,\downarrow}), \qquad (1)\end{aligned}$$

where $c^\dagger_{j,\sigma}$ ($c_{j,\sigma}$) is the creation (annihilation) operator for the jth site with spin σ and $n_{j,\sigma} = c^\dagger_{j,\sigma}c_{j,\sigma}$ is the fermion occupation number operator. The first term on the right side of Eq. (1) describes fermions hopping between sites, the second term describes the on-site interactions, and the remaining two terms describe a local potential and a magnetic field. The model demonstrates a wide range of strongly correlated phenomena, including metal-insulator transitions, unconventional Fermi liquids, and a number of inhomogeneous phases. The Fermi-Hubbard model also provides an approximate description of materials [12] including the cuprate family [13] (albeit a multiband extension of the model is necessary for quantitative correspondence to the materials), which has attracted a lot of interest because of unconventional symmetry-breaking phenomena and high-temperature superconductivity [14].

The one-dimensional (1D) Hubbard model was solved by Lieb and Wu in 1968 [15]; however, a full theoretical analysis of the 2D Hubbard model requires going beyond the validity of mean-field and perturbation theory arguments remains an open question. The 2D Hubbard model serves as a canonical microscopic physical model for strongly correlated fermionic systems. In the underdoped region of its phase diagram, multiple orders exist corresponding to a regime of maximum numerical difficulty. Significant numerical progress has been made in the identification of the unconventional Mott-insulator transition and a superconducting phase with a d-wave order parameter [16,17]. Exact numerical diagonalization is limited to about 20 sites (40 logical qubits) [18,19], which is too small for a finite-size scaling analysis. Approximate methods such as quantum Monte Carlo simulations or many-body theory expansions have been used to simulate systems with hundreds of sites, which allows for extrapolation to the thermodynamic limit. Except at half filling, quantum Monte Carlo methods suffer from sign problems which prevent them from simulating systems at very low temperatures [20]. Application of the density-matrix renormalization group (DMRG) [21] to the 2D Hubbard model requires mapping to an effective 1D problem, to which the standard DMRG is applied. Other numerical methods to determine the phase diagrams of the Hubbard model include the dynamical cluster approximation [22,23] and the density-matrix embedding theory [17,24]. These methods can asymptotically approach the exact solution with increasingly larger clusters; however, simulating those clusters requires an exponential amount of computing resources on a classical computer.

To simulate the Fermi-Hubbard model on a quantum computer, one needs to map the fermionic operators to qubit operators. In the second quantization picture, a particular spin orbital being unoccupied (occupied) can be represented by the qubit state $|0\rangle$ ($|1\rangle$). To retain the fermionic anticommutation relations, one also needs to account for the parities of qubits corresponding to other spin orbitals, e.g., by using the Jordan-Wigner transformation (JWT) [25,26]. The two terms on the second line of Eq. (1) can be implemented using only single-qubit operators, and the on-site interaction term can be implemented with two-qubit interactions. The hopping terms, however, cannot be implemented straightforwardly in more than one spatial dimension because of the nonlocal parity operators in the JWT. It is of practical importance to reduce the depth of the quantum circuits for these terms with only local qubit interactions [9], which is crucial to near-term quantum computers without quantum error correction [27,28]. We will always have local qubit interactions in mind, which is a different model from quantum computation with all-to-all interactions. The digital quantum simulations we consider here could address low-temperature properties of the Hubbard model, which thus far remain hard to access with analog simulators [29]. In addition to simulating unitary dynamics, preparing the ground state is a source of significant overhead in digital quantum simulations. We develop algorithms for state preparations with only locally coupled qubits by adopting the general ideas from Ref. [11].

Beyond the Hubbard model, there are a growing number of materials showing surprising sensitivity to impurities. It is an open issue to understand and, ultimately, to control the emergent phases from impurities [30]. At low temperatures, impurities and disorder can induce a metal-to-insulator transition by localizing the Cooper pairs [31,32]. Impurities, however, can also induce interesting phases of strongly correlated matter and help probe the underlying mechanism of exotic states [33,34]. On the other hand, quantum impurity models are useful for hybrid quantum-classical approaches to strongly correlated materials [35], where the quantum computer solves an impurity problem that is determined self-consistently with the help of a classical computer. Our methods work for systems that violate translational symmetries and thus are suited for simulating impurity models.

In this paper, we propose several quantum algorithms to simulate fermionic systems on near-term devices, e.g., on 2D or linear geometry with nearest-neighbor qubit-qubit couplings. Specifically, we propose quantum algorithms to prepare arbitrary fermionic Gaussian states which can be used as a starting point to model correlated quantum states. The same algorithms can also be used to implement arbitrary fermionic transformations with linear input-output relations. We also provide a state-of-the-art algorithm for 2D fermionic Fourier transformation, where the fermionic parity problem is overcome with negligible overhead. Our algorithms open up the possibility of a wealth of experiments to simulate many-body physics, including the Fermi-Hubbard model, in the noisy intermediate-scale quantum era [36].

The paper is organized in the following way. In Sec. II, we review the topic of mapping fermionic operators to qubit operators and introduce notation. In Sec. III, we improve an existing quantum algorithm in Ref. [11] to prepare an arbitrary Slater determinant by exploiting a unitary symmetry. In Sec. IV, we present an algorithm to prepare an arbitrary fermionic Gaussian state, improving on and generalizing an existing method for translationally invariant systems [37]. Both algorithms in Secs. III and IV are implemented as part of the open-source project OpenFermion [38]. In Sec. V, we introduce a quantum algorithm to implement the 2D fermionic Fourier transformation on a 2D qubit array with only $O(N^{1.5})$ gates and $O(\sqrt{N})$ circuit depth, a better scaling than methods based on the fermionic SWAP gates [9]. In Sec. VI, we discuss applications of our quantum algorithms using the Fermi-Hubbard model as an example. In Appendix A, we review the properties of mean-field Hamiltonians that are quadratic in the fermionic creation and annihilation operators and discuss how to diagonalize these Hamiltonians. In Appendix B, we introduce an alternative approach to solve time evolutions of the quantum lattice models, such as the Hubbard model, by using ancilla-assisted fermionic gates. In Appendix C, we discuss fermionic operations based on Hamiltonian evolution. In Appendix D, we briefly review the fermionic SWAP gate. In Appendix E, we discuss how to simulate the 2D Hubbard model with a qubit ladder using the fermionic SWAP gate. In Appendix F, we study the Trotter errors in adiabatic state preparation numerically for small system sizes.

II. MAPPING FERMIONS TO QUBITS

The first step in solving strongly correlated fermionic systems with a quantum computer is to map the Hilbert space of fermions to the states of qubits. Following Ref. [39], we represent the fermionic Hamiltonian in the second quantization picture using a discrete basis of spin orbitals. A qubit is assigned to each spin orbital; the qubit state $|0\rangle$ ($|1\rangle$) denotes an unoccupied (occupied) spin orbital. The fermionic operators satisfying the anticommutation relations can be mapped to qubit operators using the JWT [25,26], the Bravyi-Kitaev transformation (BKT) [40,41], or the Ball-Verstraete-Cirac transformation (BVCT) [42–44]. To represent the parities, the JWT requires strings of Pauli operators that act on $O(N)$ qubits, the BKT uses $O(\log N)$ nonlocal operators, and the BVCT requires only $O(1)$ local operators by introducing one ancilla qubit per logical qubit. Although the JWT seems to have the worst scaling, it has the most straightforward form and can be applied to 1D fermionic systems without encountering the parity problem. One way to get around the parity problem in the JWT for systems of higher dimensions is to use fermionic SWAP gates [9,45]. We discuss in this paper a different approach based on traveling ancilla qubits, which solves the parity problem in the JWT with negligible overhead.

The JWT maps the fermionic creation and annihilation operators to qubit operators as follows:

$$c_j^\dagger \mapsto \frac{1}{2}(X_j - iY_j)Z_1 \cdots Z_{j-1}, \tag{2a}$$

$$c_j \mapsto \frac{1}{2}(X_j + iY_j)Z_1 \cdots Z_{j-1}, \tag{2b}$$

where X, Y, and Z are the Pauli operators. It assumes an ordering of the qubits and attaches the Pauli-Z operators with smaller indices (the parity) to the raising and lowering operators. A single Slater determinant in the computational basis (Fock state) takes the form

$$c_{j_1}^\dagger \cdots c_{j_{N_f}}^\dagger |\mathrm{vac}\rangle \mapsto \left(\prod_{n=1}^{N_f} \frac{X_{j_n} - iY_{j_n}}{2}\right)|0\cdots 0\rangle, \tag{3}$$

where $j_1 < j_2 \cdots < j_{N_f}$ and $|\mathrm{vac}\rangle$ is the vacuum state. The fermionic number operator

$$c_j^\dagger c_j \mapsto \frac{1}{4}(X_j - iY_j)(X_j + iY_j) = \frac{1}{2}(I - Z_j), \tag{4}$$

where I is the 2×2 identify operator. The fermionic hopping term can be realized by a product of Pauli operators ($k > j$),

$$c_j^\dagger c_k \mapsto \frac{1}{4}(X_j - iY_j)(X_k + iY_k)Z_{j+1} \cdots Z_{k-1}. \tag{5}$$

For the case $k = j + 1$, we have

$$-i(c_j^\dagger c_{j+1} - \mathrm{H.c.}) \mapsto \frac{1}{2}(X_j Y_{j+1} - Y_j X_{j+1}), \tag{6a}$$

$$c_j^\dagger c_{j+1} + \mathrm{H.c.} \mapsto \frac{1}{2}(X_j X_{j+1} + Y_j Y_{j+1}), \tag{6b}$$

interactions which can be implemented efficiently with superconducting qubits [46]. The hopping terms between orbitals encoded far from each other in the Jordan-Wigner transformation, however, are generally hard to implement. In Secs. III and IV, we discuss state preparation algorithms that totally avoid such hopping terms. In Sec. V and Appendix B, we also discuss strategies to implement such terms by introducing ancilla qubits that store the parities.

III. PREPARING SLATER DETERMINANTS

A Slater determinant can be regarded as an eigenstate of a Hamiltonian quadratic in fermionic creation and annihilation operators, or simply as a quadratic Hamiltonian. The standard algorithm to prepare Slater determinants was described in Ref. [25] and improved in Ref. [11] using elementary operations called Givens rotations, which are rotations in the plane spanned by two coordinates axes. The circuit depth and implementability can be improved by parallelizing the Givens rotations restricted to neighboring qubits [45]. Here, we present an algorithm that reduces the total number of Givens rotations by exploiting a freedom in the representation of Slater determinants; PYTHON code for this algorithm is available as part of the OpenFermion project [38].

In the second quantization picture, a single Slater determinant takes the form

$$|\Psi_S\rangle = b_1^\dagger \cdots b_{N_f}^\dagger |\text{vac}\rangle, \qquad b_j^\dagger = \sum_{k=1}^{N} Q_{jk} c_k^\dagger, \tag{7}$$

where Q is an $N_f \times N$ matrix satisfying

$$Q^\dagger Q = P_S, \tag{8}$$

with P_S the projector (of rank N_f) onto the subspace spanned by the single-particle wave functions of the occupied spin orbitals (rows of Q). An arbitrary Slater determinant (7) can be prepared by applying a single-particle basis change $\mathcal{U}$ to an easy-to-prepare determinant in the computational basis (3):

$$|\Psi_S\rangle = \mathcal{U} c_1^\dagger \cdots c_{N_f}^\dagger |\text{vac}\rangle, \qquad \mathcal{U} c_j^\dagger \mathcal{U}^\dagger = b_j^\dagger, \tag{9}$$

for $j = 1, 2, \ldots, N_f$. The unitary $\mathcal{U}$ corresponds to a fermionic Fourier transformation when the rows of Q are orthonormal plane waves. The identity (8) remains true under the transformation $Q \to VQ$, where V is an arbitrary $N_f \times N_f$ unitary matrix. Indeed, the Slater determinant (7) remains the same (up to an overall phase) under the basis transformation V:

$$\left(\prod_{j=1}^{N_f} \sum_{k=1}^{N_f} V_{jk} b_k^\dagger\right) |\text{vac}\rangle = \det(V) |\Psi_S\rangle, \tag{10}$$

where $\det(V)$ is the determinant of V. The unitary V can be chosen to be composed of a sequence of Givens rotations on neighboring rows of Q that bring the matrix elements in its upper right corner to zeros. A Givens rotation takes the following form in the two relevant coordinate axes:

$$\begin{aligned} G(\theta,\varphi) &= \begin{pmatrix} \cos\theta & -e^{i\varphi}\sin\theta \\ \sin\theta & e^{i\varphi}\cos\theta \end{pmatrix} \\ &= \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{pmatrix}, \end{aligned} \tag{11}$$

where we generalize the original definition by allowing complex matrix elements. For example, with $N = 6$ and $N_f = 3$, we can break V into three Givens rotations,

$$\begin{aligned} Q &\to \begin{pmatrix} * & * & * & * & * & \mathbf{0} \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{pmatrix} \to \begin{pmatrix} * & * & * & * & * & 0 \\ * & * & * & * & * & \mathbf{0} \\ * & * & * & * & * & * \end{pmatrix} \\ &\to \begin{pmatrix} * & * & * & * & \mathbf{0} & 0 \\ * & * & * & * & * & 0 \\ * & * & * & * & * & * \end{pmatrix} = VQ, \end{aligned} \tag{12}$$

where $*$ represents an arbitrary matrix element, and the bold matrix elements are zeroed out with Givens rotations. This procedure does not change the Slater determinant (10), and no physical operation is required. However, the procedure makes it clear that the number of physical Givens rotations can be reduced and how this can be achieved.

The unitary in Eq. (9) corresponds to a single-particle basis change and can be decomposed into a sequence of Givens rotations [11],

$$\mathcal{U} = \mathcal{G}_1 \mathcal{G}_2 \cdots \mathcal{G}_{N_\mathcal{G}}. \tag{13}$$

The Givens rotation $\mathcal{G}(\theta,\varphi)$ acting on the jth and kth spin orbitals takes the form

$$\begin{pmatrix} \mathcal{G} c_j^\dagger \mathcal{G}^\dagger \\ \mathcal{G} c_k^\dagger \mathcal{G}^\dagger \end{pmatrix} = G(\theta,\varphi) \begin{pmatrix} c_j^\dagger \\ c_k^\dagger \end{pmatrix}, \tag{14}$$

where the 2×2 matrix $G(\theta,\varphi)$ is defined in Eq. (11). A Givens rotation of the form (14) can be implemented on a quantum computer using the circuit in Fig. 1 (the two qubits are adjacent in the JWT).

The first part of the circuit (the dotted box) describes a rotation between the qubit states $|10\rangle$ and $|01\rangle$ corresponding to a rotation in the single-particle subspace of the two

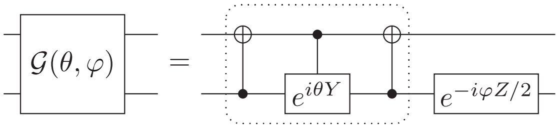

FIG. 1. Quantum circuit for a Givens rotation on neighboring qubits: The part in the dotted box represents a rotation between the two states $|01\rangle$ and $|10\rangle$.

spin orbitals; the two qubit states $|00\rangle$ and $|11\rangle$ corresponding to the unoccupied state and the double-occupied state are unchanged in this step. The fermionic unitary $\mathcal{U}$ in Eq. (13) can be represented by a matrix U corresponding to a single-particle basis change (see Appendix A),

$$\mathcal{U}\mathbf{c}^{\dagger}\mathcal{U}^{\dagger} = U\mathbf{c}^{\dagger}, \qquad U = G_{N_G}\cdots G_2 G_1, \tag{15}$$

where $\mathbf{c}^{\dagger} = (c_1^{\dagger}\cdots c_N^{\dagger})^T$. The order of the Givens rotations is reversed in Eq. (15) compared to Eq. (13); this is the case because the matrix G acts directly on the vector of fermionic creation operators in Eq. (14), and it should be placed on the right side of all matrices representing earlier rotations.

To perform the transformation in Eq. (9), we require that the first N_f rows of U be equal to VQ; see also Appendix A. With the example $N = 6$ and $N_f = 3$, this condition is equivalent to

$$VQU^{\dagger} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}. \tag{16}$$

A sequence of Givens rotations—acting on adjacent columns of VQ (corresponding to neighboring qubits in the JWT)—can achieve the desired form as follows:

$$\begin{aligned} VQ &\to \begin{pmatrix} * & * & * & \mathbf{0} & 0 & 0 \\ * & * & * & * & * & 0 \\ * & * & * & * & * & * \end{pmatrix} \to \begin{pmatrix} * & * & \mathbf{0} & 0 & 0 & 0 \\ * & * & * & * & \mathbf{0} & 0 \\ * & * & * & * & * & * \end{pmatrix} \\ &\to \begin{pmatrix} \underline{\lambda_1} & \mathbf{0} & 0 & 0 & 0 & 0 \\ \underline{0} & * & * & \mathbf{0} & 0 & 0 \\ \underline{0} & * & * & * & * & \mathbf{0} \end{pmatrix} \to \begin{pmatrix} \lambda_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \underline{\lambda_2} & \mathbf{0} & 0 & 0 & 0 \\ 0 & \underline{0} & * & * & \mathbf{0} & 0 \end{pmatrix} \\ &\to \begin{pmatrix} \lambda_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \underline{\lambda_3} & \mathbf{0} & 0 & 0 \end{pmatrix} \to VQU^{\dagger}, \end{aligned} \tag{17}$$

where the λ_j are phase factors, i.e., $|\lambda_j| = 1$. The bold matrix elements are zeroed out in each step; Givens rotations on nonoverlapping columns can be implemented in parallel [45]. The underlined ones become zeros or phase factors automatically due to the orthonormality of the rows. The phase factors are brought to ones in the last step by single-qubit rotations; this step is unnecessary if the goal is to prepare a single Slater determinant. The total number of Givens rotations needed for the transformation U is

$$\begin{aligned} N_G &= NN_f - N_f(N_f - 1)/2 - N_f(N_f + 1)/2 \\ &= (N - N_f)N_f, \end{aligned} \tag{18}$$

and the circuit depth is $N - 1$. Our algorithm requires only $N^2/4$ Givens rotations in the worst case, when $N_f = N/2$. By comparison, direct implementations without using the unitary freedom V require more Givens rotations [11,45]. We also point out that the trick to reducing N_G by interchanging the roles of particles and holes [11] is no longer needed here; i.e., particles and holes are treated on an equal footing in our algorithm.

In summary, we describe in this section a method to prepare a Slater determinant (7) using two-qubit gates that act only on neighboring qubits. Our method can be achieved in four steps:

(1) Store the wave functions of the occupied orbitals in the rows of the matrix Q.

(2) Zero out the upper-right matrix elements of Q using the freedom $Q \to VQ$.

(3) Diagonalize VQ using a sequence of Givens rotations as column transformations.

(4) Find the quantum gates that correspond to the Givens rotations in step 3.

As mentioned above, this algorithm is implemented as part of the open-source project OpenFermion [38]. The code takes as input the matrix Q from Eq. (7) describing a Slater determinant and outputs a sequence of elements of the form (j, k, θ, ϕ), which describes a Givens rotation of columns j and k. Furthermore, rotations that can be performed in parallel are grouped together. We use the code to verify that the method described here does produce the desired Slater determinant.

IV. PREPARING FERMIONIC GAUSSIAN STATES

Fermionic Gaussian states [47,48] can be regarded as a generalization of Slater determinants obtained by relaxing the constraint that the total number of particles be fixed. The celebrated Bardeen-Cooper-Schrieffer (BCS) wave function [49] for superconductivity is a special case of fermionic Gaussian states. Verstraete *et al.* [37] demonstrated how to prepare the ground state of a BCS-like Hamiltonian—for 1D translationally invariant systems—using the fermionic fast Fourier transformation and the two-mode Bogoliubov transformation. Recently, superpositions of fermionic Gaussian states were used to approximate low-energy states of quantum impurity models [50], which can be useful in a quantum-classical hybrid scheme for correlated materials [35]. Simulating quantum systems with disorder on a quantum computer may also lead to a better understanding of and, ultimately, control over the emergent phases from impurities [30].

Here, we discuss how to prepare an arbitrary fermionic Gaussian state as the ground state of a quadratic Hamiltonian,

$$\mathcal{H} = \sum_{j,k=1}^{N} M_{jk} c_j^{\dagger} c_k + \frac{1}{2}\sum_{j,k=1}^{N}\left(\Delta_{jk} c_j^{\dagger} c_k^{\dagger} + \text{H.c.}\right), \tag{19}$$

where $M = M^\dagger$ and $\Delta = -\Delta^T$ are complex matrices. PYTHON code for this algorithm is available as part of the OpenFermion project [38]. Our method can also be used to implement an arbitrary fermionic Gaussian unitary. In Appendix A, we review the standard approach to bringing the Hamiltonian (19) into the diagonal form,

$$\mathcal{H} = \sum_{j=1}^{N} \varepsilon_j b_j^\dagger b_j + c \text{ number}, \tag{20}$$

where $0 \le \varepsilon_1 \le \varepsilon_2 \cdots \le \varepsilon_N$, and b_j and $b_j^\dagger$ are a new set of fermionic operators that satisfy the canonical anticommutation relations. The new fermionic operators are linear combinations of the original ones:

$$\begin{pmatrix} \mathbf{b}^\dagger \\ \mathbf{b} \end{pmatrix} = W \begin{pmatrix} \mathbf{c}^\dagger \\ \mathbf{c} \end{pmatrix} = \begin{pmatrix} \mathcal{W}\mathbf{c}^\dagger \mathcal{W}^\dagger \\ \mathcal{W}\mathbf{c}\mathcal{W}^\dagger \end{pmatrix}, \tag{21}$$

where $(\mathbf{c}^\dagger \quad \mathbf{c})^T = (c_1^\dagger \cdots c_N^\dagger \, c_1 \cdots c_N)^T$ and $(\mathbf{b}^\dagger \quad \mathbf{b})^T = (b_1^\dagger \cdots b_N^\dagger \, b_1 \cdots b_N)^T$. The fermionic Gaussian unitary $\mathcal{W}$ performs the linear transformation W, and the ground state of the quadratic Hamiltonian (19) is

$$|\Psi_0\rangle = \mathcal{W}|\text{vac}\rangle = \mathcal{U}|\text{vac}\rangle, \tag{22}$$

where $c_j|\text{vac}\rangle = 0$ for $j = 1, 2, \ldots, N$, and $\mathcal{U} = \mathcal{W}\mathcal{V}$ for some single-particle basis transformation $\mathcal{V}$. The unitary matrix W has the block form

$$W = \begin{pmatrix} W_1^* & W_2^* \\ W_2 & W_1 \end{pmatrix}, \tag{23}$$

where the submatrices satisfy

$$W_1 W_1^\dagger + W_2 W_2^\dagger = \mathbb{1}, \tag{24}$$

$$W_1 W_2^T + W_2 W_1^T = \mathbb{0}, \tag{25}$$

with $\mathbb{1}$ and $\mathbb{0}$ being the $N \times N$ identify matrix and the zero matrix, respectively. We define the $N \times 2N$ matrix $W_L = (W_2 \; W_1)$ as the lower half of W; the jth row of W_L corresponds to the expansion coefficients of the operator b_j. The matrix W_L uniquely determines the transformation $\mathcal{W}$ up to an overall phase.

Using elementary matrix manipulations on W_L, we demonstrate that the Gaussian unitary $\mathcal{U}$ in Eq. (22) can be broken into a sequence of operations that can be implemented on a quantum computer:

$$\mathcal{U} = \mathcal{B}\mathcal{G}_1\mathcal{B}\mathcal{G}_2\mathcal{G}_3\mathcal{B} \cdots \mathcal{G}_{N_G}\mathcal{B}, \tag{26}$$

where the $\mathcal{G}_j$ values are Givens rotations on adjacent fermionic modes encoded in the JWT, and $\mathcal{B}$ denotes the particle-hole Bogoliubov transformation on the last fermionic mode,

$$\mathcal{B} c_N \mathcal{B}^\dagger = c_N^\dagger, \tag{27}$$

$$\mathcal{B} c_j \mathcal{B}^\dagger = c_j, \quad \text{for } j = 1, 2, \ldots, N-1. \tag{28}$$

The transformation $\mathcal{B}$ does not conserve the total number of particles and is crucial to preparing superpositions of states with different numbers of particles. It can be implemented easily by applying the Pauli-X operator on the last qubit; the parities of the other fermionic modes encoded in the JWT are not affected, which would not be true for any other spin orbital. In the relevant coordinate axes, the matrix representation of the Givens rotation is

$$G = \left(\begin{array}{cc|cc} \cos\theta & -e^{i\varphi}\sin\theta & 0 & 0 \\ \sin\theta & e^{i\varphi}\cos\theta & 0 & 0 \\ \hline 0 & 0 & \cos\theta & -e^{-i\varphi}\sin\theta \\ 0 & 0 & \sin\theta & e^{-i\varphi}\cos\theta \end{array}\right). \tag{29}$$

The representation of the particle-hole transformation is

$$B = B^\dagger = \begin{pmatrix} \mathbb{1} - e_N e_N^T & e_N e_N^T \\ e_N e_N^T & \mathbb{1} - e_N e_N^T \end{pmatrix}, \tag{30}$$

where $e_N = (0 \cdots 0 \, 1)^T$ is an N-dimensional unit vector. The goal is to find a $2N \times 2N$ unitary matrix U decomposed into G and B,

$$U = B G_{N_G} \cdots B G_3 G_2 B G_1 B, \tag{31}$$

such that

$$V W_L U^\dagger = (\mathbb{0} \quad \mathbb{1}), \tag{32}$$

where V is an arbitrary unitary matrix. The right-hand side of Eq. (32) represents the original annihilation operators c_j, which corresponds to the vacuum state defined in Eq. (22). We discuss how to bring W_L to the desired form (32) using an example of four spin orbitals. Following Sec. III, we use the unitary V as a freedom to zero out some matrix elements on the left side of W_L,

$$V W_L = \left(\begin{array}{cccc|cccc} 0 & 0 & 0 & * & * & * & * & \underline{0} \\ 0 & 0 & * & * & * & * & * & * \\ 0 & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * \end{array}\right), \tag{33}$$

where $*$ represents an arbitrary matrix element and the underlined matrix element is automatically zeroed out due to Eq. (25). A sequence of elementary operations that brings W_L to the desired form is

$$
\begin{aligned}
VW_L \to & \left(\begin{array}{cccc|cccc} 0 & 0 & 0 & \mathbf{0} & * & * & * & * \\ 0 & 0 & * & * & * & * & * & * \\ 0 & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * \end{array}\right) \to \left(\begin{array}{cccc|cccc} 0 & 0 & 0 & 0 & * & * & * & \underline{0} \\ 0 & 0 & \mathbf{0} & * & * & * & * & \underline{0} \\ 0 & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * \end{array}\right) \to \left(\begin{array}{cccc|cccc} 0 & 0 & 0 & 0 & * & * & \underline{0} & 0 \\ 0 & 0 & 0 & \mathbf{0} & * & * & * & * \\ 0 & \mathbf{0} & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * \end{array}\right) \\
\to & \left(\begin{array}{cccc|cccc} 0 & 0 & 0 & 0 & \underline{\lambda_1} & \underline{0} & 0 & 0 \\ 0 & 0 & 0 & 0 & \underline{0} & * & * & \underline{0} \\ 0 & 0 & \mathbf{0} & * & \underline{0} & * & * & \underline{0} \\ \mathbf{0} & * & * & * & \underline{0} & * & * & * \end{array}\right) \to \left(\begin{array}{cccc|cccc} 0 & 0 & 0 & 0 & \lambda_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \underline{\lambda_2} & \underline{0} & 0 \\ 0 & 0 & 0 & \mathbf{0} & 0 & \underline{0} & * & * \\ 0 & \mathbf{0} & * & * & 0 & \underline{0} & * & * \end{array}\right) \to \left(\begin{array}{cccc|cccc} 0 & 0 & 0 & 0 & \lambda_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \underline{\lambda_3} & \underline{0} \\ 0 & 0 & \mathbf{0} & \underline{\lambda_4} & 0 & 0 & \underline{0} & \underline{0} \end{array}\right) \\
\to & \left(\begin{array}{cccc|cccc} 0 & 0 & 0 & 0 & \lambda_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \lambda_3 & 0 \\ 0 & 0 & 0 & \mathbf{0} & 0 & 0 & 0 & \boldsymbol{\lambda_4} \end{array}\right) \to VW_L U^\dagger. \qquad (34)
\end{aligned}
$$

The bold matrix elements in the fourth column are always zeroed out by the particle-hole transformation B, and the other bold matrix elements on the left side are zeroed out by the Givens rotations G. The bold elements on the right side become nonzero due to the particle-hole transformation B, and the underlined matrix elements are brought to zeros or phase factors automatically by the condition (24) or (25). The phase factors are brought to ones in the last step by single-qubit rotations; this step is unnecessary if $\mathcal{U}$ is applied to the vacuum state. The total numbers of Givens rotations and particle-hole transformations are

$$N_G = (N-1)N/2, \qquad N_B = N, \qquad (35)$$

and the circuit depth is, at most, $2N-1$.

In summary, we describe in this section a method to prepare an arbitrary Gaussian state as the ground state of the quadratic Hamiltonian (19) using two-qubit gates that act only on neighboring qubits. Our method can be achieved in four steps:

(1) Calculate the matrix W using the procedure described in Appendix A.

(2) Zero out the upper-left matrix elements of W_L using the freedom $W_L \to VW_L$ (33).

(3) Zero out the remaining matrix elements on the left side of VW_L using the sequence (34).

(4) Find the quantum gates corresponding to the sequence in step 3.

The procedure described here also applies to the implementation of an arbitrary Gaussian unitary, where one cannot use the unitary freedom V. By rearranging Eq. (32), we have

$$W_L = V^\dagger(\mathbb{0} \quad \mathbb{1})U = (\mathbb{0} \quad \mathbb{1})\begin{pmatrix} V^T & \mathbb{0} \\ \mathbb{0} & V^\dagger \end{pmatrix} U, \qquad (36)$$

where the matrix $\mathrm{diag}(V^T, V^\dagger)$ corresponds to a basis change and can be decomposed as Givens rotations using the method discussed in Sec. III.

As mentioned above, this algorithm has also been implemented as part of OpenFermion [38]. The user can specify a quadratic Hamiltonian by inputting the matrices M and Δ from Eq. (19). The code then outputs a sequence of operations, each of which either describes a Givens rotation or indicates a particle-hole transformation on the last fermionic mode. The operations that can be performed in parallel are grouped together. Since OpenFermion also contains modules to initialize common Hamiltonian models, it can be used to easily specify, for instance, a mean-field Hamiltonian and then obtain a circuit that prepares its ground state. We use the code to verify that the method described here does produce the desired ground state.

V. FERMIONIC FOURIER TRANSFORMATIONS

The fermionic Fourier transformation is a subroutine of many quantum algorithms. It was first introduced for quantum-computing purposes in Ref. [37]. Recently, Babbush *et al.* [9] demonstrated that the 2D and 3D fermionic Fourier transformations can be implemented using a 2D qubit array with $O(N)$ depth. Here, we present an algorithm to implement the 2D fermionic Fourier transformation using a 2D qubit array with only $O(\sqrt{N})$ depth. This amount of depth is required for quantum information to travel across the array, and therefore this scaling is optimal. Our method provides an example where the parity problem in JWTs for more than one spatial dimension can be circumvented with negligible overhead. Our algorithm also works for more general transformations that can be factorized into a product of transforms on each spatial dimension, including Fourier transformations with open boundary conditions and fermionic Gaussian unitaries. This algorithm allows one to efficiently prepare the initial states for systems whose ground states are well approximated by the mean-field states, as well as to improve the efficiency of measurements.

In Sec. III, we demonstrate that any fermionic single-particle basis transformation—including the 1D fermionic Fourier transformation—can be implemented using a 1D qubit chain with $O(N^2)$ gates and $O(N)$ depth; see also Theorem 7 in Ref. [9]. Using this algorithm as a subroutine, we discuss an algorithm that implements the 2D fermionic Fourier transformation using a 2D qubit array with $O(N^{1.5})$ gates and $O(\sqrt{N})$ depth. More generally, our method works for any fermionic single-particle basis transformation $\mathcal{F}$ (or fermionic Gaussian unitary) that factorizes,

$$\mathcal{F} = \mathcal{F}_x\mathcal{F}_y = \mathcal{F}_y\mathcal{F}_x, \tag{37}$$

where the horizontal (vertical) transformation $\mathcal{F}_x$ ($\mathcal{F}_y$) is a product of commuting terms, each of which involves spin orbitals only in the same row (column). We map the fermionic operators to qubit operators using a snake-shaped JWT in row-major order; see Fig. 2. The transformation $\mathcal{F}_x$ on a single row can be implemented with $O(n_x^2)$ gates and $O(n_x)$ depth using the algorithm described in Sec. III, where n_x is the number of spin orbitals in a single row (number of columns). The vertical transformation $\mathcal{F}_y$ is much harder to implement with our mapping because of the nonlocal parity operators in the hopping terms; see Eq. (5). This difficulty can be overcome by reordering the fermionic spin orbitals in the JWT using the fermionic SWAP gates; such a strategy allows one to perform the 2D fermionic Fourier transformation using a 2D qubit array with $O(N^2)$ gates and $O(N)$ depth [9].

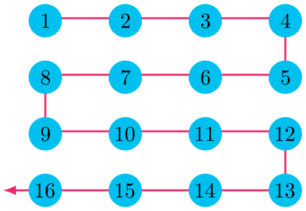

FIG. 2. The fermionic spin orbitals on a 4 × 4 array are mapped to qubits (the blue circles) on an array of the same dimensions using a snake-shaped JWT. The red arrow shows the direction of the JWT, and the qubits are numbered by their order in the JWT. Hopping terms between an odd-numbered row and the row below are called right-closed hopping terms, while those between an even-numbered row and the row below are called left-closed hopping terms.

We use a different approach that takes full advantage of the 2D qubit interactions. In our scheme, the transformation (37) is realized using the decomposition

$$\mathcal{F} = \mathcal{F}_x\mathcal{F}_y = \mathcal{F}_x\Gamma^\dagger\mathcal{F}_y^b\Gamma, \tag{38}$$

where $\Gamma = \Gamma^\dagger$ is a diagonal unitary matrix in the computational (Pauli-Z) basis with eigenvalues ±1. We will show that Γ can be implemented with $O(N)$ gates and $O(\sqrt{N})$ depth. The unitary $\mathcal{F}_y^b$ is implemented by using the Givens rotations without the parity operators attached; see Fig. 1. This procedure is equivalent to using a JWT with column-major order, and the generator of the real Givens rotation takes the form (6a)

$$K_{jk} = \frac{1}{2}(X_jY_k - Y_jX_k), \tag{39}$$

where the qubits j and k are adjacent in one column; we call these operators bare hopping terms.

The unitary transformation Γ attaches the corresponding parity operator to the bare hopping terms,

$$\Gamma^\dagger K_{jk}\Gamma = K_{jk}Z_{j+1}\cdots Z_{k-1}, \qquad k > j, \tag{40}$$

for any indices j and k adjacent in the same column. Any non-neighboring vertical hopping terms can be derived as (nested) commutators of the nearest-neighbor ones, and their parities are taken care of automatically. Because $\mathcal{F}_y$ can be decomposed into a product of Givens rotations generated by the vertical hopping terms, we have $\Gamma^\dagger\mathcal{F}_y^b\Gamma = F_y$. We denote an arbitrary state in the computational basis with $|\mathbf{s}\rangle$, where $\mathbf{s} = (s_1, s_2, \ldots, s_N)$ is a binary string. The matrix element of the hopping term K_{jk} with respect to the basis states $|\mathbf{s}\rangle$ and $|\mathbf{s}'\rangle$ takes the form

$$\langle\mathbf{s}'|\Gamma^\dagger K_{jk}\Gamma|\mathbf{s}\rangle = \gamma_{\mathbf{s}}\gamma_{\mathbf{s}'}\langle\mathbf{s}'|K_{jk}|\mathbf{s}\rangle, \tag{41}$$

where $\gamma_{\mathbf{s}}$ and $\gamma_{\mathbf{s}'}$ are the corresponding eigenvalues of Γ.

Comparing Eq. (40) to Eq. (41), we have

$$\gamma_{\mathbf{s}}\gamma_{\mathbf{s}'} = (-1)^{\sum_{l=j+1}^{k-1} s_l} = (-1)^{\sum_{l=j+1}^{k-1} s'_l}, \tag{42}$$

for any pair of basis states such that $\langle\mathbf{s}'|K_{jk}|\mathbf{s}\rangle \neq 0$. The matrix element $\langle\mathbf{s}'|K_{jk}|\mathbf{s}\rangle$ is nonzero only when $s_j \neq s'_j$, $s_k \neq s'_k$, and $s_l = s'_l$ for all $l \neq j$, k; the total parity of the qubits j and k also needs to be odd,

$$s_j + s_k = s'_j + s'_k = 1. \tag{43}$$

The unitary $\mathcal{F}_y^b$ on a single column can be implemented with $O(n_y^2)$ gates and $O(n_y)$ depth using the bare hopping terms, where n_y is the number of spin orbitals per column. By parallelizing operations on different rows or columns of qubits, one can implement either $\mathcal{F}_x$ or $\mathcal{F}_y^b$ with $O(N^{1.5})$ gates and $O(\sqrt{N})$ depth, where $N = n_x \times n_y$ is the number of spin orbitals.

A Slater determinant in the momentum basis can be prepared by applying the transformation $\mathcal{F}$ to a Slater determinant in the site basis $|\mathbf{s}\rangle$,

$$\mathcal{F}|\mathbf{s}\rangle = \mathcal{F}_x\Gamma^\dagger\mathcal{F}_y^b\Gamma|\mathbf{s}\rangle = \gamma_{\mathbf{s}}F_x\Gamma^\dagger\mathcal{F}_y^b|\mathbf{s}\rangle, \qquad (44)$$

where $\gamma_{\mathbf{s}} = \pm 1$. The transformation Γ is also useful for simulating the time evolution of fermionic systems, e.g., the Fermi-Hubbard model. The hopping terms in each Trotter step can be implemented with $O(N)$ gates and $O(\sqrt{N})$ circuit depth by using Γ.

To implement Γ, we introduce one ancilla qubit per row to store the parities (see Fig. 4); they are used only to facilitate the implementation of Γ and are disentangled with the system qubits in the end. Initially, the ancilla qubits are located on the right side in all of the system qubits, and their states are set to $|0\rangle$. In each time step, each ancilla qubit is swapped with the system qubit to the left, which allows it to interact with other system qubits. The parity stored in the ancilla qubit is updated by applying the controlled-NOT (CNOT) gate controlled by the same system qubit (now on the right side of the ancilla); see Fig. 3. After the CNOT gate, the ancilla qubits store the total parity of system qubits to their right on the same rows. We apply controlled-Z (CZ) gates between the ancilla qubits and the system qubits (see Fig. 4); each CZ gate introduces a parity (an overall ± 1 sign) to the state $|\mathbf{s}\rangle$ in the computational basis. The goal is to find the set of CZ gates, acting on neighboring qubits, such that they put the desired parities with the bare vertical hopping terms. It is instructive to first work out the cases for right-closed and left-closed hopping terms separately.

In Fig. 4, we plot the CZ gates (the dashed lines) that generate the desired parities for the right-closed hopping terms between an odd-numbered row and the row below it. We explain how this works for the hopping term between the system qubits 2 and 7. Any two basis states $|\mathbf{s}\rangle$ and $|\mathbf{s}'\rangle$ corresponding to a nonzero matrix element of the hopping term satisfy $s_l = s'_l$ for $l \in \{3, 4, 5, 6\}$; therefore, they accumulate the same parity before hitting the two CZ gates involving qubit 2; see Fig. 4(b). The states of the two ancilla qubits involved in these two CZ gates are the same for the two basis states because the parities of qubits 2 and 7 have not been added to them. After the two CZ gates in Fig. 4(b), the two basis states acquire a difference in parity equaling $(s_3 + s_4 + s_5 + s_6)$ mod 2; see Fig. 5. To simplify the notation, we will neglect mod 2 in dealing with parities hereafter. The two ancilla qubits in the first and second rows take different values for $|\mathbf{s}\rangle$ and $|\mathbf{s}'\rangle$ after the parities of qubits 2 and 7 are added to them; however, the total parities of the two ancilla qubits are still the same. The subsequent two CZ gates (qubit 1 is involved) do not introduce a parity difference to the two basis states because both s and $a + b$ are the same for the two basis states. After the ancilla qubits reach the left end, all of the right-closed hopping terms get the desired parities from the CZ gates.

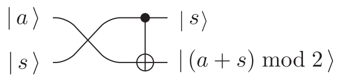

FIG. 3. The ancilla qubit a is swapped with the system qubit s to its left, and its state is then updated by the CNOT gate.

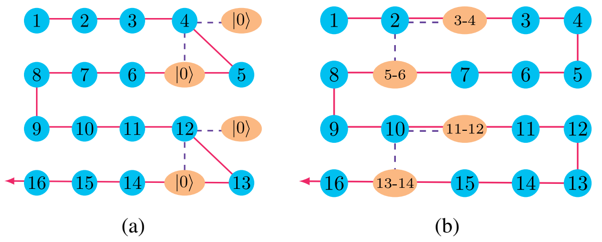

FIG. 4. The procedure to generate the desired parities for the right-closed hopping terms. The ancilla qubits (the orange ellipses) store the parity of the system qubits (the blue circles), where j-k denotes the parity $(\sum_{l=j}^{k} s_l)$ mod 2. (a) The ancilla qubits are initially located at the right side of the lattice, with their states set to $|0\rangle$. (b) They are then moved to the left while the parity stored in them being updated. The purple dashed lines represent CZ gates between the ancilla and the system qubits for right-closed hopping terms. The CZ gates in (a) can be omitted because the ancilla states are set to $|0\rangle$ initially.

In Fig. 6, we plot the CZ gates that introduce the desired parities for the left-closed hopping terms, i.e., those between an even-numbered row and the row below. We now explain how this works for the hopping term between qubits 6 and 11. Any two basis states $|\mathbf{s}\rangle$ and $|\mathbf{s}'\rangle$ corresponding to a nonzero matrix element of the hopping term satisfy $s_l = s'_l$ for $l \in \{5, 12\}$; therefore, they accumulate the same parities before hitting the two CZ gates in Fig. 6(a). These two CZ gates do not introduce a parity difference, either, because both the parity of the ancilla qubit and the total parity of the two system qubits are the same for the two basis states; see Eq. (43) and Fig. 7. The ancilla qubit takes different values for $|\mathbf{s}\rangle$ and $|\mathbf{s}'\rangle$ after the parity of qubit 11 is added to it. As a result, the following two CZ gates (qubits 7 and 10 are involved) introduce a parity difference to the two basis states equaling $s_7 + s_{10} = s'_7 + s'_{10}$; see Fig. 7. Since the state of the ancilla

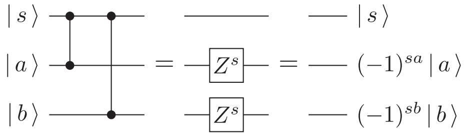

FIG. 5. The CZ gates involving one system qubit s and two ancilla qubits a and b. When $s \neq s'$ and $a + b = a' + b'$ for the two basis states, a parity difference of $a + b$ is introduced. By comparison, no parity difference is introduced when both quantities are the same for the two basis states.

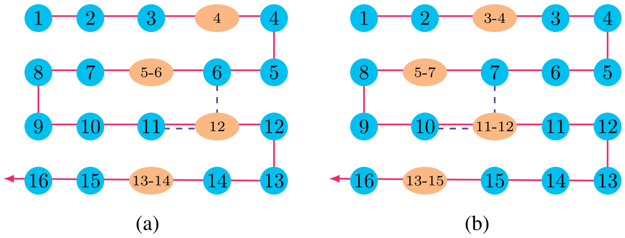

FIG. 6. The procedure to generate the desired parities for the left-closed hopping terms. The ancilla qubits (the orange ellipses) store the parity of the system qubits (the blue circles), where j-k denotes the parity $(\sum_{l=j}^{k} s_l) \bmod 2$. (a) Two CZ gates (the dashed lines) are applied to the anilla qubit labeled 12 and the system qubits labeled 6 and 11. (b) Two CZ gates are applied to the anilla qubit labeled 11-12 and the system qubits labeled 7 and 10.

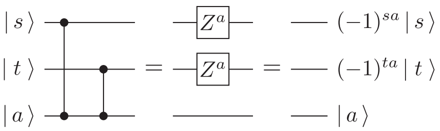

FIG. 7. The CZ gates involving two system qubits s and t and one ancilla qubit a. When $a \neq a'$ and $s + t = s' + t'$ for the two basis states, a parity difference of $s + t$ is introduced. By comparison, no parity difference is introduced when both quantities are the same.

qubit remains different for $|\mathbf{s}\rangle$ and $|\mathbf{s}'\rangle$, a parity difference of $s_8 + s_9 = s_8' + s_9'$ is introduced in the next step. After the ancilla qubits reach the left end, all of the left-closed hopping terms get the desired parities from the CZ gates.

We show how to introduce the desired parities to the right-closed and the left-closed hopping terms using the CZ gates in Figs. 4 and 6, respectively. The ancilla qubits are brought in to interact with system qubits in a particular column at a time, and they store the total parities of all of the system qubits in the same row to the right of the current column. We depict the two kinds of CZ gates in Fig. 8 for a single time step, where the system (ancilla) qubits are represented by blue (red) dots. One will not achieve the desired results for both kinds of hopping terms by simply combining the two sets of CZ gates. This is the case because the CZ gates for the right-closed terms also affect the parities of the left-closed terms, and vice versa.

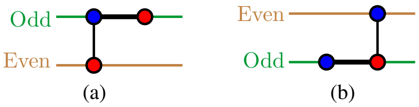

FIG. 8. The CZ gates between system (blue) and ancilla (red) qubits. (a) Right-closed hopping terms (between an odd-numbered row and the row below it), e.g., the one involving qubit 2 in Fig. 4(b). (b) Left-closed hopping terms (between an even-numbered row and the row below it), e.g., the one involving qubits 6 and 11 in Fig. 6(a).

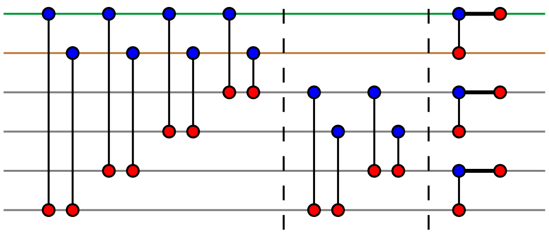

FIG. 9. The CZ gates between the system (blue) and ancilla (red) qubits in a single time step that introduce the desired parities to both the right-closed and left-closed hopping terms.

To circumvent this difficulty, we introduce the CZ gates for a single time step using an example of 6 rows in Fig. 9. In addition to the CZ gates in Fig. 8 for all of the right-closed hopping terms, we also introduce a CZ gate between each system qubit and the ancilla qubits below it, starting with the next odd-numbered row. We show that this set of CZ gates works for both right-closed and left-closed hopping terms. First, we discuss the hopping term between the first and second rows. The CZ gates are divided into three parts in Fig. 9. In the first part, the states of the ancilla qubits are the same for the two basis states $|\mathbf{s}\rangle$ and $|\mathbf{s}'\rangle$, and the total parity of the two system qubits in the first and second rows are also the same; see Eq. (43). As a result, the CZ gates in the first part do not introduce a parity difference to the two basis states. Neither do the CZ gates in the second part because they act on qubits that are not involved in the hopping term. The third part is equivalent to the circuit in Fig. 8(a). Therefore, this circuit works for any hopping term between the first and second rows.

To show how the same circuit works for the hopping terms between the second and third rows, we rearrange the CZ gates in Fig. 9 into the specific form in Fig. 10. Because all CZ gates commute, these two circuits are equivalent. The first three pairs of CZ gates and the second part in Fig. 10 do not introduce a parity difference to the two basis states, for the same reasons we discussed earlier. Neither do the last two CZ gates in the first part because the total parity of the two ancilla qubits in the second and third rows are the same for the two basis states. The third part of Fig. 10 is equivalent to the circuit in Fig. 8(b). Therefore, the same

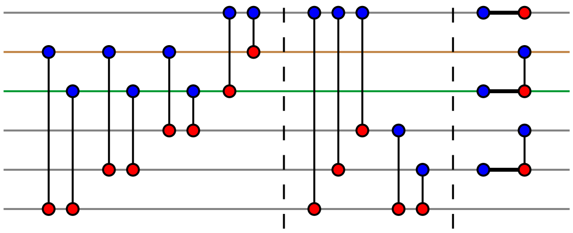

FIG. 10. Quantum circuit to demonstrate the hopping terms between the second and third rows by rearranging the CZ gates in Fig. 9.

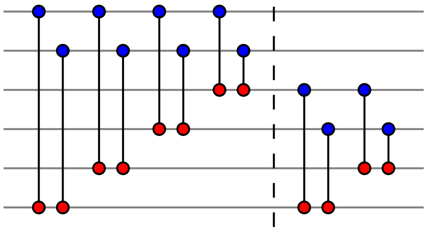

FIG. 11. The first and second parts in Fig. 9.

circuit also works for any hopping term between the second and third rows. It is straightforward to verify that the circuit works for any other nearest-neighbor vertical hopping terms in this example; see Appendix G. Therein, we also give an argument on why our approach works for systems with any even number of rows (an unused row can be added when the number of rows is odd).

The problem with implementing the circuit in Fig. 9 is that there are many nonlocal gates in the first and second parts, which we replot in Fig. 11. To deal with this difficulty, we go to the parity basis of columns by applying the circuit in Fig. 12(a) to both the system and the ancilla qubits. In this new basis, any system qubit stores the total parity of the original system qubits in the same column from the current location to the bottom; e.g., the qubit 10 in Fig. 12(b) stores the parity $s_{10} + s_{15}$ in the parity basis of the columns. Any ancilla qubit stores the total parity to the lower left of the corresponding system qubit; e.g., the ancilla qubit corresponding to system qubit 10 in Fig. 12(b) stores the parity $s_{11} + s_{12} + s_{13} + s_{14}$. To find the circuit corresponding to Fig. 11 in the new basis, we first go to the parity basis of the ancilla qubits and keep the system qubits unchanged.

With this intermediate basis, the first four (last two) pairs of CZ gates in Fig. 11 are mapped to the first (second) pair of CZ gates in Fig. 13(a). This is the case because we can combine all CZ gates acting on the same system qubit into a single CZ gate by using an ancilla qubit storing the total parity of the original ancilla qubits. We continue to go to the parity basis of the system qubits, and the circuit in Fig. 13(a) is mapped to the one in Fig. 13(b). This is the case because the total parity of the first and second (third and fourth) qubits in Fig. 13(a) is equal to the total parity of the first and third (third and fifth) qubits.

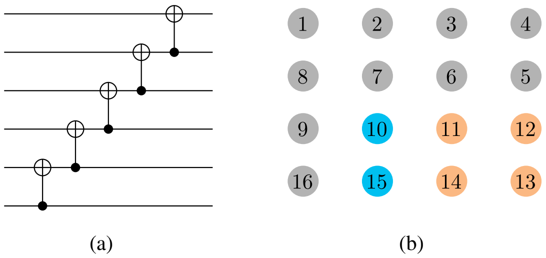

FIG. 12. Parity basis of the columns. (a) The circuit maps the computational basis to the parity basis. (b) The parities stored in system qubit 10 (blue) and the corresponding ancilla (orange).

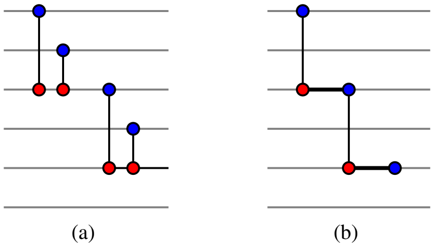

FIG. 13. Equivalent circuits to Fig. 11. (a) The ancilla qubits are in the parity basis and the system qubits are in the original basis. (b) Both the system and the ancilla qubits are in the parity basis.

In general, we have a CZ gate between the system qubit on each odd-numbered row and the ancilla qubit two rows below it and a CZ gate between the system and the ancilla qubits on each odd-numbered row except for the first row. Going to the parity basis of the system qubits might seem to be unnecessary, but we will show that it is essential for reducing the circuit depth.

We have shown that the first two parts in Fig. 9 can be implemented with only local CZ gates; however, going to the parity basis requires a sequence of $O(\sqrt{N})$ gates which increases the circuit depth. One solution is to implement, in parallel, the quantum circuit in Fig. 12(a) to all columns of system qubits at the beginning. As the ancilla qubits move to the left, they pick up the parities stored in the system qubits. These add up to the desired parities of the ancilla qubits in the parity basis, and no basis transformation is needed when the ancilla qubits move across the system qubits. The circuit in Fig. 13(b) is implemented in each time step. After the ancilla qubits reach the left end, we go back to the original basis by applying the gates in Fig. 12(a), in reverse order, to both the system and the ancilla qubits. We then move the ancilla qubits to the right by reversing the order of the CNOT and the SWAP gates when we move them to the left. The CZ gates in the last part of Fig. 9 are implemented in each time step. All of the ancilla qubits are disentangled with the system qubits when they reach the right end, and the unitary Γ has been implemented on the system qubits. We verify our procedure numerically by using random classical bit strings $\mathbf{s}$ and $\mathbf{s}'$ for system sizes up to 100 × 100 and by simulating the actual quantum circuit for the system size 4 × 4.

In summary, the unitary Γ can be implemented with the following four stages:

(1) Implement the circuit in Fig. 12(a) for each column of the system qubits.
(2) Move the ancilla qubits to the left while applying the CZ gates in Fig. 13(b).
(3) Undo the circuit in Fig. 12(a) for both the system qubits and the ancilla qubits.
(4) Move the ancilla qubits to the right while applying the CZ gates in the third part of Fig. 9.

Each of the four stages can be implemented with $O(N)$ gates and $O(\sqrt{N})$ depth; therefore, the whole procedure takes gates and depth with the same scalings. By comparison, $\mathcal{F}_x$ or $\mathcal{F}_y^b$ requires $O(N^{1.5})$ gates and $O(\sqrt{N})$ depth to implement. Our quantum algorithm provides an example in which the parity problem in simulating 2D fermionic systems can be circumvented with negligible overhead.

VI. THE FERMI-HUBBARD MODEL

One way to unravel the intricate physics in strongly correlated materials is to approximate them with idealized model Hamiltonians, such as the Fermi-Hubbard model. The Hubbard model captures many signatures of the physical systems, although it is too simple to describe real materials quantitatively. It has resisted a full solution despite decades of intense analytical and numerical studies. The Hubbard model has a relatively small number of interacting terms, allowing for easy implementation on a quantum computer. The single-band Hubbard model is described by the Hamiltonian

$$\mathcal{H}_{\mathrm{FH}} = -\sum_{\langle j,k\rangle,\sigma} t_{jk}(c_{j,\sigma}^\dagger c_{k,\sigma} + \mathrm{H.c.}) + U\sum_j n_{j,\uparrow}n_{j,\downarrow} + \sum_{j,\sigma}(\epsilon_j - \mu)n_{j,\sigma} - \sum_j h_j(n_{j,\uparrow} - n_{j,\downarrow}), \tag{45}$$

where the first term represents fermions hopping between sites, the second term represents on-site interactions, the third term is a local potential field, and the last term represents a local magnetic field.

The 2D Hubbard model is widely used as a canonical microscopic model for strongly correlated fermionic systems. It is believed to be a key ingredient in understanding the mechanism behind high-temperature superconductivity [14]. In particular, understanding the physics of the model in the intermediate interaction strength regime $U/t \sim 4$ in the underdoped region remains an open problem. Multiple orders exist in this region of its phase diagram; see Fig. 14. Even the nature of the ground state remains ambiguous due to the competition between a number of different order parameters [51]. Significant progress has been made in the identification of the unconventional Mott-insulator transition and a superconducting phase with d-wave order parameter [52,53]. Recently, the antiferromagnet phase of the Fermi-Hubbard model was realized in optical lattices of about 80 sites at a temperature of $1/4$ times the tunneling energy [29]. This region of multiple competing phases is also the most interesting regime from the point of view of modeling materials; therefore, a simulation using even a relatively small quantum computer could provide alternative qualitative and quantitative insights into the physics of the model [11,54,55]. Generalizations of the Hubbard model beyond the single-band case and including more complicated lattices could be a route to modeling a wide range of strongly correlated materials. It is also a versatile tool for exploring strongly correlated electron phases [56] in a controlled way. Simulation of the Hubbard model on a quantum computer could allow quantitative analysis of the physical characteristics beyond the phase diagram, such as dynamical effects which could help uncover the deeper physics of the strongly correlated phases in the model.

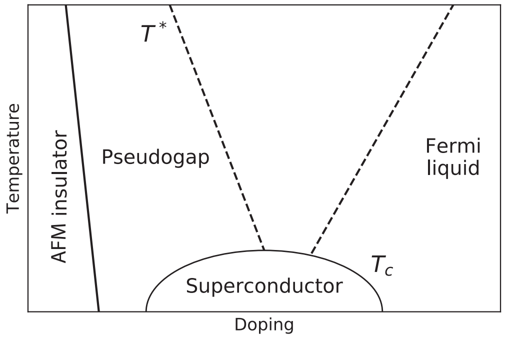

FIG. 14. Phase diagram of the 2D Fermi-Hubbard model. The critical temperature T_c depends on the value of hole doping, AFM stands for antiferromagnet.

In order to implement time evolution of the Hubbard model on a quantum computer, we decompose it into a product of available quantum gates based on the Trotter-Suzuki formula [57,58]; the desired accuracy determines the number of time steps [59,60]. We simulate the Trotter error numerically for small system sizes in Appendix F. In each time step, we implement successively the horizontal hopping terms, the vertical hopping terms, and the remaining terms. The two spin states can be mapped to two sublattices of a 2D qubit array; for an example, see Fig. 15. The horizontal hopping terms can be implemented by bringing qubits corresponding to the same spin states next to each other with the fermionic SWAP gates; for example, one can swap the qubits $j\uparrow$ with $j\downarrow$ for an odd j value in a row to implement some hopping terms, swap back, and then swap $j\uparrow$ with $j\downarrow$ for an even j value in the same row to implement the remaining hopping terms. This step requires $O(N)$ gates and $O(1)$ depth. The vertical hopping terms can be implemented using the method described in Sec. V; this requires $O(N)$ gates and $O(\sqrt{N})$ depth. Alternatively, the vertical hopping terms can be implemented using the method described in Appendix B, also with $O(N)$ gates and $O(\sqrt{N})$ depth. The on-site interaction terms in Eq. (45) are mapped to qubit operators of the form

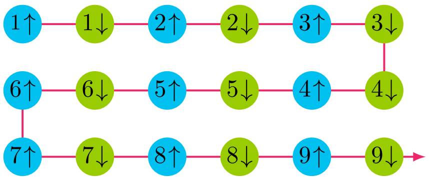

FIG. 15. One way to map the two spin states to qubits on a 2D qubit array using the JWT.

$$n_{j\uparrow}n_{j\downarrow} \mapsto \frac{1}{4}(I - Z_{j\uparrow})(I - Z_{j\downarrow}), \tag{46}$$

which can be implemented using a controlled-phase gate,

$$\exp(-i\tau U n_{j\uparrow}n_{j\downarrow}) \mapsto \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{-i\tau U} \end{pmatrix}. \tag{47}$$

The bias term and the magnetic term in the second line of Eq. (45) can be implemented straightforwardly with single-qubit operators. This step requires $O(N)$ gates and $O(1)$ depth. Putting all of these steps together, each Trotter step can be simulated with only $O(N)$ quantum gates and $O(\sqrt{N})$ depth.

Cooper proved that an arbitrarily small attraction between electrons can cause a pairing of electrons, leading to a lower energy state than the Fermi energy. As described by the BCS theory [49], the s-wave pairing wave function due to electron-phonon interactions is responsible for conventional superconductivity. The cuprate superconductivity has d-wave symmetry [61,62], wherein the superconducting wave function changes sign upon rotation by 90°. It has been predicted that d-wave pairing underlies high-T_c superconductivity in cuprates [63]; however, the mechanism of this pairing is not completely known. The mean-field Hamiltonian that describes d-wave pairing in the single-band Hubbard model is

$$\mathcal{H}_{\rm DW} = -\sum_{\langle j,k\rangle,\sigma} t_{jk}\left(c^\dagger_{j,\sigma}c_{k,\sigma} + c^\dagger_{k,\sigma}c_{j,\sigma}\right) - \mu\sum_{j,\sigma} n_{j,\sigma} \\ -\sum_{\langle j,k\rangle}\Delta^{x^2-y^2}_{jk}\left(c^\dagger_{j\uparrow}c^\dagger_{k\downarrow} - c^\dagger_{j\downarrow}c^\dagger_{k\uparrow} + \mathrm{H.c.}\right), \tag{48}$$

where the chemical potential term with μ regulating the total number of particles and $\Delta^{x^2-y^2}_{jk} = \pm\Delta/2$ are the superconducting gaps for the horizontal and vertical directions, respectively. With the mapping in Fig. 15, the pairing term $c^\dagger_{j\uparrow}c^\dagger_{k\downarrow} + \mathrm{H.c.}$ can be implemented similarly to the hopping terms.

Assuming translational symmetry and periodic boundary conditions, the ground state of the Hamiltonian (48) can be prepared using the fermionic Fourier transformation,

$$c^\dagger_{\mathbf{k},\sigma} = \frac{1}{\sqrt{n_x n_y}}\sum_{j=1}^{n_x n_y} e^{2\pi i(k_x x_j + k_y y_j)}c^\dagger_{j,\sigma}, \tag{49}$$

where n_x (n_y) is the number of sites in the horizontal (vertical) direction, and $x_j \in \{1,\ldots,n_x\}$ and $y_j \in \{1,\ldots,n_y\}$ denote the coordinates of the jth site. The discrete wave vector $\mathbf{k} = (k_x, k_y)$ satisfies the condition $k_x n_x \in \{1,\ldots,n_x\}$, and similarly for k_y.

In the momentum basis, the Hamiltonian (48) becomes

$$\mathcal{H}_{\rm DW} = \sum_{\mathbf{k},\sigma}\xi_{\mathbf{k}}c^\dagger_{\mathbf{k},\sigma}c_{\mathbf{k},\sigma} - \sum_{\mathbf{k}}\Delta_{\mathbf{k}}\left(c^\dagger_{\mathbf{k}\uparrow}c^\dagger_{-\mathbf{k}\downarrow} + \mathrm{H.c.}\right), \tag{50}$$

where $-\mathbf{k} \equiv (1-k_x, 1-k_y)$ and

$$\xi_{\mathbf{k}} = -2t[\cos(2\pi k_x) + \cos(2\pi k_y)] - \mu, \tag{51}$$

$$\Delta_{\mathbf{k}} = \Delta[\cos(2\pi k_x) - \cos(2\pi k_y)], \tag{52}$$

where μ is chosen such that $\xi_{\mathbf{k}} = 0$ at the Fermi surface. The BCS mean-field ground state of Eq. (50) is

$$|\Psi_{\rm DW}\rangle = \prod_{\mathbf{k}}\left(u_{\mathbf{k}} + v_{\mathbf{k}}c^\dagger_{\mathbf{k}\uparrow}c^\dagger_{-\mathbf{k}\downarrow}\right)|\mathrm{vac}\rangle, \tag{53}$$

$$u^2_{\mathbf{k}} = \frac{1}{2}\left(1 + \frac{\xi_{\mathbf{k}}}{\sqrt{\xi^2_{\mathbf{k}} + |\Delta_{\mathbf{k}}|^2}}\right), \tag{54}$$

$$v^2_{\mathbf{k}} = \frac{1}{2}\left(1 - \frac{\xi_{\mathbf{k}}}{\sqrt{\xi^2_{\mathbf{k}} + |\Delta_{\mathbf{k}}|^2}}\right), \tag{55}$$

where $u_{\mathbf{k}} \geq 0$ and $\mathrm{sgn}v_{\mathbf{k}} = \mathrm{sgn}\Delta_{\mathbf{k}}$. Inspired by Ref. [37], we prepare the BCS ground state in the site basis by first preparing the ground state in the momentum basis (53) and then applying the 2D Fourier transformation to the two spin states independently.

The ground state in the momentum basis can be prepared by applying a Bogoliubov transformation to the vacuum state:

$$|\Psi_{\rm DW}\rangle = \prod_{\mathbf{k}}\exp\left(\theta_{\mathbf{k}}c^\dagger_{\mathbf{k}\uparrow}c^\dagger_{-\mathbf{k}\downarrow} - \mathrm{H.c.}\right)|\mathrm{vac}\rangle, \tag{56}$$

where $\sin\theta_{\mathbf{k}} = v_{\mathbf{k}}$. The corresponding generator is

$$i\left(c^\dagger_{\mathbf{k}\uparrow}c^\dagger_{-\mathbf{k}\downarrow} - \mathrm{H.c.}\right) \mapsto \frac{1}{2}\left(X_{\mathbf{k}\uparrow}Y_{-\mathbf{k}\downarrow} + Y_{\mathbf{k}\uparrow}X_{-\mathbf{k}\downarrow}\right), \tag{57}$$

where the qubit corresponding to the spin orbital $-\mathbf{k}\downarrow$ is put next to that of $\mathbf{k}\uparrow$ in the JWT. This unitary can be implemented with the quantum circuit in Fig. 16.

The fermionic Fourier transformations, different for the two spin states due to the opposite orders in $\mathbf{k}$, can be

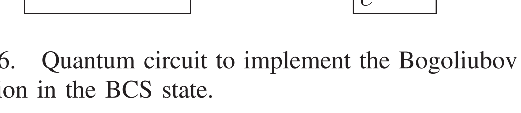

FIG. 16. Quantum circuit to implement the Bogoliubov transformation in the BCS state.

performed using the method described in Sec. V. The BCS mean-field state with periodic boundary conditions can thus be prepared using $O(N^{1.5})$ gates and $O(\sqrt{N})$ circuit depth.

An alternative way to prepare the d-wave mean-field state is by introducing the fermionic operators corresponding to the real single-particle wave functions,

$$c^\dagger_{\mathbf{k}+,\sigma} = \frac{1}{\sqrt{2}}\left(c^\dagger_{\mathbf{k},\sigma} + c^\dagger_{-\mathbf{k},\sigma}\right), \tag{58}$$

$$c^\dagger_{\mathbf{k}-,\sigma} = \frac{-i}{\sqrt{2}}\left(c^\dagger_{\mathbf{k},\sigma} - c^\dagger_{-\mathbf{k},\sigma}\right), \tag{59}$$

for $k_1,\ k_2 \leq 1/2$. With these operators, the Hamiltonian (50) takes the form

$$\begin{aligned}\mathcal{H}_{\mathrm{DW}} = &\sum_{k_1,k_2\leq 1/2,\sigma} \xi_{\mathbf{k}}\left(c^\dagger_{\mathbf{k}+,\sigma}c_{\mathbf{k}+,\sigma} + c^\dagger_{\mathbf{k}-,\sigma}c_{\mathbf{k}-,\sigma}\right)\\ &- \sum_{k_1,k_2\leq 1/2} \Delta_{\mathbf{k}}\left(c^\dagger_{\mathbf{k}+\uparrow}c^\dagger_{\mathbf{k}+\downarrow} + c^\dagger_{\mathbf{k}-\uparrow}c^\dagger_{\mathbf{k}-\downarrow} + \mathrm{H.c.}\right),\end{aligned} \tag{60}$$

where the pairing terms are also "diagonalized" as a consequence of the real transformation matrix. Therefore, one can prepare the mean-field ground state by first preparing the Bogoliubov ground state of the spin orbitals $\mathbf{k}\pm\uparrow$ and $\mathbf{k}\pm\downarrow$ before performing the same real basis transformation for the two spin states. This method might be more efficient than the one using plane waves by avoiding the phase rotations.

In experiments, it is often the case that open boundary conditions are used. In this case, the horizontal or vertical hopping terms in Eq. (48) correspond to a real triangular matrix with real eigenstates. We use these eigenstates as the single-particle basis states instead of the plane waves; therefore, the basis transformation matrix is also real. Similar to the case of periodic boundary conditions, the basis states on a 2D array with open boundary conditions can also be decomposed into a product of 1D basis states in each direction. This factorized form allows for efficient implementation of the 2D basis transformation with $O(N^{1.5})$ gates and $O(\sqrt{N})$ depth using the method described in Sec. V. When the Hamiltonian (48) does not satisfy translational symmetry, we can still prepare the mean-field ground state using the method described in Sec. IV. However, we might not be able to take advantage of the factorized form of the transformation, and it takes $O(N^2)$ gates and $O(N)$ depth to prepare the mean-field ground state in the worst case.

The ground state of the 2D Fermi-Hubbard Hamiltonian (1) can be prepared by first preparing the mean-field ground state and then slowly interpolating from H_{DW} to H_{FH} [18]. Following Ref. [11], we introduce the Hamiltonian

$$\begin{aligned}\mathcal{H}(s) = &(1-s)\mathcal{H}_{\mathrm{DW}} + s\mathcal{H}_{\mathrm{FH}}\\ &-\zeta\sum_{\langle j,k\rangle}\Delta^{x^2-y^2}_{jk}\left(c^\dagger_{j\uparrow}c^\dagger_{k\downarrow} - c^\dagger_{j\downarrow}c^\dagger_{k\uparrow}\right) + \mathrm{H.c.}\\ &-\eta\sum_{\langle\langle j,k\rangle\rangle} i\Delta^{xy}_{jk}\left(c^\dagger_{j\uparrow}c^\dagger_{k\downarrow} - c^\dagger_{j\downarrow}c^\dagger_{k\uparrow}\right) + \mathrm{H.c.},\end{aligned} \tag{61}$$

where s slowly changes from 0 to 1 in the adiabatic algorithm and $\Delta^{xy}_{jk} = \Delta/2$ for $x_j - x_k = \pm 1$ and $y_j - y_k = \pm 1$. The coefficients ζ and η introduce small gaps to avoid quantum fluctuations of the d-wave order parameter and the otherwise gapless nodal quasiparticles, respectively.

The initial Hamiltonian H_{DW} favors a mean-field ground state with d-wave symmetry of the pairing order parameter. The algorithm proceeds in the following way. We initialize the system in a specific mean-field wave function of the d-wave type,

$$\mathcal{H}_{\mathrm{DW}}|\Psi_{\mathrm{DW}}(s=0)\rangle = E_0(s=0)|\Psi_{\mathrm{DW}}(s=0)\rangle, \tag{62}$$

and adiabatically deform it to the ground state of the Hubbard model. If the initial wave function does not reflect the symmetry of the Fermi-Hubbard ground state, the adiabatic trajectory encounters a quantum phase transition. This transition manifests as a small spectral gap between the ground state and the excited states at the point of transition, which vanishes in the thermodynamic limit. If the initial mean-field state reflects the symmetry of the ground state of the Hubbard model, there is no phase transition and the spectral gap remains independent of the system size in the course of the evolution. One indication that the phase transition has occurred in the course of the evolution is that the final state contains excitations above the pairing gap, which can be detected by measuring the correlation functions. The controlled-phase gate (47) is implemented off resonantly to achieve the required high fidelity in the Xmon superconducting qubits [28]. The optimal length of the gate translates into the limited strength of the effective interaction corresponding to roughly $U \sim 10$ MHz for the Xmons. For the desired parameter regime $U \sim 4t$, an upper bound on the superconducting gap can be inferred from typical values of the order parameters obtained numerically, $\Delta \lesssim 0.04t \sim 0.01U \sim 0.1$ MHz.

VII. SUMMARY

In this paper, we discuss quantum simulation of strongly correlated fermionic systems using qubit arrays. We improve on an existing quantum algorithm to prepare an arbitrary Slater determinant [11,45] by exploiting a unitary symmetry. We also present a quantum algorithm to prepare an arbitrary fermionic Gaussian state with $O(N^2)$ gates and $O(N)$ circuit depth. This algorithm—unlike existing ones that rely on translational symmetry—is completely general and is useful for simulating disordered systems and quantum impurity models. Our quantum algorithms are optimal in the sense that the number of parameters in the quantum circuit is equal to that describing the quantum states. We implement these algorithms as a part of the open-source project OpenFermion [38].

We also present an algorithm to implement the 2D fermionic Fourier transformation on a 2D qubit array with $O(N^{1.5})$ gates and $O(\sqrt{N})$ circuit depth, both of which scale better than methods based on fermionic SWAP gates [9]. A crucial step to achieve this optimal scaling is a unitary transformation that attaches the parity operators to the hopping terms; we show that it can be implemented with $O(N)$ gates and $O(\sqrt{N})$ circuit depth. This approach provides an example where the parity problem in simulating fermionic systems in more than one spatial dimension can be circumvented with almost no additional cost. Our algorithm can also be used for any 2D transformation that factorizes into horizontal and vertical terms, such as the fermionic Fourier transformation with open boundary conditions.

Using the Fermi-Hubbard model as an example, we discuss how to use our algorithms to find the ground-state properties and phase diagrams of strongly correlated systems. We demonstrate that the d-wave pairing mean-field states of the model can be prepared with $O(N^{1.5})$ gates and $O(\sqrt{N})$ depth. We show that each Trotter step in the time evolution can be implemented with $O(N)$ gates and $O(\sqrt{N})$ depth with only local qubit interactions. We discuss how to prepare the ground state of the model by adiabatically evolving the system from the mean-field Hamiltonian to the Hubbard Hamiltonian. The methods that we develop can also be used for other quantum lattice models that suffer from the negative-sign problem, e.g., frustrated spin systems, the $t-J$ model, or lattice gauge theories.

In conclusion, we show in this paper that physical properties of strongly correlated fermionic systems can be simulated on available 2D qubit arrays with only $O(N^{1.5})$ gates and $O(\sqrt{N})$ circuit depth with very little overhead by using the Jordan-Wigner transformation. This result is one more step towards the goal of using quantum computers to investigate correlated quantum systems that are beyond the reach of any classical computer.

ACKNOWLEDGMENTS

The authors would like to acknowledge the enlightening and useful discussions with Ryan Babbush, Garnet Kin-Lic Chan, Chunjing Jia, Jarrod McClean, Andre Petukhov, Yaoyun Shi, Norman Tubman, and Fang Zhang. This work is supported by the NASA Advanced Exploration Systems program and the NASA Ames Research Center. The research is based in part upon work supported by the Office of the Director of National Intelligence (ODNI). K. J. S. acknowledges support from NSF Grant No. 1717523. K. K. acknowledges support by NASA Academic Mission Services, Contract No. NNA16BD14C. All of the quantum circuits in this work are plotted using the ⟨q|pic⟩ system developed by Thomas Draper and Samuel Kutin. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

APPENDIX A: QUADRATIC HAMILTONIANS

Hamiltonians that are quadratic in fermionic creation and annihilation operators are important in the mean-field descriptions of many-body quantum systems. The most general form of a quadratic Hamiltonian is

$$\mathcal{H} = \sum_{j,k=1}^{N} \Big(M_{jk} - \mu\delta_{jk} \Big) c_j^\dagger c_k + \frac{1}{2}\sum_{j,k=1}^{N} \Big(\Delta_{jk} c_j^\dagger c_k^\dagger + \text{H.c.} \Big), \tag{A1}$$

where $M = M^\dagger$ and $\Delta = -\Delta^T$ are complex matrices and the chemical potential μ regulates the total number of particles; hereafter, μ will be absorbed into M to simplify notation. We review the standard results on how to bring $\mathcal{H}$ into the diagonal form,

$$\mathcal{H} = \sum_{j=1}^{N} \varepsilon_j b_j^\dagger b_j + c \text{ number}, \tag{A2}$$

where b_j and $b_j^\dagger$ are a new set of fermionic operators that satisfy the canonical anticommutation relations.

When the number of particles is conserved ($\Delta = 0$), the Hamiltonian (A1) takes the form

$$\mathcal{H} = \sum_{j,k=1}^{N} M_{jk} c_j^\dagger c_k. \tag{A3}$$

The commutator of $\mathcal{H}$ and a single creation operator is

$$\begin{aligned} [\mathcal{H}, c_l^\dagger] &= \sum_{jk} M_{jk} [c_j^\dagger c_k, c_l^\dagger] \\ &= \sum_{jk} M_{jk} c_j^\dagger \delta_{kl} = \sum_j M_{jl} c_j^\dagger. \end{aligned} \tag{A4}$$

In matrix form, we have

$$[\mathcal{H}, \mathbf{c}^\dagger] = M^T \mathbf{c}^\dagger = M^* \mathbf{c}^\dagger, \tag{A5}$$

where $\mathbf{c}^\dagger = (c_1^\dagger \cdots c_N^\dagger)^T$. The time evolution under the Hamiltonian $\mathcal{H}$ takes the form

$$e^{-i\tau\mathcal{H}}\mathbf{c}^\dagger e^{i\tau\mathcal{H}} = e^{-i\tau M^T}\mathbf{c}^\dagger. \tag{A6}$$

Therefore, the time evolution of the $2^N \times 2^N$ matrix $\mathcal{H}$ can be represented by the $N \times N$ unitary matrix $e^{-i\tau M^T}$. The quadratic form (A3) can be diagonalized into the form (A2) by introducing the fermionic operators b_j and $b_j^\dagger$ such that

$$\mathbf{b}^\dagger = U\mathbf{c}^\dagger, \qquad UM^TU^\dagger = \mathrm{diag}(\varepsilon_1, \ldots, \varepsilon_N), \tag{A7}$$

where U is an $N \times N$ unitary matrix and the eigenvalues satisfy $\varepsilon_1 \leq \varepsilon_2 \cdots \leq \varepsilon_N$. In the N_f-particle sector, the ground state of the Hamiltonian (A3) corresponds to a Slater determinant with the Q matrix being the first N_f rows of U; see Eq. (7).

When the number of particles is not conserved, we rewrite the Hamiltonian (A1) in matrix form,

$$\mathcal{H} = \frac{1}{2}\begin{pmatrix} \mathbf{c}^\dagger & \mathbf{c} \end{pmatrix}\begin{pmatrix} \Delta & M \\ -M^* & -\Delta^* \end{pmatrix}\begin{pmatrix} \mathbf{c}^\dagger \\ \mathbf{c} \end{pmatrix} + c \text{ number}, \tag{A8}$$

where the extra constant comes from different ordering of the fermionic operators; the matrix form (A8) is symmetrically ordered, while the form in Eq. (A1) is normally ordered. The standard way to diagonalize the Hamiltonian (A8) is by solving the Bogoliubov–de Gennes equations, which is somewhat cumbersome for numerical implementations. Alternatively, one introduces the Majorana fermion operators,

$$f_j = \frac{1}{\sqrt{2}}(c_j^\dagger + c_j), \qquad f_{j+N} = \frac{i}{\sqrt{2}}(c_j^\dagger - c_j), \tag{A9}$$

for $j = 1, 2, \ldots, N$, which satisfy the anticommutation relations

$$\{f_j, f_k\} = \delta_{jk}, \quad \text{for } j, k = 1, 2, \ldots, 2N. \tag{A10}$$

In matrix form, the transformation (A9) reads

$$\mathbf{f} = \Omega\begin{pmatrix} \mathbf{c}^\dagger \\ \mathbf{c} \end{pmatrix}, \qquad \Omega = \frac{1}{\sqrt{2}}\begin{pmatrix} \mathbb{1} & \mathbb{1} \\ i\mathbb{1} & -i\mathbb{1} \end{pmatrix}, \tag{A11}$$

where $\mathbf{f} = (f_1\ f_2 \cdots f_{2N})^T$. The Hamiltonian (A8) can be rewritten in terms of the Majorana fermion operators:

$$\mathcal{H} = \frac{i}{2}\mathbf{f}^TA\mathbf{f} + c \text{ number}, \tag{A12}$$

where A is a $2N \times 2N$ real antisymmetric matrix,

$$A = -i\Omega^*\begin{pmatrix} \Delta & M \\ -M^* & -\Delta^* \end{pmatrix}\Omega^\dagger. \tag{A13}$$

Conversely, any real antisymmetric matrix A corresponds to a quadratic Hamiltonian of the form (A1). The commutator between a single Majorana fermion and the Hamiltonian $\mathcal{H}$ is

$$[\mathcal{H}, f_l] = \frac{i}{2}\sum_{j\neq k}A_{jk}[f_jf_k, f_l] = i\sum_{j\neq l}A_{jl}f_j. \tag{A14}$$

Using the matrix form $[\mathcal{H}, \mathbf{f}] = -iA\mathbf{f}$, we have

$$e^{-i\tau H}\mathbf{f}e^{i\tau H} = e^{-\tau A}\mathbf{f}. \tag{A15}$$

Therefore, the unitary evolution of the Hamiltonian $\mathcal{H}$ can be represented by the orthogonal matrix $e^{-\tau A}$. The matrix A can be brought into the standard form (equivalent to the Schur form up to a permutation) by an orthogonal transformation R,

$$RAR^T = \begin{pmatrix} 0 & \mathcal{E} \\ -\mathcal{E} & 0 \end{pmatrix}, \qquad \mathcal{E} = \mathrm{diag}(\varepsilon_1, \ldots, \varepsilon_N), \tag{A16}$$

where $0 \leq \varepsilon_1 \leq \varepsilon_2 \cdots \leq \varepsilon_N$. This transformation corresponds to a basis change of the Majorana fermions $\mathbf{f}' = R\mathbf{f}$, where the new operators $\mathbf{f}'$ also satisfy the anticommutation relations (A10). Going back to the picture of creation and annihilation operators, such a basis transformation takes the form

$$\begin{pmatrix} \mathbf{b}^\dagger \\ \mathbf{b} \end{pmatrix} = W\begin{pmatrix} \mathbf{c}^\dagger \\ \mathbf{c} \end{pmatrix}, \tag{A17}$$

where $(\mathbf{b}^\dagger\ \mathbf{b})^T = (b_1^\dagger \cdots b_N^\dagger\, b_1 \cdots b_N)^T$ and $(\mathbf{c}^\dagger\ \mathbf{c})^T = (c_1^\dagger \cdots c_N^\dagger\, c_1 \cdots c_N)^T$. The unitary matrix W defines a new set of fermionic operators b_j and $b_j^\dagger$, which brings the Hamiltonian (A1) to the diagonal form (A2). The matrix W is related to R through a basis transformation,

$$W = \Omega^\dagger R\Omega. \tag{A18}$$

It is the starting point of our procedure in Sec. IV, which prepares the ground state of the Hamiltonian (A1) on a quantum computer. In summary, W can be calculated in four steps:

(1) Write the quadratic Hamiltonian (A1) in the matrix form (A8).
(2) Find the real antisymmetric matrix A using the matrix form (A8) and Eq. (A13).
(3) Bring A into the standard form (A16) using an orthogonal transformation R.
(4) Calculate the matrix W using R and Eq. (A18).

APPENDIX B: ANCILLA-ASSISTED FERMIONIC GATES

In Sec. V, we discuss how to implement the 2D fermionic Fourier transformation with $O(N^{1.5})$ gates and $O(\sqrt{N})$ circuit depth. A crucial ingredient of our method is a unitary that attaches the corresponding parity operators to the bare hopping terms. This procedure allows efficient implementation of both horizontal and vertical hopping terms on a 2D qubit array. The same strategy can be used to simulate Hamiltonian time evolution—again on a 2D qubit array—by Trotterization, where a single time step can be achieved with only $O(N)$ gates and $O(\sqrt{N})$ depth. Here, we propose an alternative method to implement the hopping terms in a Trotter step using ancilla qubits to store the parities also with $O(N)$ gates and $O(\sqrt{N})$ depth. However, it is less efficient in circuit depth to perform the 2D Fourier transformation compared to the method in Sec. V due to difficulties in parallelization.

We discuss how to implement the hopping terms on a 2D qubit array using the same example of array size 4 × 4. The ancilla qubits move horizontally in the array to facilitate vertical hopping terms; see Fig. 17. The ancilla qubits are initially at the far left side of the qubit array; the states of those ancilla qubits for the right-closed hopping terms are set to $|0\rangle$, while those for the left-closed hopping terms store the parities of the two corresponding rows; e.g., the ancilla on the second row initially stores the parity $Z_{5\text{–}12}$. The vertical hopping terms are implemented with the parity stored in the ancilla qubits, whose states are updated constantly. For example, the vertical hopping term between qubits 3 and 6 in Fig. 17(d) is

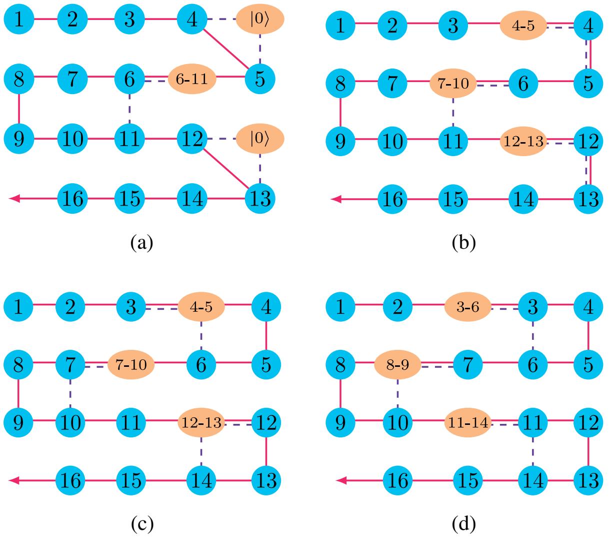

FIG. 17. (a)–(d) Implementing the hopping terms on a 4 × 4 qubit array. The red arrow indicates the direction of the JWT and the system qubits (the blue circles) are numbered by their order in the JWT. The ancilla qubits (the orange ellipses) store the parity information of the system qubits, where j-k denotes the parity $Z_{j\text{-}k} \equiv Z_j \cdots Z_k$. The purple dashed lines represent quantum gates between neighboring qubits.

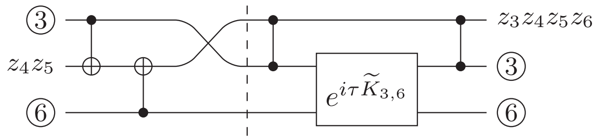

FIG. 18. Quantum circuit to implement the vertical hopping term between qubits 3 and 6.

$$\tilde{K}_{3,6}Z_{4\text{–}5} = -\tilde{K}_{3,6}Z_{3\text{–}6}, \tag{B1}$$

where $\tilde{K}_{3,6} = (X_3X_6 + Y_3Y_6)/2$ denotes the bare hopping term. The time evolution $\exp(-i\tau\tilde{K}_{3,6})$ can be implemented using the second part of the circuit in Fig. 18, where the parity stored in the ancilla qubit is attached to the bare hopping term $\tilde{K}_{3,6}$ by the CZ gates; the first part of the circuit, corresponding to Fig. 17(c), updates the parity and swaps the ancilla qubit with system qubit 3. The order of the two parts in Fig. 18 may be reversed for other hopping terms, e.g., the one between system qubits 7 and 10; see Figs. 17(c) and 17(d). After the ancilla qubits have swept over the entire rows, they are pushed back in the reverse direction and the same procedure starts over again. The vertical hopping terms require $O(N)$ two-qubit gates to implement for a single Trotter step. The horizontal hopping terms can be implemented straightforwardly as long as the two involved qubits are at the same side of the ancilla qubit (no need to update the parity).

In Appendix E, we discuss how to simulate the 2D Fermi-Hubbard model with qubits on a ladder, i.e., two coupled chains. The basic idea is to use the fermionic SWAP gate to change between row-major order and column-major order so that both the horizontal and vertical couplings can be implemented with only local interactions. This idea was also proposed by Kivlichan *et al.* [45] independently. Because such a strategy does not take full advantage of the 2D qubit interactions, the number of gates scales as $O(N^{1.5})$, which is worse than $O(N)$, by using the methods described above and in Sec. V.

APPENDIX C: HAMILTONIAN-BASED FERMIONIC GATES

Using methods in quantum optimal control [64], one can prepare the desired quantum states by applying sequences of Hamiltonian evolutions to initial states that are easy to prepare. When the fermionic Hamiltonians are quadratic in creation and annihilation operators, one can use their matrix representation to find the optimal control sequences.

As a simple example, we discuss how to prepare the ground state of a hopping Hamiltonian on three orbitals,

$$\mathcal{H}_{J_X} = \frac{1}{\sqrt{2}}\left(c_2^\dagger c_1 + c_3^\dagger c_2 + \text{H.c.}\right). \tag{C1}$$

Besides $\mathcal{H}_{J_X}$, we also need $\mathcal{H}_{J_Z}$ in our control set,

$$\mathcal{H}_{J_Z} = c_1^\dagger c_1 - c_3^\dagger c_3, \tag{C2}$$

whose ground states are easy to prepare. Using the normal ordered form in Eq. (A3), we have the matrix representations for $\mathcal{H}_{J_X}$ and $\mathcal{H}_{J_Z}$,

$$J_X = \frac{1}{\sqrt{2}}\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \qquad J_Z = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix}, \tag{C3}$$

which are the same as the corresponding spin-1 angular momentum operators. Using two rotations along the X axis and the Z axis, we have

$$J_X = e^{-i\pi J_Z/2} e^{-i\pi J_X/2} J_Z e^{i\pi J_X/2} e^{i\pi J_Z/2}. \tag{C4}$$

Using the isomorphism (A6), one can prepare eigenstates of the hopping Hamiltonian $\mathcal{H}_{J_X}$ by applying $\mathcal{U}$ to the corresponding eigenstates of $\mathcal{H}_{J_Z}$, where

$$\mathcal{U} = \exp(-i\pi\mathcal{H}_{J_Z}/2)\exp(-i\pi\mathcal{H}_{J_X}/2). \tag{C5}$$

Since J_Z is diagonal, the eigenstates of $\mathcal{H}_{J_Z}$ are easy-to-prepare Slater determinants in the computational basis (3).

As a second example, we discuss the permutation of orbitals,

$$\begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}, \tag{C6}$$

which requires three fermionic SWAP (FSWAP) gates on neighboring qubits in the JWT. We show how it can be achieved with a single multiqubit gate using Hamiltonian evolution by $\mathcal{H}_{J_X}$. The matrix powers of J_X are

$$J_X^2 = \frac{1}{2}\begin{pmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 0 & 1 \end{pmatrix}, \qquad J_X^3 = J_X. \tag{C7}$$

From Eq. (C7), we can derive the exponential of J_X,

$$e^{-i\tau J_X} = 1 - i\sin\tau J_X + (\cos\tau - 1)J_X^2. \tag{C8}$$

For $\tau = \pi$, we have

$$e^{-i\pi J_X} = 1 - 2J_X^2 = -\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}. \tag{C9}$$

Therefore, the unitary that swaps the first orbital with the third orbital is equal to

$$\mathcal{F}_{\text{SWAP}}^{1,3} = \exp\left(-i\pi\sum_{j=1}^{3} c_j^\dagger c_j\right)\exp(-i\pi\mathcal{H}_{J_X}), \tag{C10}$$

where the first term on the right-hand side fixes the extra minus sign in Eq. (C9). The two constituent parts in Eq. (C10) commute with each other, and the first part can be implemented with only single-qubit gates with the JWT.

APPENDIX D: FERMIONIC SWAP GATE

The FSWAP gate on two neighboring qubits takes the form [37]

$$F_{\text{SWAP}}^{q,q+1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}, \tag{D1}$$

where we use the standard order of basis states $|00\rangle, |01\rangle, |10\rangle, |11\rangle$. One way to implement the fermionic SWAP gate is to use the ISWAP gate

$$\text{ISWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & i & 0 \\ 0 & i & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = e^{(i\pi/2)H_{\text{ISWAP}}^{q,q+1}}, \tag{D2}$$

$$H_{\text{ISWAP}}^{q,q+1} = \frac{1}{2}\left(X_q X_{q+1} + Y_q Y_{q+1}\right). \tag{D3}$$

To fix the phases in ISWAP, we introduce

$$e^{(i\pi/4)(Z_q+Z_{q+1})} = \begin{pmatrix} i & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -i \end{pmatrix}, \tag{D4}$$

which commutes with the ISWAP gate. The fermionic SWAP gate then takes the form

$$F_{\text{SWAP}}^{q,q+1} = -ie^{(i\pi/4)(Z_q+Z_{q+1})}e^{(i\pi/2)H_{\text{ISWAP}}^{q,q+1}}. \tag{D5}$$

APPENDIX E: SIMULATING THE 2D HUBBARD MODEL WITH A QUBIT LADDER

Experimentally, it is often easier to build qubits on a ladder (two coupled chains) than on a 2D lattice. The two spin chains in a ladder can be used to map fermionic modes with two different spin states; see Fig. 19. Because there are no hopping terms between the two spin states in the Hubbard model, the quantum state can be mapped using two independent JWTs on the two spin chains. One needs only to implement hopping terms within the chains and interaction terms between corresponding qubits in the two chains. In Ref. [65], the authors discussed physical implementations of analog quantum simulators of the 1D Hubbard model on a qubit ladder. Here, we discuss a digital quantum simulation of the 2D Hubbard model using the same architecture. The challenge is to implement both the horizontal and vertical hopping terms with only local qubit operations. We solve this problem by reordering the spin orbitals encoded in the JWT using the fermionic SWAP gate (see Appendix D). Similar ideas were proposed independently by Kivlichan *et al.* [45].

We demonstrate our approach using an example in which the Fermi-Hubbard model on a 3×3 lattice is mapped to a qubit ladder of size 2×9; see Fig. 19. The spin-up (-down) orbitals are mapped to the upper (lower) chain in the ladder using the JWT. To implement both the horizontal and vertical hopping terms, one can switch between the row-major and column-major orders,

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \xrightarrow{\text{transposition}} \begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{pmatrix}, \qquad \text{(E1)}$$

which corresponds to the following *in situ* transposition of the fermionic orbitals in the JWT:

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 4 & 7 & 2 & 5 & 8 & 3 & 6 & 9 \end{pmatrix}. \qquad \text{(E2)}$$

This permutation can be decomposed into elementary permutations involving only neighboring qubits; see Fig. 20. The pink permutation in Fig. 20 can be implemented using three FSWAP gates involving only neighboring qubits; in Appendix C, we discuss an alternative approach to implementing this permutation using Hamiltonian evolution. The number of two-qubit gates required to implement the *in situ* transposition for the two chains is $N_{\text{trans}} = 2 \times 9 = 18$. The numbers of two-qubit gates needed for the on-site interaction terms and the hopping terms are $N_{\text{int}} = 9$ and $N_{\text{hop}} = 2 \times 2 \times 6 = 24$, respectively. The total number of two-qubit gates needed to simulate a single Trotter step is $N_{\text{Trott}} = N_{\text{trans}} + N_{\text{int}} + N_{\text{hop}} = 51$.

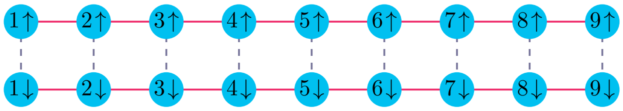

FIG. 19. A 2×9 qubit ladder (two chains) is used to simulate the 2D Hubbard model on a 3×3 lattice. The spin-up and -down orbitals are mapped to the upper and lower qubit chains, respectively, using two independent JWTs. The solid red lines represent nearest-neighbor interactions between qubits in the same JWT, and the dashed blue lines represent ZZ interactions between qubits representing different spin states.

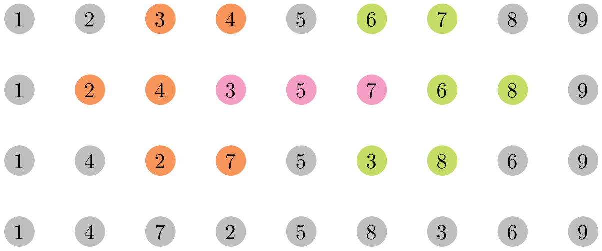

FIG. 20. The *in situ* matrix transposition (E2) can be decomposed into elementary fermionic permutations involving only neighboring qubits (time goes from top to bottom). The first line is the initial ordering and the last line is the target ordering. In each line, the qubits plotted with the same bright color are involved in one elementary permutation, and their positions are updated in the next line. The gray qubits remain in the same place. The pink permutation can be implemented with three FSWAP gates involving only neighboring qubits, or by Hamiltonian evolution, as described in Appendix C. The rest of the permutations require a single FSWAP gate.

To end this appendix, we consider simulating the 2D Fermi-Hubbard model on a lattice of dimension $n_x \times n_y$ using a qubit ladder of length $n_x n_y$. Without loss of generality, we assume that $n_y \geq n_x$. The fermionic orbitals are mapped to qubits in row-major order using the JWT. Implementing the *in situ* matrix transposition on the entire chain requires $O(N^2)$ two-qubit gates and is inefficient. Instead, we can implement the *in situ* matrix transposition on orbitals in neighboring rows. This modification allows one to implement all of the hopping terms on a lattice with only $O(N^{1.5})$ two-qubit gates. For example, a single Trotter step can be implemented with seven constituent operations:

(1) Implement the horizontal hopping terms and the on-site interaction terms.
(2) Implement *in situ* transpositions on spin orbitals on the first and second rows, third, and fourth rows, and so on.
(3) Apply vertical hopping terms between these pairs of rows.
(4) Undo the *in situ* transposition.
(5) Implement *in situ* transposition on spin orbitals on the second and third rows, fourth and fifth rows, and so on.
(6) Apply vertical hopping terms between these pairs of rows.
(7) Undo the *in situ* transposition.

A single *in situ* transposition involving two rows requires $(n_x - 1)n_x/2$ FSWAP gates on neighboring qubits to implement. Because there are $4(n_y - 1)$ *in situ* transpositions in one Trotter step, the total number of two-qubit gates required is $N_{\text{trans}} = 2(n_y - 1)(n_x - 1)n_x = Nn_x - N - 2n_x^2 + 2n_x$, where $N = 2n_y n_x$ is the total number of spin orbitals.

The numbers of two-qubit gates needed for the on-site interaction terms and the hopping terms are $N_{\rm inter} = N/2$ and $N_{\rm hop} = 2(n_y - 1)n_x + 2(n_x - 1)n_y = 2N - 2(n_y + n_x)$, respectively. For square lattices, the total number of two-qubit gates needed for a single Trotter step is $N_{\rm Trott} = N_{\rm trans} + N_{\rm inter} + N_{\rm hop} \simeq N^{3/2}/\sqrt{2} + N/2$. For example, the total number of two-qubit gates is 275 for 50 spin orbitals on a square lattice with a size of 5 × 5.

APPENDIX F: NUMERICAL STUDY OF THE TROTTER ERROR

One way to prepare strongly correlated quantum states is by adiabatic evolution. On a digital quantum computer, we can simulate the evolution by dividing it into discrete time steps and approximating the Hamiltonian within each time step. Here, we numerically study the errors that arise from using a second-order Trotter formula to simulate adiabatic state preparation for the ground states of the Fermi-Hubbard model.

For simplicity, we consider only the hopping terms:

$$\mathcal{H}_{\rm hop} = -\sum_{\langle j,k\rangle,\sigma} t_{jk}\left(c^\dagger_{j,\sigma}c_{k,\sigma} + {\rm H.c.}\right), \tag{F1}$$

and the interaction term

$$\mathcal{H}_{\rm int} = U\sum_j n_{j,\uparrow}n_{j,\downarrow} - \mu\sum_{j,\sigma} n_{j,\sigma}. \tag{F2}$$

We introduce a chemical potential term in $\mathcal{H}_{\rm int}$ to fix the particle number of the ground state so that it occurs at half filling. Since the ground state of $\mathcal{H}_{\rm hop}$ is also half filled and both $\mathcal{H}_{\rm hop}$ and $\mathcal{H}_{\rm int}$ conserve particle number, the particle number stays constant throughout the evolution. To prepare a ground state of $\mathcal{H}_{\rm int}$ starting from a ground state of $\mathcal{H}_{\rm hop}$, we apply the time-dependent Hamiltonian

$$\mathcal{H}(s) = (1-s)\mathcal{H}_{\rm hop} + s\mathcal{H}_{\rm int}, \tag{F3}$$

where $s = t/T \in [0,1]$, with T being the total evolution time. If we split T into n time steps, then a single Trotter step takes the form

$$e^{-i(1-s)\Delta t\mathcal{H}_{\rm hop}}e^{-is\Delta t\mathcal{H}_{\rm int}}, \tag{F4}$$

where $\Delta t = T/n$.

For the purposes of this paper, we ignore the issue of approximating the two time-evolution operators in Eq. (F4). We use the parameter settings

$$t_{jk} = 1 \quad \text{for all cases of } \langle j,k\rangle, \quad U = 4, \quad \text{and} \quad \mu = 1 \tag{F5}$$

on two-dimensional grids of various sizes. We use the open-source package OpenFermion [38] to initialize the Hamiltonians of the Hubbard model and compute their ground states. Then, we use the open-source package QuTiP

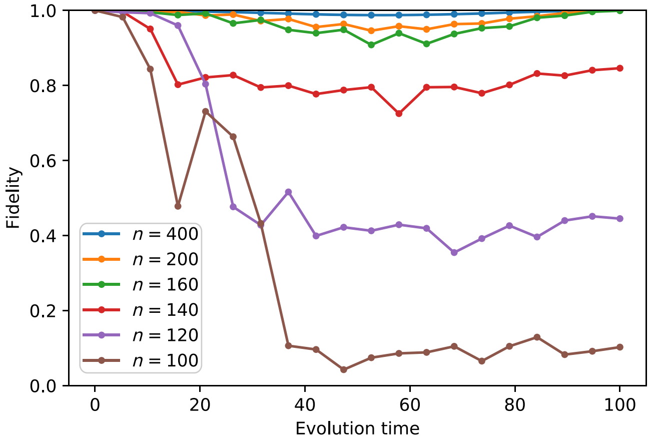

FIG. 22. Fidelity with states from adiabatic evolution for various Trotter step numbers on a 3 × 2 grid.

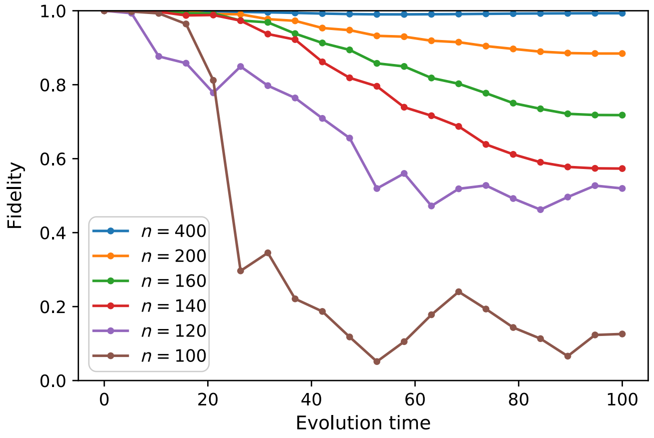

FIG. 21. Fidelity with states from adiabatic evolution for various Trotter step numbers on a 2 × 2 grid.

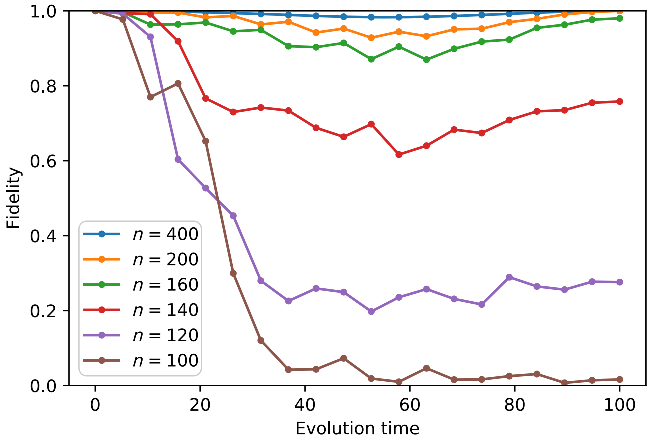

FIG. 23. Fidelity with states from adiabatic evolution for various Trotter step numbers on a 4 × 2 grid.

[66] to compute the exact evolution by the time-dependent Hamiltonian (F3), starting with the ground state of $\mathcal{H}_{\rm hop}$. We adjust the total evolution time T until the final state is within the ground space of $\mathcal{H}_{\rm int}$, indicating that the evolution is adiabatic. For all of the grid sizes in our work, $T = 100$ is sufficient. Finally, we approximate the evolution using the second-order Trotter formula with various values of n, the number of steps. Higher values of n give a better approximation. For each step size, we record the quantum state 20 times throughout the evolution and compute the fidelity of these states with the states from the true adiabatic evolution. The numerical results for grid sizes 2×2, 3×2, and 4×2 are shown in Figs. 21, 22, and 23, respectively.

APPENDIX G: REARRANGED QUANTUM CIRCUITS FOR DIFFERENT HOPPING TERMS

In Sec. V, we provide an algorithm to implement the 2D fermionic Fourier transformation with $O(N^{1.5})$ gates and $O(\sqrt{N})$ depth. A crucial step is a unitary transformation Γ, diagonalized in the computation basis, that introduces the desired parities to the bare vertical hopping terms. Therein, we provide a method to implement Γ with $O(N)$ gates and $O(\sqrt{N})$ depth by introducing one ancilla per row to store the parities of the system qubits. In each time step, the ancilla qubits interact with system qubits in a particular column with the circuit in Fig. 24. The first and second parts of the circuit consist of CZ gates between each system qubit and the ancilla qubits below, starting with the next odd-numbered row. This circuit is used to demonstrate the hopping terms between the first and second rows, and a rearranged circuit in Fig. 25 is used for hopping terms between the second and third rows. Here, we provide the other arrangements of the same circuit for the remaining pairs of rows that are adjacent to each other. The circuit in Fig. 24 can be transformed into those in Figs. 26 and 28 by regrouping neighboring pairs of gates, which can be generalized to any right-closed hopping term for a system with an even number of rows. Similarly, the first three pairs of gates and the last two gates in the second part in Fig. 25 can be regrouped into gates in the corresponding positions in Fig. 27, which can also be generalized to any left-closed hopping term for larger system sizes. Each of the remaining gates in Fig. 25 (except for the third part) involve the first system qubit and an ancilla qubit below; these gates do not affect any left-closed hopping term, either.

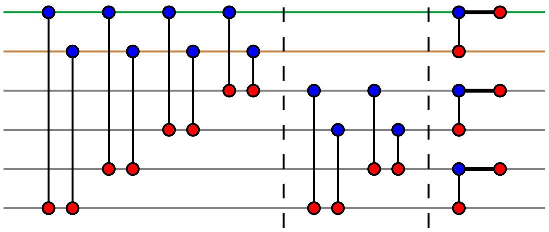

FIG. 24. The CZ gates between the system (blue) and ancilla (red) qubits in a single time step.

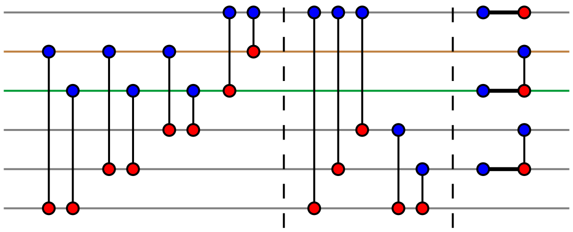

FIG. 25. Rearranged circuit for hopping terms between the second and third rows.

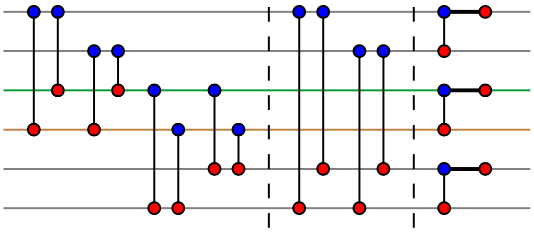

FIG. 26. Rearranged circuit for hopping terms between the third and fourth rows.

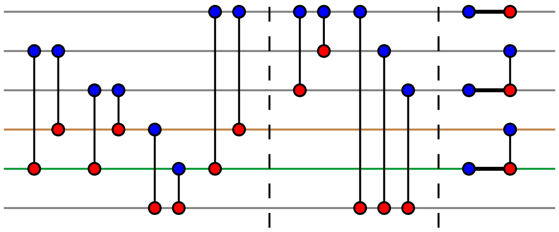

FIG. 27. Rearranged circuit for hopping terms between the fourth and fifth rows.

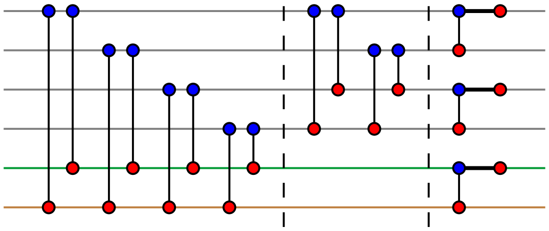

FIG. 28. Rearranged circuit for hopping terms between the fifth and sixth rows.

[1] P. Hohenberg and W. Kohn, Inhomogeneous electron gas, Phys. Rev. 136, B864 (1964).

[2] J. Quintanilla and C. Hooley, The strong-correlations puzzle, Phys. World **22**, 32 (2009).

[3] D. N. Basov, R. D. Averitt, D. van der Marel, M. Dressel, and K. Haule, Electrodynamics of correlated electron materials, Rev. Mod. Phys. **83**, 471 (2011).

[4] R. P. Feynman, Simulating physics with computers, Int. J. Theor. Phys. **21**, 467 (1982).

[5] I. Georgescu, S. Ashhab, and F. Nori, Quantum simulation, Rev. Mod. Phys. **86**, 153 (2014).

[6] M. B. Hastings, D. Wecker, B. Bauer, and M. Troyer, Improving quantum algorithms for quantum chemistry, Quantum Inf. Comput. **15**, 1 (2015).

[7] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets, Nature (London) **549**, 242 (2017).

[8] M. Reiher, N. Wiebe, K. M. Svore, D. Wecker, and M. Troyer, Elucidating reaction mechanisms on quantum computers, Proc. Natl. Acad. Sci. U.S.A. **114**, 7555 (2017).

[9] R. Babbush, N. Wiebe, J. McClean, J. McClain, H. Neven, and G. K.-L. Chan, Low-Depth Quantum Simulation of Electronic Structure, arXiv:1706.00023 [Phys. Rev. X (to be published)].

[10] J. Hubbard, Electron correlations in narrow energy bands, Proc. R. Soc. A **276**, 238 (1963).

[11] D. Wecker, M. B. Hastings, N. Wiebe, B. K. Clark, C. Nayak, and M. Troyer, Solving strongly correlated electron models on a quantum computer, Phys. Rev. A **92**, 062318 (2015).

[12] S. M. Griffin, P. Staar, T. C. Schulthess, M. Troyer, and N. A. Spaldin, A bespoke single-band Hubbard model material, Phys. Rev. B **93**, 075115 (2016).

[13] M. S. Hybertsen, E. B. Stechel, M. Schluter, and D. R. Jennison, Renormalization from density-functional theory to strong-coupling models for electronic states in Cu-O materials, Phys. Rev. B **41**, 11068 (1990).

[14] E. Dagotto, Correlated electrons in high-temperature superconductors, Rev. Mod. Phys. **66**, 763 (1994).

[15] E. H. Lieb and F. Y. Wu, Absence of Mott Transition in an Exact Solution of the Short-Range, One-Band Model in One Dimension, Phys. Rev. Lett. **20**, 1445 (1968).

[16] T. Maier, M. Jarrell, T. Pruschke, and J. Keller, *d*-Wave Superconductivity in the Hubbard Model, Phys. Rev. Lett. **85**, 1524 (2000).

[17] B.-X. Zheng and G. K.-L. Chan, Ground-state phase diagram of the square lattice Hubbard model from density matrix embedding theory, Phys. Rev. B **93**, 035126 (2016).

[18] C. J. Jia, B. Moritz, C.-C. Chen, B. S. Shastry, and T. P. Devereaux, Fidelity study of the superconducting phase diagram in the two-dimensional single-band Hubbard model, Phys. Rev. B **84**, 125113 (2011).

[19] S. Yamada, T. Imamura, and M. Machida, in *Proceedings of the ACM/IEEE SC 2005 Conference in Supercomputing, Seattle, 2005* (IEEE, New York, 2005), p. 44.

[20] M. Troyer and U.-J. Wiese, Computational Complexity and Fundamental Limitations to Fermionic Quantum Monte Carlo Simulations, Phys. Rev. Lett. **94**, 170201 (2005).

[21] S. R. White, Density-matrix algorithms for quantum renormalization groups, Phys. Rev. B **48**, 10345 (1993).

[22] M. H. Hettler, M. Mukherjee, M. Jarrell, and H. R. Krishnamurthy, Dynamical cluster approximation: Nonlocal dynamics of correlated electron systems, Phys. Rev. B **61**, 12739 (2000).

[23] G. Kotliar, S. Y. Savrasov, K. Haule, V. S. Oudovenko, O. Parcollet, and C. A. Marianetti, Electronic structure calculations with dynamical mean-field theory, Rev. Mod. Phys. **78**, 865 (2006).

[24] G. Knizia and G. K.-L. Chan, Density Matrix Embedding: A Simple Alternative to Dynamical Mean-Field Theory, Phys. Rev. Lett. **109**, 186404 (2012).

[25] G. Ortiz, J. E. Gubernatis, E. Knill, and R. Laflamme, Quantum algorithms for fermionic simulations, Phys. Rev. A **64**, 022319 (2001).

[26] J. D. Whitfield, J. Biamonte, and A. Aspuru-Guzik, Simulation of electronic structure Hamiltonians using quantum computers, Mol. Phys. **109**, 735 (2011).

[27] A. A. Houck, H. E. Türeci, and J. Koch, On-chip quantum simulation with superconducting circuits, Nat. Phys. **8**, 292 (2012).

[28] R. Barends *et al.*, Digitized adiabatic quantum computing with a superconducting circuit, Nature (London) **534**, 222 (2016).

[29] A. Mazurenko, C. S. Chiu, G. Ji, M. F. Parsons, M. Kanász-Nagy, R. Schmidt, F. Grusdt, E. Demler, D. Greif, and M. Greiner, A cold-atom Fermi-Hubbard antiferromagnet, Nature (London) **545**, 462 (2017).

[30] S. Seo, X. Lu, J.-X. Zhu, R. R. Urbano, N. Curro, E. D. Bauer, V. A. Sidorov, L. D. Pham, T. Park, Z. Fisk, and J. D. Thompson, Disorder in quantum critical superconductors, Nat. Phys. **10**, 120 (2014).

[31] B. Sacépé, T. Dubouchet, C. Chapelier, M. Sanquer, M. Ovadia, D. Shahar, M. Feigel'man, and L. Ioffe, Localization of preformed Cooper pairs in disordered superconductors, Nat. Phys. **7**, 239 (2011).

[32] S. S. Kondov, W. R. McGehee, W. Xu, and B. DeMarco, Disorder-Induced Localization in a Strongly Correlated Atomic Hubbard Gas, Phys. Rev. Lett. **114**, 083002 (2015).

[33] J.-X. Zhu, I. Martin, and A. R. Bishop, Spin and Charge Order around Vortices and Impurities in High-T_c Superconductors, Phys. Rev. Lett. **89**, 067003 (2002).

[34] B. M. Andersen, P. J. Hirschfeld, A. P. Kampf, and M. Schmid, Disorder-Induced Static Antiferromagnetism in Cuprate Superconductors, Phys. Rev. Lett. **99**, 147002 (2007).

[35] B. Bauer, D. Wecker, A. J. Millis, M. B. Hastings, and M. Troyer, Hybrid Quantum-Classical Approach to Correlated Materials, Phys. Rev. X **6**, 031045 (2016).

[36] J. Preskill, Quantum computing in the NISQ era and beyond, arXiv:1801.00862.

[37] F. Verstraete, J. I. Cirac, and J. I. Latorre, Quantum circuits for strongly correlated quantum systems, Phys. Rev. A **79**, 032316 (2009).

[38] J. R. McClean, I. D. Kivlichan, D. S. Steiger, Y. Cao, E. S. Fried, C. Gidney, T. Häner, V. Havlíček, Z. Jiang, M. Neeley, J. Romero, N. Rubin, N. P. D. Sawaya, K. Setia, S. Sim, W. Sun, K. Sung, and R. Babbush, OpenFermion: The electronic structure package for quantum computers, arXiv:1710.07629.

[39] B. P. Lanyon, J. D. Whitfield, G. G. Gillett, M. E. Goggin, M. P. Almeida, I. Kassal, J. D. Biamonte, M. Mohseni, B. J. Powell, M. Barbieri, A. Aspuru-Guzik, and A. G. White, Towards quantum chemistry on a quantum computer, Nat. Chem. **2**, 106 (2010).

[40] S. B. Bravyi and A. Y. Kitaev, Fermionic quantum computation, Ann. Phys. (Amsterdam) **298**, 210 (2002).

[41] J. T. Seeley, M. J. Richard, and P. J. Love, The Bravyi-Kitaev transformation for quantum computation of electronic structure, J. Chem. Phys. **137**, 224109 (2012).

[42] R. C. Ball, Fermions without Fermion Fields, Phys. Rev. Lett. **95**, 176407 (2005).

[43] F. Verstraete and J. I. Cirac, Mapping local Hamiltonians of fermions to local Hamiltonians of spins, J. Stat. Mech. (2005) P09012.

[44] J. D. Whitfield, V. Havlíček, and M. Troyer, Local spin operators for fermion simulations, Phys. Rev. A **94**, 030301 (2016).

[45] I. D. Kivlichan, J. McClean, N. Wiebe, C. Gidney, A. Aspuru-Guzik, G. K. -L. Chan, and R. Babbush, Quantum simulation of electronic structure with linear depth and connectivity, arXiv:1711.04789 [Phys. Rev. Lett. (to be published)].

[46] G. Wendin, Quantum information processing with superconducting circuits: A review, Rep. Prog. Phys. **80**, 106001 (2017).

[47] V. Bach, E. H. Lieb, and J. P. Solovej, Generalized Hartree-Fock theory and the Hubbard model, J. Stat. Phys. **76**, 3 (1994).

[48] E. Greplová, Master's thesis, Ludwig-Maximilians-Universität München, 2013.

[49] J. Bardeen, L. N. Cooper, and J. R. Schrieffer, Microscopic theory of superconductivity, Phys. Rev. **106**, 162 (1957).

[50] S. Bravyi and D. Gosset, Complexity of quantum impurity problems, Commun. Math. Phys. **356**, 451 (2017).

[51] B.-X. Zheng, C.-M. Chung, P. Corboz, G. Ehlers, M.-P. Qin, R. M. Noack, H. Shi, S. R. White, S. Zhang, and G. K.-L. Chan, Stripe order in the underdoped region of the two-dimensional Hubbard model, Science **358**, 1155 (2017).

[52] C. J. Halboth and W. Metzner, Renormalization-group analysis of the two-dimensional Hubbard model, Phys. Rev. B **61**, 7364 (2000).

[53] T. A. Maier, M. Jarrell, T. C. Schulthess, P. R. C. Kent, and J. B. White, Systematic Study of *d*-Wave Superconductivity in the 2D Repulsive Hubbard Model, Phys. Rev. Lett. **95**, 237001 (2005).

[54] D. S. Abrams and S. Lloyd, Simulation of Many-Body Fermi Systems on a Universal Quantum Computer, Phys. Rev. Lett. **79**, 2586 (1997).

[55] D. S. Abrams and S. Lloyd, Quantum Algorithm Providing Exponential Speed Increase for Finding Eigenvalues and Eigenvectors, Phys. Rev. Lett. **83**, 5162 (1999).

[56] S. Sachdev, in *String Theory and Its Applications*, edited by M. Dine, T. Banks, and S. Sachdev (World Scientific, Singapore, 2012), p. 559.

[57] H. F. Trotter, On the product of semi-groups of operators, Proc. Am. Math. Soc. **10**, 545 (1959).

[58] M. Suzuki, Generalized Trotter's formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems, Commun. Math. Phys. **51**, 183 (1976).

[59] R. Babbush, J. McClean, D. Wecker, A. Aspuru-Guzik, and N. Wiebe, Chemical basis of Trotter-Suzuki errors in quantum chemistry simulation, Phys. Rev. A **91**, 022311 (2015).

[60] D. Poulin, M. B. Hastings, D. Wecker, N. Wiebe, A. C. Doberty, and M. Troyer, The Trotter step size required for accurate quantum simulation of quantum chemistry, Quantum Inf. Comput. **15**, 361 (2015).

[61] D. A. Wollman, D. J. Van Harlingen, W. C. Lee, D. M. Ginsberg, and A. J. Leggett, Experimental Determination of the Superconducting Pairing State in YBCO from the Phase Coherence of YBCO-Pb dc SQUIDs, Phys. Rev. Lett. **71**, 2134 (1993).

[62] C. C. Tsuei and J. R. Kirtley, Pairing symmetry in cuprate superconductors, Rev. Mod. Phys. **72**, 969 (2000).

[63] D. J. Scalapino, The case for $d_{x^2-y^2}$ pairing in the cuprate superconductors, Phys. Rep. **250**, 329 (1995).

[64] J. Werschnik and E. K. U. Gross, Quantum optimal control theory, arXiv:0707.1883.

[65] J.-M. Reiner, M. Marthaler, J. Braumüller, M. Weides, and G. Schön, Emulating the one-dimensional Fermi-Hubbard model by a double chain of qubits, Phys. Rev. A **94**, 032338 (2016).

[66] J. R. Johansson, P. D. Nation, and F. Nori, QuTiP2: A PYTHON framework for the dynamics of open quantum systems, Comput. Phys. Commun. **184**, 1234 (2013).