Preparing Bethe Ansatz Eigenstates on a Quantum Computer

John S. Van Dyke,[1] George S. Barron,[1] Nicholas J. Mayhall,[2] Edwin Barnes,[1] and Sophia E. Economou[1]

[1]*Department of Physics, Virginia Tech, Blacksburg, VA 24061*
[2]*Department of Chemistry, Virginia Tech, Blacksburg, VA 24061*
(Dated: July 5, 2021)

Several quantum many-body models in one dimension possess exact solutions via the Bethe ansatz method, which has been highly successful for understanding their behavior. Nevertheless, there remain physical properties of such models for which analytic results are unavailable, and which are also not well-described by approximate numerical methods. Preparing Bethe ansatz eigenstates directly on a quantum computer would allow straightforward extraction of these quantities via measurement. We present a quantum algorithm for preparing Bethe ansatz eigenstates of the XXZ spin chain that correspond to real-valued solutions of the Bethe equations. The algorithm is polynomial in the number of T gates and circuit depth, with modest constant prefactors. Although the algorithm is probabilistic, with a success rate that decreases with increasing eigenstate energy, we employ amplitude amplification to boost the success probability. The resource requirements for our approach are lower than other state-of-the-art quantum simulation algorithms for small error-corrected devices, and thus may offer an alternative and computationally less-demanding demonstration of quantum advantage for physically relevant problems.

Quantum computers hold the promise of transformative applications in a variety of fields including cryptanalysis [1], quantum chemistry [2, 3], materials science [4, 5], and potentially combinatorial optimization [6, 7]. To realize the full potential of quantum computing, large-scale, fault-tolerant devices will ultimately be necessary. As these do not yet exist, much recent work has studied possible near-term applications in the present era of noisy, intermediate-scale quantum computers (NISQ) [8, 9]. In this context, a key question concerns the demonstration of 'quantum advantage' – that is, the ability to perform computations that can not be done efficiently with classical methods. Recently, quantum advantage was shown for a superconducting processor sampling random quantum circuits [10] and photonic-based Gaussian boson sampling [11]. Although these are important achievements, the specific tasks performed were not closely related to the practical applications mentioned above, but were essentially designed for the purpose of demonstrating advantage. Thus, the realization of quantum advantage for a problem of practical interest remains open.

The concept of hybrid quantum-classical algorithms has served as a guiding principle for near-term applications. This approach divides a given computational task into sub-tasks that are separately handled by a quantum and a classical processor, so as to maximize the benefits of the quantum computer while limiting the overhead for operations that can be efficiently performed classically. A prominent example of this idea is the variational quantum eigensolver (VQE), in which the quantum processor calculates energy expectation values of a Hamiltonian with respect to a variational wave function whose parameters (or even form) is optimized classically [12–15]. Although VQEs for small (classically tractable) problems have been successfully demonstrated [16, 17], the scalability of solving the high-dimensional optimization problem to non-classically-simulable systems remains unclear, for instance due to barren plateaus [18] and the generic

difficulty of performing such optimizations.

A question of increasing interest is what applications become feasible with small-scale error-corrected devices, i.e., in the intermediate era between NISQ and fully scaled-up fault-tolerant quantum computers. Recent estimates suggest that algorithms that provide only a quadratic speedup over classical methods may have difficulty achieving quantum advantage on problem sizes accessible with small devices [19–21]. The simulation of quantum systems, on the other hand, can yield exponential improvement over conventional approaches. These still require formidable resources, despite recent algorithmic advances [22–25]. This motivates the search for physically-interesting problems and algorithms that can lead to quantum advantage with fewer resources in the near future.

We propose the study of Bethe ansatz (BA) states on a quantum computer as a computationally less-demanding route to the demonstration of quantum advantage for problems relevant to physics. The BA technique yields exact solutions to a class of one-dimensional quantum many-body models, including the Heisenberg and Hubbard models, among others [26–29]. The resulting wave functions depend on algebraic equations that can be efficiently solved classically. The exponential growth of the Hilbert space with the system size L has historically limited the direct computational studies of the eigenstates to small systems. Instead, various mathematical techniques have been extensively developed to access physical quantities in the thermodynamic limit $L \to \infty$, bypassing the calculation of the wave function itself. While many different quantities can be determined, the difficulty of their calculation varies widely. In particular, arbitrary-range and higher-order correlation functions have been very challenging to access, and remain an active area of research [30–32]. Quantum computers, however, can compute such correlation functions [5, 33] straightforwardly, thus suggesting the possibility of quantum advantage for

this task. The importance of higher-order correlation functions for strongly correlated systems has recently been emphasized [34]. Calculating such observables using a quantum computer in turn hinges on the possibility of efficiently preparing the Bethe ansatz states.

To this end, we demonstrate an efficient quantum algorithm that can prepare a subset of the Bethe ansatz eigenstates of the one-dimensional XXZ chain, a model which is fundamental to the study of quantum magnetism. Our algorithm uses the so-called coordinate Bethe ansatz method, and has polynomial scaling in the circuit depth and T-gate complexity, along with low constant prefactors. As we show with explicit gate counts for the corresponding circuits, the approach scales to large enough systems for the calculation of classically inaccessible quantities in near-term error-corrected devices. While the algorithm we present is probabilistic, we also show that amplitude amplification can be used to increase the success rate [35].

Algorithms have been previously given (and also implemented) that prepare exact eigenstates of quantum many-body models [36–38]. However, these were limited to cases that are equivalent to non-interacting fermions, for instance, under the Jordan-Wigner mapping. In contrast, the XXZ model corresponds to an interacting fermionic problem, which is computationally much more difficult. We note that the possibility of constructing circuits to diagonalize Bethe ansatz-solvable models and measure challenging correlation functions was previously suggested, though without an indication of how this could be done [36].

The importance of Bethe ansatz-solvable models for benchmarking NISQ devices has been previously recognized [39, 40], as they provide exact values for quantities (such as the energy) to compare against the results of noisy quantum computations. On the other hand, the direct preparation of Bethe ansatz states has been relatively unexplored. This question was recently studied in Ref. [41], which considered treating Bethe ansatz states variationally (using the algebraic Bethe ansatz) and concluded that the approach was not scalable. Furthermore, that work did not make a connection to the possibility of quantum advantage. Apart from direct preparation of Bethe ansatz eigenstates, other works have considered variational approaches using generic ansatzes [39, 40, 42–44]. The comparison of the computational complexity of these methods to that of the direct construction is an interesting question for future studies, as are probabilistic algorithms for preparing other strongly-correlated states [45].

The paper is organized as follows. Section I introduces the XXZ model and the elements of the Bethe ansatz solution needed for the construction of the algorithm. Section II describes the Bethe ansatz state preparation algorithm. Section III presents numerical results that validate the method and studies its success probability. This section also includes resource estimates for classically intractable problem sizes. Section IV describes the

amplitude amplification procedure for our algorithm, and presents numerical calculations that confirm its success. Section V compares our algorithm with conceptually simpler but less efficient approaches to the same task, explicitly verifying the enormous speedup of our method. Section VI argues that quantum advantage can be achieved with Bethe state preparation by comparison with classical computational methods, and presents additional applications of the algorithm. Finally, we conclude in Section VII.

I. MODEL AND SOLUTION

We consider the one-dimensional spin-$1/2$ XXZ chain on L sites with periodic boundary conditions, whose Hamiltonian is given by

$$H = \sum_{i=0}^{L-1} J_{xy} \left(S_i^x S_{i+1}^x + S_i^y S_{i+1}^y \right) + J_z S_i^z S_{i+1}^z, \qquad (1)$$

with $S_L^\alpha \equiv S_0^\alpha$ ($\alpha = x, y, z$). The exact solution of this model via the Bethe ansatz method was presented in Ref. [46], and many introductions to the problem (in both the coordinate and algebraic formulations) exist [47–51]. We follow the account of the Bethe ansatz method given in Ref. [49]. The eigenstates of the above Hamiltonian are given by

$$\psi(x_1, \ldots, x_M) = \sum_P A_P \exp \left[i \sum_{j=1}^M k_{Pj} x_j \right], \qquad (2)$$

where $x_1, \ldots, x_M$ label the positions of the M down spins in the chain (the Hamiltonian conserves the z component of the total spin, $S_{tot}^z = \sum_i S_i^z$) and the momenta k_i label the different states. The wave function of Eq. (2) gives the amplitude for the M down spins to occur on the sites $x_j = 0, \ldots, L-1$. The summation here is over the $M!$ permutations of the down spin sites. These permutations arise from the fact that, within Bethe ansatz models, scattering processes exchange momenta between particles, but do not alter their magnitudes. The coefficients A_P are related by

$$\frac{A_P}{A_{P'}} = -\frac{1 + e^{i(k_{Pl} + k_{P'l})} - 2\Delta e^{ik_{Pl}}}{1 + e^{i(k_{Pl} + k_{P'l})} - 2\Delta e^{ik_{P'l}}}$$
$$\equiv -e^{-i\Theta(k_{Pl}, k_{P'l})}. \qquad (3)$$

To fix the coefficients, we take $A_I = 1$, where I is the identity permutation. Here $\Delta = J_z/J_{xy}$ is the anisotropy in the interactions, and P and P' are permutations that differ by a single transposition between adjacent elements, $P(l+1) = P'(l)$ and $P(l) = P'(l+1)$. The momenta k_i are constrained by the quantization conditions,

$$L k_i = 2\pi I_i + \sum_j \Theta(k_i, k_j), \qquad (4)$$

with I_i an integer (half-integer) for M odd (even). Physically, these constraints on k_i arise from imposing periodic boundary conditions on the model. Eqs. (3) and (4) are a set of algebraic equations (the Bethe equations) for the quantum numbers $\{k_i\}$. In general, these equations admit complex solutions, but for the special case when all $\{k_i\}$ are real, $\Theta(k_i, k_j)$ is also real and is given by

$$\Theta(k_i, k_j) = 2\arctan\left(\frac{\Delta \sin(\frac{k_i - k_j}{2})}{\Delta \cos(\frac{k_i - k_j}{2}) - \cos(\frac{k_i + k_j}{2})}\right). \quad (5)$$

In the spirit of hybrid quantum-classical algorithms, we solve the Bethe equations classically to obtain the momenta $\{k_i\}$ and phases $\Theta(k_i, k_j)$. These values are then used as input to our quantum algorithm for generating the corresponding eigenstate. Our algorithm allows for the preparation of Bethe ansatz eigenstates for which the $\{k_i\}$ are real, such that A_P and $e^{ik_{P_j}x_j}$ amount to complex phases applied to the second-quantized basis states of the system.

II. BETHE ANSATZ STATE PREPARATION ALGORITHM

The quantum algorithm for preparing a Bethe ansatz state consists of several steps, and is summarized in Algorithm 1. The general structure of our approach is inspired by the linear combination of unitaries (LCU) method for Hamiltonian simulation [52, 53]. However, a key difference is that our algorithm aims to generate specific quantum states starting from a particular initial state, rather than compiling a generic unitary evolution operator. In addition to the L qubits representing the system, a register of M^2 ancilla qubits are used to label the different permutation terms in Eq. (2). By preparing a superposition of the allowed label values on these ancillas, using these to apply controlled operations on the system, and finally disentangling the label and system registers, we perform the summation over all permutations present in Eq. (2). This process is facilitated by introducing a second ancillary register of M qubits that we call the "faucet register", along with one additional ancilla work qubit. Thus, the algorithm requires a total of $M^2 + M + 1$ ancilla qubits.

Algorithm 1 Bethe state preparation

1: Prepare the Dicke state $|D_{L,M}\rangle$ on the system qubits
2: Create permutation labels while applying pieces of A_P
3: Apply $e^{ik_{P_j}x_j}$ using the "faucet" method
4: Reverse permutation label (without phases)
5: Measure permutation label, with success on $|00\cdots0\rangle$

The algorithm begins by preparing the Dicke state on L sites with M down spins, $|D_{L,M}\rangle$. Relabeling $|\uparrow\rangle \equiv |0\rangle$, $|\downarrow\rangle \equiv |1\rangle$, $|D_{L,M}\rangle$ is the equal superposition (that is, without relative phases) of all basis states on L qubits with Hamming weight M. This state forms the underlying

"canvas" on which the phases in Eq. (2) are applied. Dicke state preparation can be accomplished using the recent deterministic algorithm of Ref. [54], for which the gate count was improved in Ref. [55]. This algorithm uses an inductive method to prepare smaller Dicke states which are subsequently combined to yield the desired one. We have used this algorithm in our explicit circuit constructions, though any other deterministic method of preparing $|D_{L,M}\rangle$ would also work.

As discussed above, the amplitudes that must be applied to $|D_{L,M}\rangle$ to generate a Bethe ansatz state depend on the permutations $\{P\}$ of M objects. We use the permutation label register to create the different permutations and their associated phases A_P. Naively, one could use an integer encoded in a binary representation to label each of the permutations. The difficulty with this approach is that the number of permutations is $M!$, so that imprinting the phases A_P and $e^{ik_{P_j}x_j}$ onto $|D_{L,M}\rangle$ would require combinatorially many operations. This leads to circuit depths and complexities that are superexponential in M, quickly becoming unfeasible as M grows (we explore a concrete realization of this approach in Section V). To overcome this fundamental limitation of this method and design an efficient algorithm, we introduce a conceptually distinct approach for labeling the permutations. Rather than assigning an arbitrary number to a given permutation, we implement its explicit action on the string consisting of the numbers $1, \ldots, M$. As described below, this allows for an efficient generation of the permutation labels, while also generating the distinct A_P simultaneously.

The permutation label register consists of M subregisters, each of which can store an integer value $k \in \{1, \ldots, M\}$. To represent a valid permutation, the subregisters must contain distinct values (for instance, $|213\rangle$ is valid whereas $|233\rangle$ is not). We use a one-hot encoding such that each subregister consists of M qubits, and the number k is represented by a 1 on the kth qubit and 0s on the rest. Thus, for $M = 3$ the allowed states on each subregister are $|1\rangle \equiv |001\rangle$, $|2\rangle \equiv |010\rangle$, and $|3\rangle \equiv |100\rangle$. This one-hot encoding requires M^2 qubits to represent the complete label. The use of the one-hot encoding facilitates a trade-off between time and space resources [22, 56, 57] by reducing the number of controls required to implement the necessary phase gates.

The goal of step 2 of Algorithm 1 is to create the state $\frac{1}{\sqrt{M!}}\sum_P A_P |P\rangle$ on the permutation label register. The phases A_P are kicked back onto the system qubits, while the $|P\rangle$ are used to apply the conditional gates needed in step 3, as explained below. For clarity, we first describe the construction of the equal superposition of all permutation labels. We then show how to slightly modify this procedure to simultaneously generate the phases A_P for all $M!$ permutations. We use an iterative method to construct the permutation label state starting from the vacuum state $|00\ldots0\rangle$ on M^2 qubits. The complete label superposition state is built up sequentially from the first (rightmost) subregister to the last (leftmost) using

a series of exchange-type gates. We describe the method inductively as follows. The zeroth sublabel is prepared by setting the zeroth qubit of the zeroth subregister to 1 (in the following, the index k is enumerated starting from 0). Assume the kth sublabel (i.e., an equal super-position of permutations of integers 1 through $k+1$) has been constructed on the $k+1$ rightmost subregisters. Set the $(k+1)$th qubit of the $(k+1)$th subregister to 1, thus introducing the next integer value to be included in the permutation label state. Perform the exchange-type ASWAP gate [58, 59],

$$A(\theta, \phi) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & e^{i\phi}\sin(\theta) & 0 \\ 0 & e^{-i\phi}\sin(\theta) & -\cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \qquad (6)$$

between the $(k+1)$th qubits of subregisters $k+1$ and k, with $\theta = \arccos(1/\sqrt{k+2})$, $\phi = 0$. This generates a superposition state consisting of two sets of terms: those in which the 1 remains in the $(k+1)$th subregister and those in which it is transferred to the kth. In the latter

case, the $(k+1)$th subregister now contains all 0s, while the kth has two qubits with 1, which is not valid. This is fixed by applying controlled-SWAP gates between all qubits $l < k+1$ in subregisters k and $k+1$, controlled on the state of qubit $k+1$ in subregister k. Taken together, these operations produce a partial SWAP between subregisters $k+1$ and k,

$$\begin{aligned} &|k+1\rangle_{k+1}|m\rangle_k \\ &\rightarrow \frac{1}{\sqrt{k+2}}|k+1\rangle_{k+1}|m\rangle_k + \sqrt{\frac{k+1}{k+2}}|m\rangle_{k+1}|k+1\rangle_k, \end{aligned} \tag{7}$$

where $|i\rangle_j$ is the one-hot encoded state for i on subregister j, and $m < k$ by construction. One repeats this partial swapping process, now between subregisters k and $k-1$, then between $k-1$ and $k-2$, and so on, until the last register has been swapped. By implementing the inductive process up to the $(M-1)$th subregister, the complete equally-weighted superposition of permutation labels is formed. As an example, for $M = 3$ the above algorithm generates the following sequence of state transformations:

$$\begin{aligned}
|000\rangle|000\rangle|000\rangle &\xrightarrow{X} |000\rangle|000\rangle|001\rangle \xrightarrow{X} |000\rangle|010\rangle|001\rangle \xrightarrow{\text{ASWAP}} \frac{1}{\sqrt{2}}|000\rangle\big(|010\rangle|001\rangle + |000\rangle|011\rangle\big) \\
&\xrightarrow{\text{CSWAPs}} \frac{1}{\sqrt{2}}|000\rangle\big(|010\rangle|001\rangle + |001\rangle|010\rangle\big) \xrightarrow{X} \frac{1}{\sqrt{2}}|100\rangle\big(|010\rangle|001\rangle + |001\rangle|010\rangle\big) \\
&\xrightarrow{\text{ASWAP}} \frac{1}{\sqrt{6}}|100\rangle|010\rangle|001\rangle + \frac{1}{\sqrt{3}}|000\rangle|110\rangle|001\rangle + \frac{1}{\sqrt{6}}|100\rangle|001\rangle|010\rangle + \frac{1}{\sqrt{3}}|000\rangle|101\rangle|010\rangle \\
&\xrightarrow{\text{CSWAPs}} \frac{1}{\sqrt{6}}|100\rangle|010\rangle|001\rangle + \frac{1}{\sqrt{3}}|010\rangle|100\rangle|001\rangle + \frac{1}{\sqrt{6}}|100\rangle|001\rangle|010\rangle + \frac{1}{\sqrt{3}}|001\rangle|100\rangle|010\rangle \\
&\xrightarrow{\text{ASWAP}} \frac{1}{\sqrt{6}}|100\rangle|010\rangle|001\rangle + \frac{1}{\sqrt{6}}|010\rangle|100\rangle|001\rangle + \frac{1}{\sqrt{6}}|010\rangle|000\rangle|101\rangle + \frac{1}{\sqrt{6}}|100\rangle|001\rangle|010\rangle \\
&\quad + \frac{1}{\sqrt{6}}|001\rangle|100\rangle|010\rangle + \frac{1}{\sqrt{6}}|001\rangle|000\rangle|110\rangle \\
&\xrightarrow{\text{CSWAPs}} \frac{1}{\sqrt{6}}|100\rangle|010\rangle|001\rangle + \frac{1}{\sqrt{6}}|010\rangle|100\rangle|001\rangle + \frac{1}{\sqrt{6}}|010\rangle|001\rangle|100\rangle + \frac{1}{\sqrt{6}}|100\rangle|001\rangle|010\rangle \\
&\quad + \frac{1}{\sqrt{6}}|001\rangle|100\rangle|010\rangle + \frac{1}{\sqrt{6}}|001\rangle|010\rangle|100\rangle \tag{8}
\end{aligned}$$

The explicit circuit for $M = 3$ is shown in Fig. 1, where the gates inside the red dashed rectangles are excluded at this point. Furthermore, gates acting on distinct qubits have been pushed to the left, thereby reducing the circuit depth. This produces a different sequence of intermediate states than above, but the final state is the same.

A slight modification of the above procedure allows one to simultaneously apply the phases A_P to the appropriate terms in the superposition. After each partial subregister swap, one applies a controlled phase gate with angle $\Theta(k_i, k_j) + \pi$, where i, j are the integer val-

ues that have been swapped, and where the additional phase π implements the signature of the permutation (red dashed rectangles in Fig. 1). Suppose, as in Eq. (7), that the value $(k+1)$ is swapped to the right. Then the $(k+1)$th qubit of the right subregister can serve as the target qubit in the required controlled phase. Since the left subregister can store any value $m < k+1$, a separate controlled phase is used for each possibility, where the control bits are given by the values m. This leads to a total of $M^3/3 - M^2/2 + M/6$ controlled phase gates for this part of the algorithm.
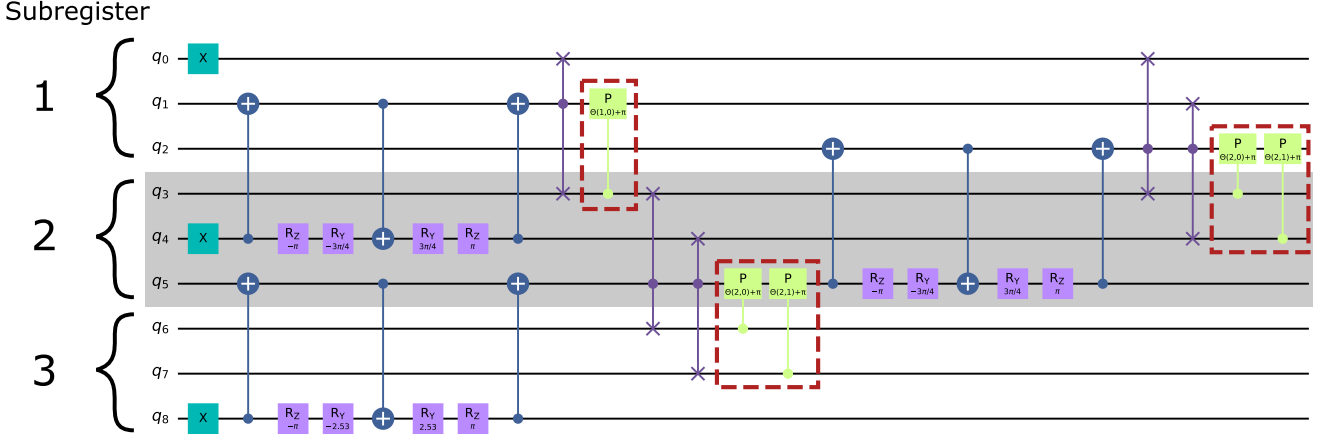
FIG. 1. Circuit to prepare the permutation label ancilla state for $M = 3$. When the gates inside the red dashed rectangles are excluded, the circuit realizes the equally-weighted superposition encoding the permutations of three objects. With these gates included, the circuit applies the additional permutation-dependent phases A_P. The three subregisters are indicated by shading, and the initial state of each qubit is $|0\rangle$. The alternating CNOT gates interleaved with R_y and R_z rotations realize the ASWAP gates.

By the end of the construction, the phases A_P have been applied to the corresponding permutation label state, having been successively built-up from the elementary transpositions of which they are composed. The explicit circuit for $M = 3$ is displayed in Fig. 1, where now the gates in the red dashed rectangles are included, in order to produce the phases A_P. At this point in the construction, the total state of the physical system and the permutation label is

$$\left(\frac{1}{\sqrt{M!}} \sum_P A_P \otimes_{j=M}^{1} |Pj\rangle_p \right) |D_{L,M}\rangle_s , \qquad (9)$$

where $|\cdots\rangle_p$ is a state of the permutation label qubits and $|\cdots\rangle_s$ is a state of the system qubits.

In the next step, one applies the position-dependent phase factors $e^{ik_{Pj}x_j}$ to the relevant basis states on the system qubits (step 3 of Algorithm 1). To do this, we introduce an efficient method that acts on all the $\binom{L}{M}$ basis states in $|D_{L,M}\rangle$ simultaneously, yielding an enormous speedup over classical approaches. The technique, which we call the "faucet" method, is based on the observation that the positions x_j take integer values $x_j = 0, \ldots, L-1$, so that the total phase $e^{ik_{Pj}x_j}$ can be produced by x_j repetitions of the phase $e^{ik_{Pj}}$.

For this part of the algorithm, we use the M additional ancilla qubits comprising the faucet register. Each of these new qubits is initialized to $|1\rangle$. In the outer loop of the faucet subroutine, one traverses the register of the system qubits site by site from $x = 0$ to $x = L - 1$. At each site, if it is occupied by a down spin (i.e. the bit is 1), one turns off the next faucet ancilla qubit , $|1\rangle \rightarrow |0\rangle$. This is achieved through a sequence of multi-controlled X gates, which are controlled on the previous ancilla being in the state $|0\rangle$ and the next one being in the state $|1\rangle$, along with the additional control that the current system site is a $|1\rangle$. Since the meaning of the "next faucet ancilla"

at a given site depends on the bitstring, one must generically apply a multi-controlled X gate for each ancilla at every step[60].

Next, the phases $e^{ik_{Pj}}$ ($j = 1, \ldots, M$) are applied to the system qubits, each being controlled on the state of one of the faucet ancillas. For the jth ancilla, this gate is also controlled on the state of the permutation label subregister j (since the value of k_{Pj} is permutation-dependent). By the end of the system bitstring, all the faucet register qubits are in state $|0\rangle$. Thus, the subroutine can be compared to a set of M running faucets (that correspond to applying the phases $e^{ik_{Pj}}$), which are turned off at the appropriate times (upon encountering a '1' in the traversal of the system register). This analogy is illustrated in Fig. 2. In total, this step requires M^2L doubly-controlled phase gates and M^2L $(M-1)$-controlled X gates. In our numerical implementation of the method, we introduce additional work qubits to facilitate the construction of these gates using chains of Toffolis [61].

After the relevant phases have been applied, it remains to disentangle the system from the permutation label (the entanglement having been generated during the faucet method, since the phases there are permutation dependent). This is accomplished by applying the inverse of the circuit that generates the permutation label superposition, without the additional controlled-phase gates that were used to produce the A_P phases. The fact that the phases A_P and $e^{ik_{Pj}x_j}$ have been applied to the initial Dicke state implies that the permutation label reversal will not completely disentangle the system qubits from the permutation label register. However, it turns out that the $|00\ldots0\rangle$ component of the permutation state corresponds precisely with the occurrence of the target Bethe ansatz state on the system qubits. That is, the full
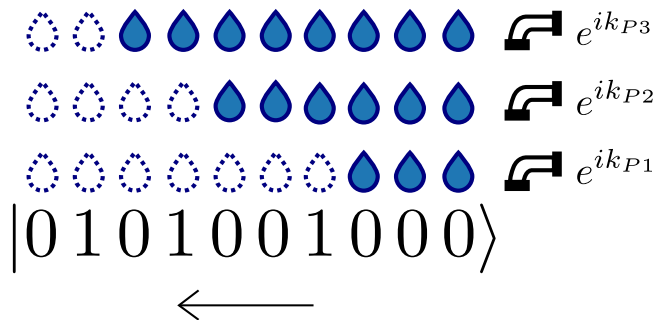
FIG. 2. Schematic diagram illustrating the idea of the faucet method. Phases $e^{ik_{Pj}}$ are applied for each system qubit while traversing the bitstring. When a '1' is encountered, the next faucet in the list is turned off, so that no more phases with the given k_{Pj} value are applied.

state vector takes the form

$$|\psi\rangle = \alpha|00\ldots0\rangle_p|\psi_B\rangle_s + \beta|\phi_j\rangle_p|\psi_j\rangle_s, \qquad (10)$$

where $|\psi_B\rangle_s$ is the normalized target Bethe ansatz state on the system qubits, $|\phi_j\rangle_p$ is a junk state on the permutation label qubits with $\langle00\ldots0|\phi_j\rangle_p = 0$, and $|\psi_j\rangle_s$ is a junk state on the system qubits (we note as an empirical finding that $\langle\psi_B|\psi_j\rangle_s = 0$ as well). Thus, by measuring the permutation label qubits, the target Bethe ansatz state is successfully prepared on the outcome $|00\ldots0\rangle$. This result is essentially that obtained from LCU methods [53], though here we have the additional construction of A_P during the label preparation step, which is not present in the standard LCU. To illustrate the full algorithm, the complete circuit to construct a BA eigenstate with $L = 4$, $M = 2$ is given in Appendix A. As discussed in our numerical simulations below, the success probability $|\alpha|^2$ depends on the system parameters, and also varies between different eigenstates. Thus, in general one must repeat the procedure multiple times to obtain the correct state, which can then be used to calculate physical quantities or in other applications, as we discuss in Section VI. We also show in Section IV that amplitude amplification can be used to boost the success rate, thereby reducing the overall resource requirements.

III. NUMERICAL SIMULATIONS

To calculate the momenta $\{k_i\}$ defining the Bethe eigenstates, we have solved the Bethe equations iteratively using the approach presented in Ref. [49]. We then performed numerical simulations of Algorithm 1 using the IBM Qiskit library's state vector simulator [62]. These calculations verify the correctness of our algorithm, and reveal its success probabilities for the sufficiently small systems that can be studied on a classical computer. However, we can also explicitly construct the circuits that would need to be run for much larger instances, far beyond what is classically tractable. The

corresponding circuit depths and gate counts in these cases indicate that our algorithm is feasible for near-term error-corrected quantum computers. We stress that our analysis does not rely on asymptotic resource scaling arguments, but rather provides exact gate counts, since the corresponding circuits are precisely known. Although we have not compiled our algorithm down to an error-correcting code such as the surface code, we estimate the required number of T gates below.

In Fig. 3(a) we present the numerically-calculated success probability of the algorithm for preparing selected eigenstates when $L = 2M$ and $M = 2, 3, 4$, with $J_{xy} = 1$, $J_z = -1/2$. The interaction strength in this case corresponds to the critical regime of the ferromagnetic model. We note the eigenstates included in Fig. 3 are not meant to comprise the complete set of real-valued solutions, but rather are simply those for which we obtained numerical solutions of the Bethe equations by using the algorithm presented in Ref. [49]. Two general trends are apparent: a significant suppression of the success rate with increasing M, and a more moderate suppression as a function of the eigenstate energy, within each set of system parameters L, M. The worst-case probabilities are roughly consistent with $1/M!$, although we have only limited values of M to support this (larger M being outside of our computational resources for classical simulation). This value is further supported by Fig. 3(b), which shows the success probabilities for various eigenstates when $M = 3$ and for different values of L, with the clear trend that increasing L tends to flatten the success probability across the spectrum. Although the low energy states enjoy less of an advantage over the higher energy ones in this case, the lowest probabilities are still around $1/M!$ on average. However, the behavior of the success rate changes significantly depending on the value of J_z. These effects are considered Appendix B.

These results suggest one can go to very large system sizes L, while still preserving a reasonably large success probability, if M is sufficiently small. More specifically, it appears feasible to access values of L and M that would not be classically simulable (even by approximate methods), while maintaining relatively modest resource requirements for the algorithm. This is demonstrated in Fig. 4, which provides circuit depths [Fig. 4(a)], Toffoli gate counts [Fig. 4(b)], controlled phase gate counts [Fig. 4(c)], and the number of qubits required [Fig. 4(d)] for the Bethe state preparation algorithm. The linear scaling of all these metrics in L is immediately apparent. But apart from the asymptotic scaling behavior, the absolute values of the circuit depths and gate counts are seen to be very low, on the order of 10^3–10^4, even for large systems of $L \sim 100$ sites. Furthermore, the total number of qubits required ($\sim 10^2$) is also quite reasonable for small error-corrected devices. In Fig. 5 we show the total gate and measurement counts for the case $M = 5$ as L is varied. This indicates that the controlled-phase, R_y, and Toffoli gates are the most prevalent non-Clifford operations in the algorithm. To estimate the fault-tolerant re-
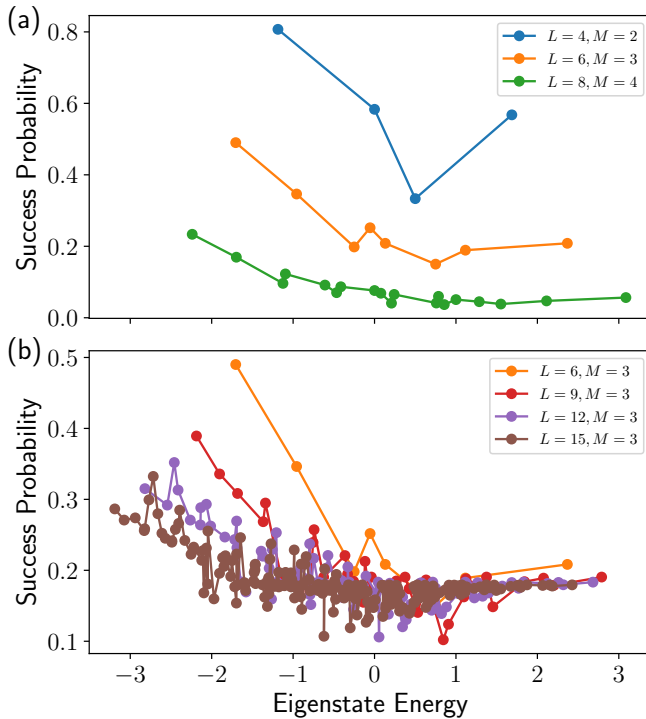
FIG. 3. (a) Success probabilities for preparation of selected eigenstates as a function of their energies, when $L = 2M$ and $M = 2, 3, 4$. (b) Success probabilities of selected eigenstates for $M = 3$ and varying L. Hamiltonian parameter values are $J_{xy} = 1$ and $J_z = -1/2$.

sources needed, we therefore convert the counts for these gates into the corresponding numbers of T gates. Following Ref. [63], we assume a worst-case scenario for the number of T gates needed to realize an arbitrary z rotation to be given by $4 \log_2(1/\epsilon) + 11$, where ϵ is the rotation synthesis error [64]. Similarly, arbitrary y rotations can be performed by conjugation with Clifford operations. As in Ref. [23], we replace each Toffoli gate with two T gates [65]. For $L = 100$, $M = 5$ this leads to a T count of $\sim 6.2 \times 10^5$ for a single run of the state preparation algorithm, with $\epsilon = 10^{-10}$. Assuming a worst-case success probability of $1/M!$, approximately 120 attempts would need to be performed on average to correctly generate the target eigenstate. This yields $\sim 7.4 \times 10^7$ T gates overall, which is comparable to the estimates for simulating the Hubbard model using the methods of Ref. [23]. We note that the estimates in that work involve optimizing an error budget between multiple sources (Trotterization, phase estimation, and rotation synthesis), and do not appear to include the cost of preparing a good initial state for the phase estimation routine. Furthermore, the above estimate for our algorithm assumes the seemingly worst-case scenario in the number of repetitions ($\sim M!$), whereas the results in Fig. 3 suggest that lower energy states require fewer repetitions in general. To reduce the number of repetitions required for our algorithm, we im-

plement amplitude amplification in the following section.

IV. AMPLITUDE AMPLIFICATION

Amplitude amplification, a generalization of the well-known Grover search algorithm, is a quantum subroutine which can boost the probability of a desired measurement outcome, leading in general to a square root improvement in the number of repetitions required for the success of a probabilistic algorithm [35]. For our problem, we use $\mathcal{B}$ to denote Algorithm 1 with the measurement step removed. Amplitude amplification defines an operator

$$\mathcal{Q} = -\mathcal{B} S_0 \mathcal{B}^{-1} S_B, \qquad (11)$$

where S_B changes the relative sign of the "good" states in the Hilbert space, while S_0 changes the relative sign of the vacuum state $|00 \dots 0\rangle$. In the present case, the good states are the components of the Bethe ansatz state, which correspond to $|00 \dots 0\rangle_p$ on the permutation label qubits. S_B can therefore be implemented using a OR circuit on the permutation label, followed by Z on the work qubit that stores the result, after which the OR is uncomputed. In our numerical calculations, we use the implementation of OR in Qiskit's standard circuit library. To produce S_0 we use the same approach, with the OR circuit extended to include the system qubits (we do not implement the -1 in Eq. (11), as it is an overall phase). We present numerical results for amplitude amplification in Fig. 6. This confirms the clear enhancement of the algorithm success probability using this method. Although only one round of amplification has been applied here, the protocol can be repeated in the standard way to further increase the success rate. Applying this improvement to the resource estimate of the previous section, the $M = 5$ worst-case eigenstates should require on average $\sqrt{120} \approx 11$ repetitions of the algorithm $\mathcal{B}$ to achieve success, leading to an overall T count of $\sim 4.1 \times 10^6$ (neglecting the costs of S_B and S_0).

We have also implemented a different version of amplitude amplification, which is a modified form of the oblivious amplitude amplification of Ref. [53]. Unfortunately, this method leads to a reduced fidelity of the actually prepared state with the exact target state, though in some cases the overlap remains quite high (> 0.99). We attribute this reduced fidelity to the non-unitarity of summing exponentials with unit modulus, since $|e^{ia} + e^{ib}| \neq 1$ in general. We note, however, that nearly deterministic success of the oblivious amplitude amplification procedure was obtained for the application of Ref. [53] (simulation of Hamiltonian dynamics with Taylor series expansions). For this reason, it is less clear how the approach will fare for the Bethe state preparation problem at larger values of M. In addition, further modification to the method of Ref. [53] may ameliorate some of the difficulties with applying it to Bethe state preparation in its present form.
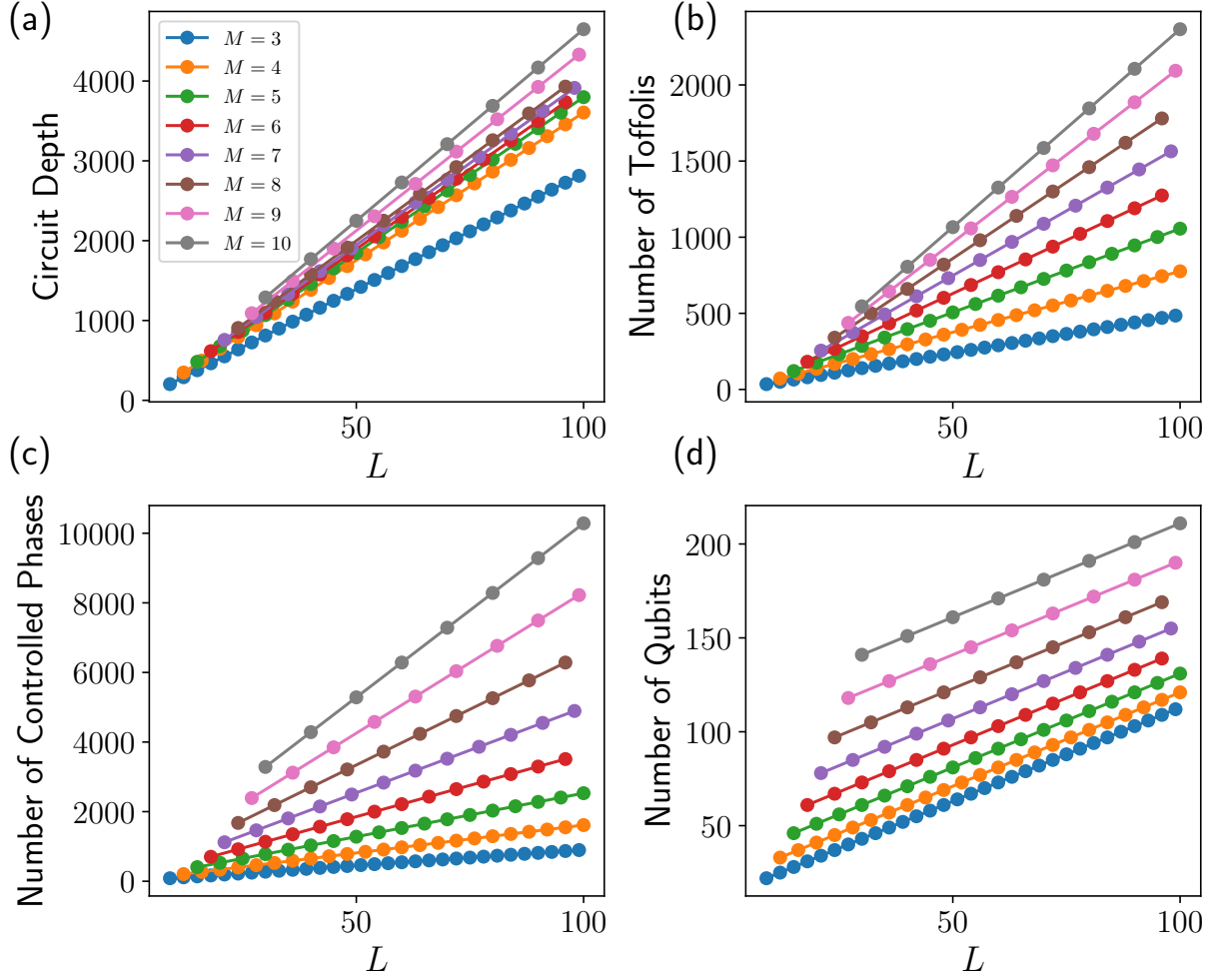
FIG. 4. Bethe state preparation (a) circuit depth, (b) number of Toffoli gates, (c) number of controlled phase gates, and (d) number of qubits versus system size L, for different numbers of down spins M.
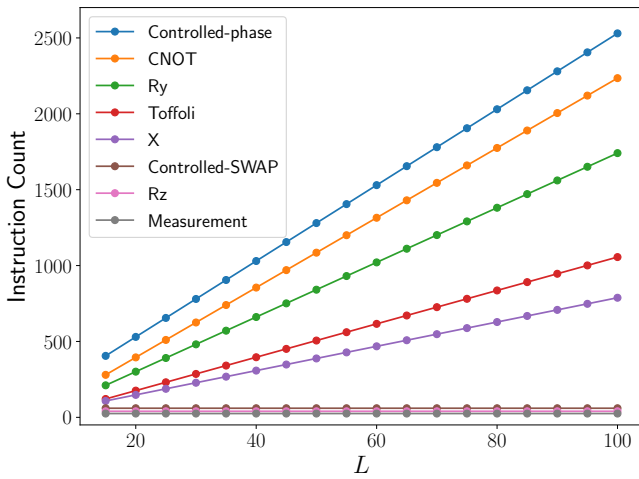

FIG. 5. Bethe state preparation gate and measurement counts for $M = 5$ as a function of L.

V. COMPARISON WITH ALTERNATIVE ALGORITHMS

To highlight the advantages of Algorithm 1, we compare it with conceptually simpler, but much less efficient, methods of preparing Bethe ansatz states on a quantum computer. First, one could imagine applying controlled phase gates directly to each term in the Dicke state superposition to generate the desired eigenstate. This approach still requires permutation label ancillas to generate the linear combination of phases needed in Eq. (2). However, it has the seeming advantage of allowing one to combine the phases A_P and $e^{ik_{P_j}x_j}$ into a single controlled-phase rotation, whereas the former term required order M^3 and the latter one order M^2 controlled-phases to implement using Algorithm 1. Nevertheless, it is clear that this benefit is vastly outweighed by the large number of terms in the superposition that must be separately addressed, $M!\binom{L}{M}$. For the case $L = 100$, $M = 5$ this amounts to $\sim 9.0 \times 10^9$ controlled phases, compared
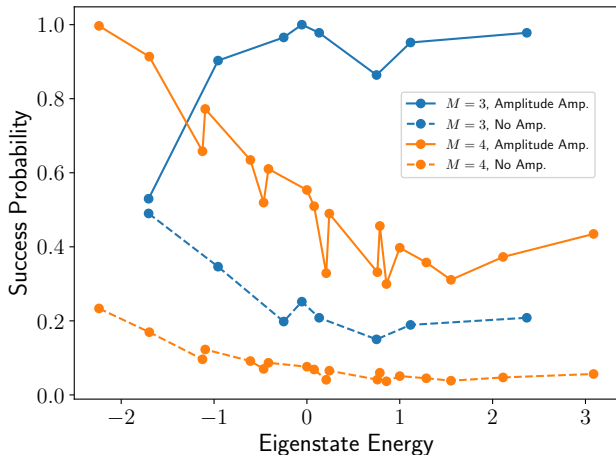
FIG. 6. Success probability for Bethe state preparation with (solid lines) and without (dashed lines) amplitude amplification. In the former case, a single round of amplification is applied. System parameters are $J_{xy} = 1$, $J_z = -1/2$, $L = 2M$.

FIG. 7. Comparison between Algorithm 1 and the alternative method of the number of (a) Toffoli and (b) controlled-phase gates used, for $M = 5$.

to the 2530 of Algorithm 1.

A more promising approach is to use the "faucet" method of Algorithm 1 to handle the $e^{ik_{P_j}x_j}$ phases while still applying the full A_P in a single multi-controlled-phase gate, rather than decomposing it into its elementary transpositions. This replaces the $M!\binom{L}{M}$ dependence above with $M!LM$. While the scaling is still inferior to that of Algorithm 1 for large M, it is conceivable that for small M this alternative method may be competitive. In particular, one can replace the complicated permutation label of Algorithm 1, which required M^2 qubits to construct A_P in terms of individual transpositions, with a compressed label that simply assigns a number to each permutation. This approach uses significantly fewer qubits, at the expense of requiring more controls for the relevant phase gates. Since the permutation label construction still needs to be reversed to disentangle the system from the ancillas, it is important that it can still be executed in a unitary fashion. This in turn requires an efficient method for generating an equal superposition of $M!$ states. We implement such states using the prime factorization $M! = 2^{n_2}3^{n_3}\cdots$. We then construct the permutation label as the tensor product of the binary representation of 2^{n_2} (using n_2 qubits) and W_n states for the odd prime factors. The equal superposition for the binary part of the label is easily generated by applying H gates to the relevant qubits, while various efficient algorithms exist for W_n state preparation [66–68]. We implemented this algorithm numerically and verified that it successfully prepares Bethe ansatz eigenstates. Since the fundamental approach for creating the linear combination of phases is the same between this method and Algorithm 1, their success probabilities are equal. Unfortunately, explicit construction of the circuits for the alternative method indicates that the resource require-
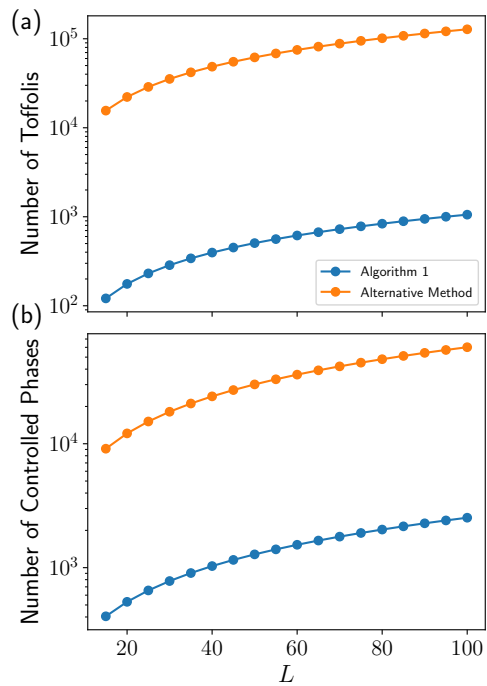
ments are significantly higher, even for small M. This is shown in Fig. 7, which reveals that the number of controlled phase and Toffoli gates required for the alternative method vastly exceeds those of Algorithm 1, even for $M = 5$.

VI. DISCUSSION

To achieve quantum advantage for a physically relevant problem, it should be the case that no classical method could deliver results of a comparable accuracy. Since the present quantum algorithm exactly prepares eigenstates of the XXZ chain, it is conceptually most similar to exact diagonalization of finite-size systems. In a recent study, a matrix-free approach was used to investigate Heisenberg spin chains up to length $L = 26$ [69]. This work had vastly reduced the memory requirements compared to conventional methods, though the scaling remained exponential with system size. Specifically the $S^z = 0$ subspace ($M = 13$) was considered, for which the dimension is $\sim 10^7$. In contrast, the $L = 100$, $M = 5$ subspace is roughly seven times larger (dimension $\sim 7.5 \times 10^7$). Although state vectors of this size can still be stored in memory, we note that the computation time is also exponential in the system size, ultimately limiting the practicality of exact diagonalization.

In addition to numerically exact calculations, approximate tensor network methods have been highly success-

ful for studying one-dimensional quantum many-body systems with the matrix product state (MPS) ansatz [70, 71]. However, these approaches are best-suited for states with a relatively low amount of entanglement, such as gapped ground states obeying an area law for the entanglement entropy. This makes simulation of long-time dynamics challenging, due to the growth of entanglement from, for instance, an initial product state. Our Bethe ansatz algorithm can prepare eigenstates throughout the spectrum at the same computational cost, including highly excited states that obey volume law entanglement scaling. This yields an advantage over MPS methods for large systems when targeting these strongly entangled states. An explicit link between the Bethe ansatz and MPS was developed in Refs. [72, 73], which used the algebraic Bethe ansatz to produce exact tensor network representations for generic eigenstates. These networks have a PEPS-like structure (but with fewer physical indices), which underscores the computational intractability of such states for large systems.

In addition to computing arbitrary-range and higher-order correlation functions that are inaccessible with traditional Bethe ansatz methods, our algorithm has a number of other applications. Simulation of the real-time dynamics of many-body systems is widely recognized as a task allowing for quantum advantage. Such simulations often take the form of quench experiments, for which the system is initialized in an easy-to-prepare product state, then allowed to evolve under the influence of an interacting many-body Hamiltonian. Our algorithm would enable interesting variations on this approach, for instance by initializing the system in an eigenstate of a given value of the interaction strength, then subsequently evolving it with a different value. The evolution here can be performed using any of the known algorithms for quantum simulation, whether by Trotterization [23, 74, 75], Taylor expansions [53, 76], or other approaches [52, 77].

In a different direction, one could use our algorithm as a starting point to explore integrability-breaking perturbations. Thus, we consider a Hamiltonian of the form $\mathcal{H} = \mathcal{H}_0 + \mathcal{H}_p$, where $\mathcal{H}_0$ is solvable by the Bethe ansatz and $\mathcal{H}_p$ includes perturbations that break the integrability of the total Hamiltonian $\mathcal{H}$ (such as disorder in the coupling strengths). In this case, the Bethe state prepration algorithm is used to prepare an eigenstate of $\mathcal{H}_0$, which then serves as an initial state for quantum annealing or phase estimation on $\mathcal{H}$. For sufficiently weak perturbations, the overlap of this state with the corresponding exact eigenstate of $\mathcal{H}$ should be much greater than that of a mean-field or non-interacting trial state.

Although we have focused on deploying our algorithm on small error-corrected quantum computers, one may also consider implementing it on present-day or near-term NISQ devices. Many of the controlled-phase gates in our algorithm involve repetitions of the same basic rotation angles, of which there are only $(M^2 + M)/2$ distinct values. This suggests replacing the exact values with variational parameters, similarly to Ref. [41]. The resulting variational form can then be optimized under the cost function $|E - E_B|$, where E is the energy calculated on the quantum computer and E_B is the exact value, known analytically from the Bethe ansatz solution. We note that the present optimization problem should be significantly easier than that of a standard VQE, since the ideal values of the rotation angles can serve as a good initial guess. Updates to the parameters then serve to directly mitigate systematic errors due to over- or under-rotation in the controlled-phase gates.

VII. CONCLUSIONS

We have presented a quantum algorithm for the efficient preparation of Bethe ansatz eigenstates of the XXZ model. To our knowledge, this is the first quantum algorithm for the direct preparation of eigenstates of an interacting many-body problem. The circuit depth and gate counts of the algorithm scale linearly in the system size, for a fixed number of down spins. Our algorithm is feasible to perform on small error-corrected devices of order 100 qubits, provided that the number of down spins is small. The usefulness of the approach can be extended through amplitude amplification. In particular, quantum advantage over classical computational methods appears to be achievable, with resource estimates that are comparable to the most efficient known quantum simulation algorithms. Our work suggests directions for future research, including the modification of the algorithm for the case of complex-valued $\{k_i\}$, and its generalization to other Bethe ansatz-solvable models, such as the one-dimensional Hubbard model.

Appendix A: Full circuit for $L = 4$, $M = 2$

In Fig. 8 we present the full quantum circuit to prepare a Bethe ansatz eigenstate with $L = 4$ and $M = 2$. We note that the work qubit is not needed when $M = 2$ case, but we have included it here as a reminder that it is used to implement multi-controlled gates for $M > 2$.
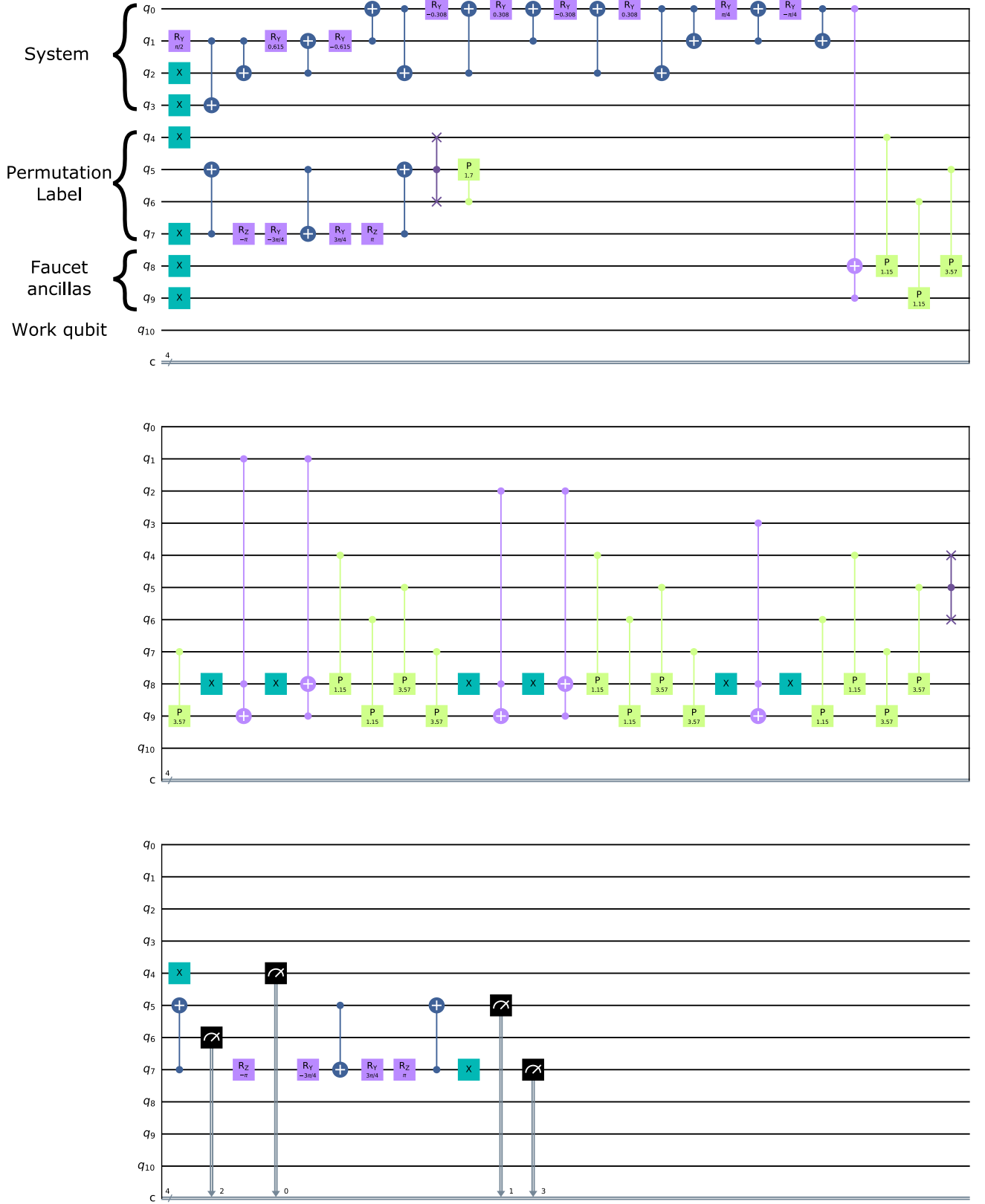
FIG. 8. Quantum circuit to prepare the Bethe ansatz eigenstate with $L = 4$, $M = 2$, $k_1 = 1.14676529$, $k_2 = 3.56562369$.
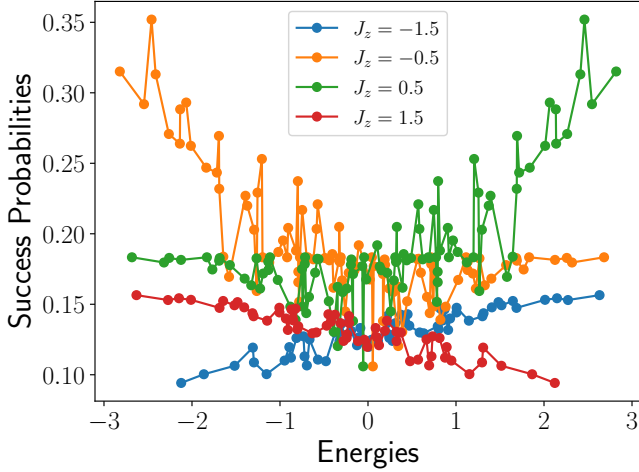
FIG. 9. Success probabilities for preparation of selected eigenstates as a function of their energies, for $L = 12$, $M = 3$, $J_{xy} = 1$, with varying interaction strengths J_z.

Appendix B: Success Probability for different J_Z

The success rate of the Bethe ansatz state preparation changes qualitatively with both the sign of J_z and whether it lies in the critical ($|J_z| < 1$) or non-critical ($|J_z| \geq 1$) regimes. These effects are illustrated in Fig. 9. Whereas the ferromagnetic model in the critical regime has higher success probabilities for lower energy states, the antiferromagnetic case shows the opposite behavior. In fact, the success probabilities for different states are exactly mirrored across the $E = 0$ axis. This is especially interesting since the corresponding eigenstates at $E = -E_i$ for the FM case and $E = E_i$ for the AFM one are different in general. It is unclear at present why these distinct states should have the same success probability. When J_z is outside the critical regime, we find that the success probability can be lower than the $1/M!$ value (≈ 0.17 for $M = 3$) that appears to bound the critical case.

[1] P. Shor, Algorithms for quantum computation: discrete logarithms and factoring (IEEE Comput. Soc. Press, 1994) pp. 124–134.

[2] A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head-Gordon, Simulated Quantum Computation of Molecular Energies, Science **309**, 1704 (2005).

[3] S. McArdle, S. Endo, A. Aspuru-Guzik, S. C. Benjamin, and X. Yuan, Quantum computational chemistry, Reviews of Modern Physics **92**, 015003 (2020).

[4] D. S. Abrams and S. Lloyd, Simulation of Many-Body Fermi Systems on a Universal Quantum Computer, Physical Review Letters **79**, 2586 (1997).

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

[6] L. K. Grover, Quantum Mechanics Helps in Searching for a Needle in a Haystack, Physical Review Letters **79**, 325 (1997).

[7] E. Farhi, J. Goldstone, and S. Gutmann, A Quantum Approximate Optimization Algorithm (2014), arXiv:1411.4028.

[8] J. Preskill, Quantum Computing in the NISQ era and beyond, Quantum **2**, 79 (2018).

[9] I. H. Deutsch, Harnessing the Power of the Second Quantum Revolution, PRX Quantum **1**, 020101 (2020).

[10] F. Arute *et al.*, Quantum supremacy using a programmable superconducting processor, Nature **574**, 505 (2019).

[11] H.-S. Zhong, H. Wang, Y.-H. Deng, M.-C. Chen, L.-C. Peng, Y.-H. Luo, J. Qin, D. Wu, X. Ding, Y. Hu, P. Hu, X.-Y. Yang, W.-J. Zhang, H. Li, Y. Li, X. Jiang, L. Gan, G. Yang, L. You, Z. Wang, L. Li, N.-L. Liu, C.-Y. Lu, and J.-W. Pan, Quantum computational advantage using photons, Science **370**, 1460 (2020).

[12] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, A variational eigenvalue solver on a photonic quantum processor, Nature Communications **5**, 1 (2014).

[13] D. Wecker, M. B. Hastings, and M. Troyer, Progress towards practical quantum variational algorithms, Physical Review A **92**, 042303 (2015).

[14] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, The theory of variational hybrid quantum-classical algorithms, New Journal of Physics **18**, 023023 (2016).

[15] H. R. Grimsley, S. E. Economou, E. Barnes, and N. J. Mayhall, An adaptive variational algorithm for exact molecular simulations on a quantum computer, Nature Communications **10**, 1 (2019).

[16] 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 **549**, 242 (2017).

[17] C. Hempel, C. Maier, J. Romero, J. McClean, T. Monz, H. Shen, P. Jurcevic, B. P. Lanyon, P. Love, R. Babbush, A. Aspuru-Guzik, R. Blatt, and C. F. Roos, Quantum Chemistry Calculations on a Trapped-Ion Quantum Simulator, Physical Review X **8**, 031022 (2018).

[18] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Barren plateaus in quantum neural network training landscapes, Nature Communications **9**, 1 (2018).

[19] E. Campbell, A. Khurana, and A. Montanaro, Applying quantum algorithms to constraint satisfaction problems, Quantum **3**, 167 (2019).

[20] Y. R. Sanders, D. W. Berry, P. C. S. Costa, L. W. Tessler, N. Wiebe, C. Gidney, H. Neven, and R. Babbush, Compilation of Fault-Tolerant Quantum Heuristics for Combinatorial Optimization, PRX Quantum **1**, 020312 (2020).

[21] R. Babbush, J. McClean, C. Gidney, S. Boixo, and H. Neven, Focus beyond quadratic speedups for error-corrected quantum advantage (2020), arXiv:2011.04149.

[22] R. Babbush, C. Gidney, D. W. Berry, N. Wiebe, J. Mc-Clean, A. Paler, A. Fowler, and H. Neven, Encoding Electronic Spectra in Quantum Circuits with Linear T Complexity, Physical Review X **8**, 041015 (2018).

[23] I. D. Kivlichan, C. Gidney, D. W. Berry, N. Wiebe, J. McClean, W. Sun, Z. Jiang, N. Rubin, A. Fowler, A. Aspuru-Guzik, H. Neven, and R. Babbush, Improved Fault-Tolerant Quantum Simulation of Condensed-Phase Correlated Electrons via Trotterization, Quantum **4**, 296 (2020).

[24] V. von Burg, G. H. Low, T. Häner, D. S. Steiger, M. Reiher, M. Roetteler, and M. Troyer, Quantum computing enhanced computational catalysis, arXiv.org (2020).

[25] J. Lee, D. Berry, C. Gidney, W. J. Huggins, J. R. McClean, N. Wiebe, and R. Babbush, Even more efficient quantum computations of chemistry through tensor hypercontraction (2020), arXiv:2011.03494.

[26] H. Bethe, Zur Theorie der Metalle i. Eigenwerte und Eigenfunktionen der linearen Atomkette, Zeitschrift für Physik **71**, 205 (1931).

[27] 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, Physical Review Letters **20**, 1445 (1968).

[28] D. C. Mattis, *The Many-Body Problem: An Encyclopedia of Exactly Solved Models in One Dimension(3rd Printing with Revisions and Corrections)* (1993).

[29] F. H. L. Essler, H. Frahm, F. Ghmann, A. Klmper, and V. E. Korepin, *The One-Dimensional Hubbard Model* (2005).

[30] F. Göhmann, M. Karbach, A. Klümper, K. K. Kozlowski, and J. Suzuki, Thermal form-factor approach to dynamical correlation functions of integrable lattice models, Journal of Statistical Mechanics: Theory and Experiment **2017**, 113106 (2017).

[31] C. Babenko, F. Ghmann, K. K. Kozlowski, and J. Suzuki, A thermal form factor series for the longitudinal two-point function of the Heisenberg-Ising chain in the antiferromagnetic massive regime, Journal of Mathematical Physics **62**, 041901 (2021).

[32] C. Babenko, F. Göhmann, K. K. Kozlowski, J. Sirker, and J. Suzuki, Exact real-time longitudinal correlation functions of the massive XXZ chain (2020), arXiv:2012.07378.

[33] R. Somma, G. Ortiz, J. E. Gubernatis, E. Knill, and R. Laflamme, Simulating physical phenomena by quantum networks, Physical Review A **65**, 042323 (2002).

[34] A. Bohrdt, Y. Wang, J. Koepsell, M. Kánasz-Nagy, E. Demler, and F. Grusdt, Dominant Fifth-Order Correlations in Doped Quantum Antiferromagnets, Physical Review Letters **126**, 026401 (2021).

[35] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, Quantum Amplitude Amplification and Estimation, Quantum Computation and Quantum Information **305**, 53 (2002).

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

[37] P. Schmoll and R. Orús, Kitaev honeycomb tensor networks: Exact unitary circuits and applications, Physical Review B **95**, 045112 (2017).

[38] A. Cervera-Lierta, Exact Ising model simulation on a quantum computer, Quantum **2**, 114 (2018).

[39] P.-L. Dallaire-Demers, M. Stęchły, J. F. Gonthier, N. T. Bashige, J. Romero, and Y. Cao, An application

[40] benchmark for fermionic quantum simulations (2020), arXiv:2003.01862.

[40] M. J. Cervia, A. B. Balantekin, S. N. Coppersmith, C. W. Johnson, P. J. Love, C. Poole, K. Robbins, and M. Saffman, Exactly solvable model as a testbed for quantum-enhanced dark matter detection (2020), arXiv:2011.04097.

[41] R. I. Nepomechie, Bethe ansatz on a quantum computer?, Quantum Information & Computation **32**, 255 (2021).

[42] W. W. Ho and T. H. Hsieh, Efficient variational simulation of non-trivial quantum states, SciPost Physics **6**, 029 (2019).

[43] R. Wiersema, C. Zhou, Y. d. Sereville, J. F. Carrasquilla, Y. B. Kim, and H. Yuen, Exploring Entanglement and Optimization within the Hamiltonian Variational Ansatz, PRX Quantum **1**, 020319 (2020).

[44] A. Cervera-Lierta, J. S. Kottmann, and A. Aspuru-Guzik, The Meta-Variational Quantum Eigensolver (Meta-VQE): Learning energy profiles of parameterized Hamiltonians for quantum simulation (2020), arXiv:2009.13545.

[45] B. Murta and J. Fernández-Rossier, Gutzwiller wave function on a digital quantum computer (2021), arXiv:2103.15445.

[46] R. Orbach, Linear Antiferromagnetic Chain with Anisotropic Coupling, Physical Review **112**, 309 (1958).

[47] V. E. Korepin, N. M. Bogoliubov, and A. G. Izergin, *Quantum Inverse Scattering Method and Correlation Functions* (1993).

[48] M. Takahashi, *Thermodynamics of One-Dimensional Solvable Models*, 1st ed. (Cambridge University Press, 1999).

[49] T. Giamarchi, *Quantum physics in one dimension*, The international series of monographs on physics No. 121 (Clarendon ; Oxford University Press, Oxford : New York, 2004) oCLC: ocm52784724.

[50] B. Sutherland, *Beautiful Models: 70 Years of Exactly Solved Quantum Many-Body Problems* (World Scientific, 2004).

[51] M. Gaudin and J.-S. Caux, *The Bethe Wavefunction* (2014).

[52] A. M. Childs and N. Wiebe, Hamiltonian simulation using linear combinations of unitary operations, Quantum Information & Computation **12**, 901 (2012).

[53] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, Simulating Hamiltonian Dynamics with a Truncated Taylor Series, Physical Review Letters **114**, 090502 (2015).

[54] A. Bärtschi and S. Eidenbenz, Deterministic Preparation of Dicke States, in *Fundamentals of Computation Theory* (Springer, Cham, 2019) pp. 126–139, dOI: 10.1007/978-3-030-25027-0_9.

[55] C. S. Mukherjee, S. Maitra, V. Gaurav, and D. Roy, Preparing dicke states on a quantum computer, IEEE Transactions on Quantum Engineering **1**, 3102517 (2020).

[56] A. M. Childs, D. Maslov, Y. Nam, N. J. Ross, and Y. Su, Toward the first quantum simulation with quantum speedup, Proceedings of the National Academy of Sciences **115**, 9456 (2018).

[57] K. Wan, Exponentially faster implementations of Select(H) for fermionic Hamiltonians, Quantum **5**, 380 (2021).

[58] P. K. Barkoutsos, J. F. Gonthier, I. Sokolov, N. Moll, G. Salis, A. Fuhrer, M. Ganzhorn, D. J. Egger, M. Troyer, A. Mezzacapo, S. Filipp, and I. Tavernelli, Quantum algorithms for electronic structure calculations: Particle-hole Hamiltonian and optimized wave-function expansions, Physical Review A **98**, 022322 (2018).

[59] B. T. Gard, L. Zhu, G. S. Barron, N. J. Mayhall, S. E. Economou, and E. Barnes, Efficient symmetry-preserving state preparation circuits for the variational quantum eigensolver algorithm, npj Quantum Information **6**, 10.1038/s41534-019-0240-1 (2020).

[60] One can decrease the number of gates for sites near the edges of the chain. For instance, at site 2 at most 2 faucets could have been turned off, so that the later ones do not need to be checked.

[61] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information* (Cambridge University Press, Cambridge, 2019) oCLC: 1129119093.

[62] H. Abraham *et al.*, Qiskit: An open-source framework for quantum computing (2019).

[63] M. Reiher, N. Wiebe, K. M. Svore, D. Wecker, and M. Troyer, Elucidating reaction mechanisms on quantum computers, Proceedings of the National Academy of Sciences **114**, 7555 (2017).

[64] P. Selinger, Efficient Clifford+T approximation of single-qubit operators, Quantum Information & Computation **15**, 159 (2015).

[65] C. Gidney and A. G. Fowler, Efficient magic state factories with a catalyzed $|CCZ\rangle$ to $2|T\rangle$ transformation, Quantum **3**, 135 (2019).

[66] C. Yesilyurt, S. Bugu, F. Ozaydin, A. A. Altintas, M. Tame, L. Yang, and Ş. K. Özdemir, Deterministic local doubling of W states, JOSA B **33**, 2313 (2016).

[67] D. Cruz, R. Fournier, F. Gremion, A. Jeannerot, K. Komagata, T. Tosic, J. Thiesbrummel, C. L. Chan, N. Macris, M.-A. Dupertuis, and C. Javerzac-Galy, Efficient Quantum Algorithms for GHZ and W States, and Implementation on the IBM Quantum Computer, Advanced Quantum Technologies **2**, 1900015 (2019).

[68] C. Gidney, General construction of W_n state, https://quantumcomputing.stackexchange.com/questions/4350/general-construction-of-w-n-state.

[69] R. Van Beeumen, G. D. Kahanamoku-Meyer, N. Y. Yao, and C. Yang, A Scalable Matrix-Free Iterative Eigensolver for Studying Many-Body Localization, in *HP-CAsia2020: Proceedings of the International Conference on High Performance Computing in Asia-Pacific Region* (ACM, 2020) pp. 179–187.

[70] F. Verstraete and J. I. Cirac, Matrix product states represent ground states faithfully, Physical Review B **73**, 094423 (2006).

[71] F. Verstraete, V. Murg, and J. I. Cirac, Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems, Advances in Physics 10.1080/14789940801912366 (2008).

[72] V. Murg, V. E. Korepin, and F. Verstraete, Algebraic Bethe ansatz and tensor networks, Physical Review B **86**, 045125 (2012).

[73] Y. Q. Chong, V. Murg, V. E. Korepin, and F. Verstraete, Nested algebraic Bethe ansatz for the supersymmetric $t - j$ model and tensor networks, Physical Review B **91**, 195132 (2015).

[74] J. D. Whitfield, J. Biamonte, and A. Aspuru-Guzik, Simulation of electronic structure Hamiltonians using quantum computers, Molecular Physics (2011).

[75] M. B. Hastings, D. Wecker, B. Bauer, and M. Troyer, Improving quantum algorithms for quantum chemistry, Quantum Information & Computation (2015).

[76] R. Babbush, D. W. Berry, I. D. Kivlichan, A. Y. Wei, P. J. Love, and A. Aspuru-Guzik, Exponentially more precise quantum simulation of fermions in second quantization, New Journal of Physics **18**, 033032 (2016).

[77] G. H. Low and I. L. Chuang, Hamiltonian Simulation by Qubitization, Quantum **3**, 163 (2019).

[78] P. Weinberg and M. Bukov, QuSpin: a Python package for dynamics and exact diagonalisation of quantum many body systems part I: spin chains, SciPost Physics **2**, 003 (2017).