

Variational adiabatic transport of tensor networks

Hyeongjin Kim,^{1,2,*} Matthew Fishman,² and Dries Sels^{2,3}

¹*Department of Physics, Boston University, Boston, Massachusetts 02215, USA*

²*Center for Computational Quantum Physics, Flatiron Institute, New York, New York 10010, USA*

³*Department of Physics, New York University, New York, New York 10003, USA*

(Dated: November 3, 2023)

We discuss a tensor network method for constructing the adiabatic gauge potential – the generator of adiabatic transformations – as a matrix product operator, which allows us to adiabatically transport matrix product states. Adiabatic evolution of tensor networks offers a wide range of applications, of which two are explored in this paper: improving tensor network optimization and scanning phase diagrams. By efficiently transporting eigenstates to quantum criticality and performing intermediary density matrix renormalization group (DMRG) optimizations along the way, we demonstrate that we can compute ground and low-lying excited states faster and more reliably than a standard DMRG method at or near quantum criticality. We demonstrate a simple automated step size adjustment and detection of the critical point based on the norm of the adiabatic gauge potential. Remarkably, we are able to reliably transport states through the critical point of the models we study.

I. INTRODUCTION

In the past few decades, the density matrix renormalization group (DMRG) [1] has established itself as one of most prominent numerical algorithms for simulating one-dimensional quantum lattice systems (see Refs. [2–4] for reviews). It is commonly realized as a variational tensor network method that uses a matrix product state (MPS) as an ansatz for quantum wave functions and has been extensively used to compute low energy states and dynamics of various low-dimensional quantum systems. Determining excited states provides important physical information of the system: for example, excited energy spectra can provide information on quantum phase transitions and thus help determine quantum critical points (the critical points) [5, 6] or can be used to construct the low temperature spectral function. While ground state methods such as DMRG are well established, excited state methods are still in development. Even so, there have been numerous extensions of DMRG that find excited states accurately and efficiently. One natural extension is to compute the lowest energy states of the Hamiltonian in multiple symmetry sectors, provided that they exist [2, 7]. Another, more general, method involves adding a penalty term in the Hamiltonian to penalize, already found, lower energy states [4]. There is a plethora of other extensions including but not limited to Refs. [7–16]. However, even with these efforts, computing excited states remains a computationally challenging task, especially near the critical points where the spectrum is dense.

Adiabaticity plays a fundamental role in physics with numerous applications in quantum state preparation, quantum computation, and even heat engines in thermodynamics. Specifically, the adiabatic theorem states

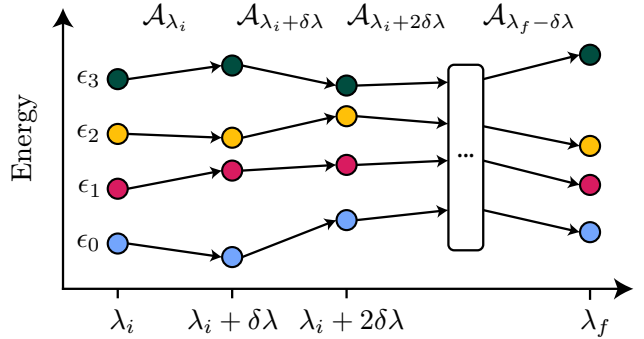


FIG. 1. Adiabatic evolution of energy eigenstates using the adiabatic gauge potential. As an illustrative example, we begin with four lowest-lying states at λ_i . Then, we compute the AGP \mathcal{A}_{λ_i} at λ_i to propagate these states to the next parameter point $\lambda_i + \delta\lambda$ and continue this procedure iteratively until we reach λ_f . In practice, the spacing $\delta\lambda$ can be automatically adjusted based on the norm of the AGP to add more steps where necessary, for example near quantum critical points, as discussed in Sec. III. Optimization algorithms like DMRG can be used to find more accurate states during the course of the adiabatic evolution.

that a quantum system, prepared in an eigenstate of a Hamiltonian, stays an eigenstate of the Hamiltonian as long as the Hamiltonian is varied slowly enough. One way to formalize this adiabatic transport is through the adiabatic gauge potential (AGP) [17–19]. In short, the AGP is the generator of adiabatic eigenstate deformations and has been used for various applications, ranging from counterdiabatic driving for optimal control [20–22] to studying quantum chaos of many-body systems [23]. Here, we demonstrate that the AGP can be efficiently and reliably represented numerically as a tensor network, specifically a matrix product operator (MPO), and introduce a simple numerical optimization method for obtaining the MPO representation of the AGP. This provides a numerically efficient and general way of performing adi-

* hkim12@bu.edu

adiabatic evolution of MPS, with straightforward generalizations to other tensor networks, by evolving with the AGP (in contrast to previous work on adiabatic evolution of tensor networks [24–28] based on parent Hamiltonian constructions) and thereby opens up numerous applications in tensor networks.

We demonstrate that AGP evolution of MPS can be utilized to significantly improve the reliability and efficiency of computing low energy excited states represented as MPSs at or near criticality, focusing in this work on a particular flavor of excited state DMRG. Our approach involves computing MPS excitations far from the critical point and adiabatically evolving these states towards the critical point, as shown schematically in Fig. 1, where calculations become significantly more difficult. We show that this method finds a set of low energy eigenstates more reliably, accurately, and efficiently at the critical point than a standard excited state DMRG method. Relatedly, we demonstrate how adiabatic evolution of tensor networks with the AGP can be used to automatically scan the entire phase diagram of quantum many-body systems by evolving low-lying energy eigenstates through different phases. We demonstrate that the norm of the AGP can be used to automatically adjust the step size of the adiabatically evolution, with a higher norm and therefore more steps needed at or near the critical point, giving a way to automatically detect different phases and phase transitions.

II. TENSOR NETWORK REPRESENTATION OF THE ADIABATIC GAUGE POTENTIAL

In this section, we give a brief introduction to the AGP with emphasis on its role as a generator of adiabatic eigenstate deformations and the minimization approach for its construction. While much of this has been discussed in earlier papers [17, 21, 29], it is provided here as context for when we derive a method for constructing the AGP as an MPO. The latter was recently also discussed in detail in Ref. [29].

A. Background on the adiabatic gauge potential

Suppose we have a Hamiltonian $\mathcal{H}(\lambda)$ with a tunable parameter λ . The AGP, denoted as \mathcal{A}_λ , is a generator of adiabatic eigenstate deformations from changing λ via

$$i\partial_\lambda |n(\lambda)\rangle = \mathcal{A}_\lambda |n(\lambda)\rangle, \quad (1)$$

where $|n(\lambda)\rangle$ is an energy eigenstate of $\mathcal{H}(\lambda)$ and, in general, the AGP depends on λ . Using this, we can adiabatically transform the eigenstates of $\mathcal{H}(\lambda)$ to those of $\mathcal{H}(\lambda + d\lambda)$ with

$$|n(\lambda + d\lambda)\rangle = e^{-i\mathcal{A}_\lambda d\lambda} |n(\lambda)\rangle, \quad (2)$$

where $d\lambda$ is an infinitesimal perturbation on λ .

While the AGP can be numerically computed exactly, it typically requires exact diagonalization, which is computationally infeasible for large systems. To combat this, variational approaches for approximately computing the AGP have been explored in earlier work [20–22, 30], which usually involve minimizing the action S_λ

$$\mathcal{A}_\lambda = \arg \min_{\mathcal{X}} S_\lambda(\mathcal{X}), \quad (3)$$

where

$$S_\lambda(\mathcal{X}) = \|G_\lambda(\mathcal{X})\|^2 = \|\partial_\lambda \mathcal{H} + i[\mathcal{X}, \mathcal{H}]\|^2, \quad (4)$$

and $\|\cdot\|^2$ is the Frobenius norm. To compute \mathcal{A}_λ , we take the gradient of S_λ

$$\begin{aligned} \nabla_{\mathcal{X}} S_\lambda(\mathcal{X}) = & 2(\mathcal{X}\mathcal{H}^2 + \mathcal{H}^2\mathcal{X} - 2\mathcal{H}\mathcal{X}\mathcal{H}) \\ & - 2i(\partial_\lambda \mathcal{H}\mathcal{H} - \mathcal{H}\partial_\lambda \mathcal{H}), \end{aligned} \quad (5)$$

and set it equal to zero. After setting $\nabla_{\mathcal{X}} S_\lambda(\mathcal{X}) = 0$ in Eq. (5), we solve for \mathcal{X} in the linear equation

$$\mathcal{X}\mathcal{H}^2 + \mathcal{H}^2\mathcal{X} - 2\mathcal{H}\mathcal{X}\mathcal{H} = i(\partial_\lambda \mathcal{H}\mathcal{H} - \mathcal{H}\partial_\lambda \mathcal{H}), \quad (6)$$

where we used the fact that $\mathcal{H}^\dagger = \mathcal{H}$. One method of solving this linear equation is to use a variational ansatz involving the commutator expansion of the form [21]

$$\mathcal{A}_\lambda^{(\ell)} = i \sum_{k=1}^{\ell} \alpha_k \underbrace{[\mathcal{H}, [\mathcal{H}, \dots [\mathcal{H}, \partial_\lambda \mathcal{H}]]]}_{2k-1}, \quad (7)$$

where the coefficients $\{\alpha_1, \alpha_2, \dots, \alpha_k\}$ can be obtained from solving Eq. (6). Notably, the exact AGP is recovered when $\ell \rightarrow \infty$. Due to the growing terms in the summation, the calculations quickly becomes inefficient for sufficiently large ℓ and the convergence can be slow. Instead, we propose a tensor network method for directly computing the approximate AGP as an MPO whose local support depends on its bond dimension D . This allows us to construct the AGP for large system sizes in a well-controlled manner.

B. Adiabatic gauge potential as a matrix product operator

To compute the approximate AGP as an MPO, we first express \mathcal{H} and $\partial_\lambda \mathcal{H}$ as MPOs. These MPOs can be efficiently constructed with low bond dimension (that is either constant or only grows polynomially with system size) for Hamiltonians with quasi-local interactions [31–33]. Then, we solve the linear equation in Eq. (6) in terms of tensor networks (see Fig. 2) to obtain \mathcal{X} that approximates \mathcal{A}_λ .

We recast the linear equation to the form $Ax = b$, where A is an MPO and x, b are MPSs acting in a doubled Hilbert space. We convert \mathcal{X} and the right side of Eq. (6) to MPSs $|\mathcal{X}\rangle$ and $|b\rangle$, respectively, and recast the

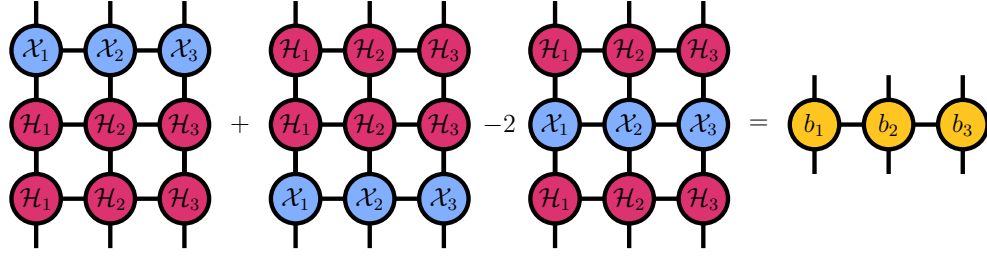


FIG. 2. Tensor network representation of Eq. (6) for solving for an approximate representation of the AGP as an MPO. For simplicity, we show an example with 3 sites, but the method generalizes to any number of sites (even an infinite number of sites). The tensors b_i represent the right hand side term of Eq. (6). This diagram represents the form of the linear equation that we can solve to obtain an approximation to the AGP parametrized by \mathcal{X}_i (given \mathcal{H}_i and b_i) using standard tensor network methods.

operators acting on \mathcal{X} on the left side of the equation as a super-operator acting on $|\mathcal{X}\rangle$:

$$(I \otimes \mathcal{H}^2 + \mathcal{H}^2 \otimes I - 2\mathcal{H} \otimes \mathcal{H}) |\mathcal{X}\rangle = |b\rangle \rightarrow A |\mathcal{X}\rangle = |b\rangle, \quad (8)$$

where A is an MPO acting on the doubled Hilbert space. We numerically solve this equation by using $\mathcal{A}_\lambda^{(1)} = i\alpha_1[\mathcal{H}, \partial_\lambda \mathcal{H}]$ from Eq. (7) as an initial guess to obtain the MPS $|\mathcal{X}\rangle \approx |\mathcal{A}_\lambda\rangle$ that approximately satisfies Eq. (8). Specifically, we use a modification of the DMRG algorithm for solving linear equations of MPS [34, 35] in order to approximately solve Eq. (8). Finally, we recast $|\mathcal{A}_\lambda\rangle$ back as an MPO. The details of this method have been summarized in Alg. 1.

Algorithm 1 Computing the AGP

Input: $\mathcal{H}(\lambda)$, $\partial \mathcal{H}(\lambda)$
 $A \leftarrow I \otimes \mathcal{H}^2 + \mathcal{H}^2 \otimes I - 2\mathcal{H} \otimes \mathcal{H}$
 $|b\rangle \leftarrow i |\partial_\lambda \mathcal{H} \mathcal{H} - \mathcal{H} \partial_\lambda \mathcal{H}\rangle$
 $|\mathcal{X}_0\rangle \leftarrow i\alpha_1 |[\mathcal{H}, \partial_\lambda \mathcal{H}]\rangle \quad \triangleright$ Use $\ell = 1$ in Eq. (7)
 $|\mathcal{A}_\lambda\rangle \leftarrow$ Solve for $|\mathcal{X}\rangle$ in $A |\mathcal{X}\rangle = |b\rangle \quad \triangleright$ Initialize with $|\mathcal{X}_0\rangle$
 $\mathcal{A}_\lambda \leftarrow |\mathcal{A}_\lambda\rangle \quad \triangleright$ Recast MPS as an MPO

Now, we numerically construct the AGP as an MPO and study its properties as we vary the bond dimension D of the AGP. To test the properties of Alg. 1, we consider the transverse field Ising model with open boundary conditions

$$\mathcal{H}_{\text{TFIM}} = \sum_{i=1}^{L-1} \sigma_i^x \sigma_{i+1}^x + g \sum_{i=1}^L \sigma_i^z, \quad (9)$$

$$\partial_g \mathcal{H}_{\text{TFIM}} = \sum_{i=1}^L \sigma_i^z, \quad (10)$$

where g is the transverse field strength and L is the system size. Notably, a phase transition occurs at $g = 1$ ($g \leq 1$ when L is finite): the eigenstates transition from a paramagnetic phase when $g > 1$ to an anti-ferromagnetic phase when $g < 1$. In Fig. 3, we construct \mathcal{A}_g as an MPO using Alg. 1 for $L = 10, 120$ at $g = 0.5, 1.0, 2.0$ with various bond dimensions $D = 5, 10, 20, 40, 80, 160$ of \mathcal{A}_g . As

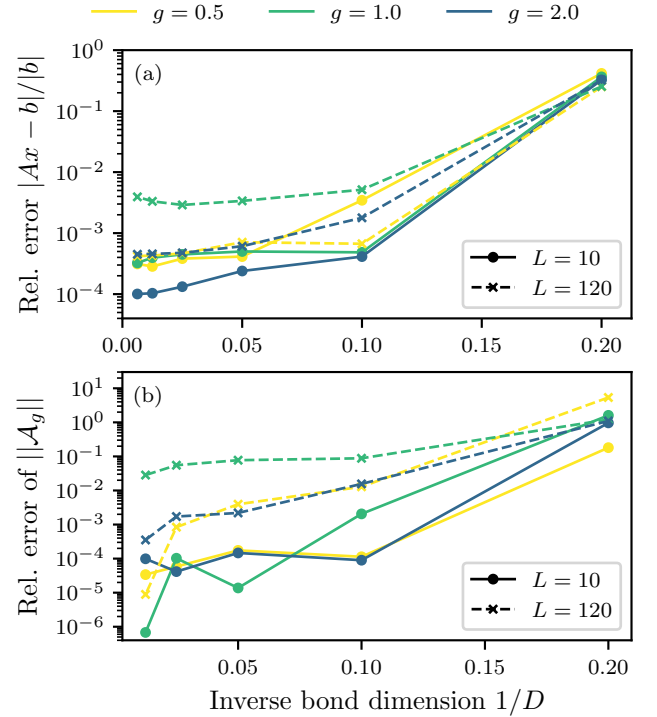


FIG. 3. Convergence properties of solving for the MPO representation of the AGP with Alg. 1. We compute \mathcal{A}_g as an MPO for $L = 10$ (shown in solid lines with dots) and $L = 120$ (shown in dashed lines with crosses). (a) shows the relative error $|Ax - b|/|b|$ determined by Eq. (8) for various bond dimensions D of \mathcal{A}_g . (b) shows the relative error of the norm of the MPO approximation of the AGP $\|\mathcal{A}_g\|$ at bond dimension D compared to the MPO approximation of $\|\mathcal{A}_g\|$ computed at $D = 160$.

shown in Fig. 3a, the relative error $|Ax - b|/|b|$ generally decreases as D increases for different realizations of g and L used. Furthermore, Fig. 3b shows similar convergence behaviors in $\|\mathcal{A}_g\|$ as D increases, even though we do not directly solve to minimize the relative error of $\|\mathcal{A}_g\|$. Note that even near the critical point, where the bond dimension of the AGP would formally diverge

in the thermodynamic limit, we can efficiently find approximate AGP with reasonable bond dimensions for this model. For example, on a personal computer, computing the AGP with bond dimension $D = 40$ at $L = 120$ took a couple of minutes.

III. ADIABATIC EVOLUTION OF TENSOR NETWORKS WITH THE AGP

In this section, we briefly describe how adiabatic evolution of MPS can be performed efficiently using an MPO representation of the AGP, and describe its application to preparing states for iterative optimizations as well as scanning phase diagrams.

Following Alg. 1, we can construct the approximate AGP as an MPO for any quantum system with quasi-local Hamiltonian. In principle, we can use this AGP to adiabatically evolve any quantum state represented as an MPS to any parameter regime of the system. For example, as shown schematically in Fig. 1, we can evolve MPS excitations from $\mathcal{H}(\lambda_i)$ to $\mathcal{H}(\lambda_f)$ by iteratively applying the AGPs on the states throughout the trajectory. Specifically, we use the time dependent variational principle method (TDVP) [36, 37] to adiabatically evolve the matrix product states, which can be formulated as a natural generalization of DMRG to performing evolution of an MPS and is well suited for evolving MPS with evolution generated by MPOs, unlike alternative MPS evolution methods like the time evolving block decimation (TEBD) [38] or time dependent DMRG [39] methods. In practice, we find that TDVP works very reliably for evolving MPS eigenstates with the AGP to perform adiabatic evolution.

Now, we present two applications of using the AGP to perform adiabatic evolution of tensor networks. Firstly, we show how it can be used to improve the calculation of low energy states with DMRG by combining AGP evolution with DMRG. We also describe how AGP evolution of MPS can be used to automatically scan phase diagrams of low dimensional quantum systems.

Applying the technique introduced in Sec. III, we combine our method for computing the MPO representation of the AGP (summarized in Alg. 1) with excited state DMRG with the goal of improving the computation time, reliability, and accuracy. Given that finding excited states (and even ground states) with DMRG can be computationally challenging around the critical point, where the states become nearly degenerate, we instead compute low energy MPS eigenstates sufficiently far away from the critical point, for example in a gapped phase where the states are less entangled and easier to find with DMRG or at a point in the phase diagram where a tensor network state can be efficiently constructed from another ansatz, such as a free fermion state [40–45], and adiabatically evolve these states towards the critical point.

To be specific, suppose that the critical point is located at λ_f in Fig. 1. We start by computing the low

energy MPS at λ_i with DMRG, where we choose λ_i to be a point in the phase diagram far from the critical point where DMRG is reliable and fast. Then, we compute \mathcal{A}_{λ_i} as an MPO using Alg. 1 and evolve the states to $\lambda_i + \delta\lambda$ using TDVP. We compute a new AGP MPO at $\lambda_i + \delta\lambda$ (perhaps using the AGP we previously found at λ_i as an initial guess for the optimization), and continue this procedure until we reach λ_f , at which point we perform excited state DMRG using the propagated MPS excitations as an ansatz to find excited states at λ_f . This serves as a controlled method for preparing good initial states for DMRG that are already close to the true low energy eigenstates, and therefore DMRG requires fewer iterations to converge and more reliably converges to states close to the true eigenstates. It is clear that the computational advantage of this method relies on the ability to (1) quickly find MPO approximations of AGPs, (2) efficiently evolve states using the MPO AGPs (for which we employ the TDVP method), and (3) maintaining accuracy of states during the evolution. As shown later in Sec. IV, we find that (1) and (2) can be easily achieved while (3) can be done by incorporating inexpensive excited state DMRG in the intermediate steps. The steps of this method combining DMRG with adiabatic evolution with the AGP are summarized in Alg. 2.

Algorithm 2 AGP evolution of MPS

Require: $\lambda_f > \lambda_i$
 $\{\epsilon_n(\lambda_i), |n(\lambda_i)\rangle\} \leftarrow \text{DMRG}$
 $\mathcal{A}_{\lambda_i} \leftarrow \text{AGP}(\mathcal{A}_{\lambda_i}^{(1)}, \mathcal{H}(\lambda_i), \partial\mathcal{H}(\lambda_i))$ ▷ Refer to Alg. 1
 $\lambda \leftarrow \lambda_i$
while $\lambda < \lambda_f$ **do**
 $\delta\lambda \leftarrow \delta\lambda \propto 1/\|\mathcal{A}_\lambda\|$
 $\{\epsilon_n(\lambda), |n(\lambda)\rangle\} \leftarrow \{\epsilon_n(\lambda + \delta\lambda), |n(\lambda + \delta\lambda)\rangle\}$ using TDVP
Optional: $\{\epsilon_n(\lambda), |n(\lambda)\rangle\} \leftarrow \text{DMRG on } \{\epsilon_n(\lambda), |n(\lambda)\rangle\}$
 $\mathcal{A}_\lambda \leftarrow \text{AGP}(\mathcal{A}_\lambda, \mathcal{H}(\lambda + \delta\lambda), \partial\mathcal{H}(\lambda + \delta\lambda))$
 $\lambda \leftarrow \lambda + \delta\lambda$
end while

Notably, the AGP calculations can be performed independently from the DMRG and TDVP computations given that the AGP evolution step sizes ($\delta\lambda$) can be determined by examining $\|\mathcal{A}_\lambda\|$: as shown in Eqs. (1) and (2), the eigenstates change the most via adiabatic evolution when $\|\mathcal{A}_\lambda\|$ is the largest. Interestingly, this usually coincides with the critical point as the norm of the AGP is equivalent to the fidelity susceptibility, which has been extensively used to probe quantum phase transitions and peaks or even diverges at the critical point [46–49]. Hence, we use smaller step sizes near the critical point by varying $\delta\lambda$ to be inversely proportional to $\|\mathcal{A}_\lambda\|$ as shown in Alg. 2 with an illustrative example shown in Fig. 4.

A modification of the method described above is to perform DMRG calculations at each intermediate value of λ 's between λ_i and λ_f and thereby compute accurate low-lying MPS excitations at each parameter point, where neither λ_i nor λ_f has to be at the critical point.

By varying $\delta\lambda$ to be inversely proportional to $\|\mathcal{A}_\lambda\|$, the algorithm can naturally focus on potential regimes of criticality and so better understand the low energy dynamics of the system. Therefore, we can efficiently scan phase diagrams between any parameter regimes, even while crossing phase transitions or critical points. Additionally, we believe this method for adiabatic evolution of tensor networks with the AGP can find many applications beyond the ones described here, some of which we discuss in the conclusion.

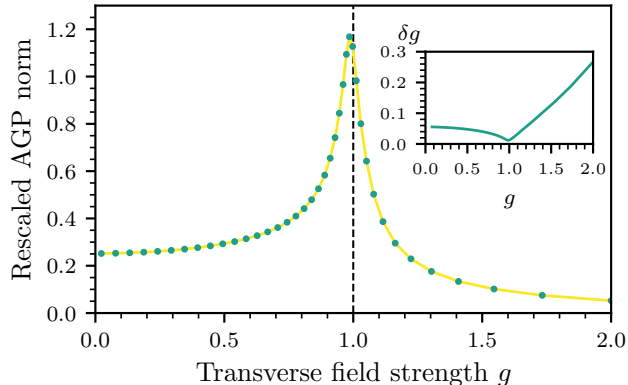


FIG. 4. Norm of the AGP for the transverse field Ising model at system size $L = 120$. We scaled the norm of the AGP ($\|\mathcal{A}_g\|$) by $1/\sqrt{L2^L}$ to remove extensiveness due to $\partial_g \mathcal{H}$. The vertical line at $g = 1$ signifies the critical point and the inset shows the spacing δg against g .

IV. APPLICATIONS OF AGP EVOLUTION OF TENSOR NETWORKS

In Sec. IV A, we demonstrate the utility of this approach for improving the reliability and performance of optimizing sets of low energy eigenstates with DMRG, focusing on the transverse field Ising model so that we can carefully benchmark our results. Then, in Sec. IV B, we use AGP evolution of tensor networks, in combination with DMRG, to scan the phase diagram of the transverse field Ising model, with or without an integrability breaking longitudinal field, by adiabatically evolving a set of approximate low energy MPS eigenstates across the phase transition.

A. Improving tensor network optimization with AGP evolution

In this section, we demonstrate the utility of adiabatically evolving tensor networks with the AGP for improving the optimization of tensor networks. Many tensor network optimization methods, like DMRG, are iterative methods. The number of iterations needed to reach convergence, and whether or not the method converges to

the global optimum, can depend on the initial state of the optimization. The convergence can be particularly sensitive to the initialization when there are other states very close to the globally optimal state, for example at or around quantum critical points, in which case the optimization can get stuck in a nearby state (i.e. a local minimum). Adiabatic evolution can provide a controlled way to prepare high quality initial states for iterative methods like DMRG. For example, if one can find high quality low energy states in a phase of the model that is relatively easy (for example, deep in a gapped phase, where the states generally have lower entanglement and the optimization is more reliable), one can then adiabatically evolve those states to a parameter regime that is more challenging and use them as initial states for an iterative optimization algorithm like DMRG.

Hence, we will focus on the problem of computing a set of low lying energy eigenstates near a critical point. For simplicity, we will focus on a particular flavor of excited state DMRG that finds higher excited states by adding weighted projectors of previously found states. One can then apply DMRG on the modified Hamiltonian of the form

$$\mathcal{H}' = \mathcal{H} + w \sum_{m < n} |m\rangle\langle m|, \quad (11)$$

where $w > 0$ is the penalty weight and the n th state is the desired excited state [4]. We emphasize that there are many other excited state DMRG techniques [7–16] and, in principle, incorporating adiabatic evolution via the AGP to any iterative ground or excited state DMRG method to prepare initial states could improve the convergence, including the computational time and/or precision.

We demonstrate the utility of AGP evolution for computing a set of low-lying excited states of the transverse field Ising model (Eq. (9)) at the critical point, $g = 1$. This model preserves σ^z parity, which separates the Hamiltonian into even and odd parity sectors. In the following, we will denote the energy eigenvalues in the even sector as ϵ^+ and in the odd sector as ϵ^- . This model is integrable and is numerically solvable by a mapping to a free fermion model via the Jordan-Wigner transformation [50, 51], allowing us to exactly compute errors in the energies of MPSs.

For our approach that combines adiabatic evolution using the AGP and DMRG optimization, we start by running DMRG to find the ten lowest-lying states in even and odd sectors, respectively, at $g = 2$, until the relative energy difference between DMRG sweeps is less than 10^{-5} . Then, we propagate those approximate eigenstates to $g = 1$ using AGP evolution and DMRG calculations with only a few sweeps after each step adiabatic evolution. To ensure these intermediate DMRG runs do not dominate the computation time, they are run until the relative energy difference between sweeps is less than 10^{-6} , which we find is enough to prevent accumulating large errors from taking large steps for the adiabatic evo-

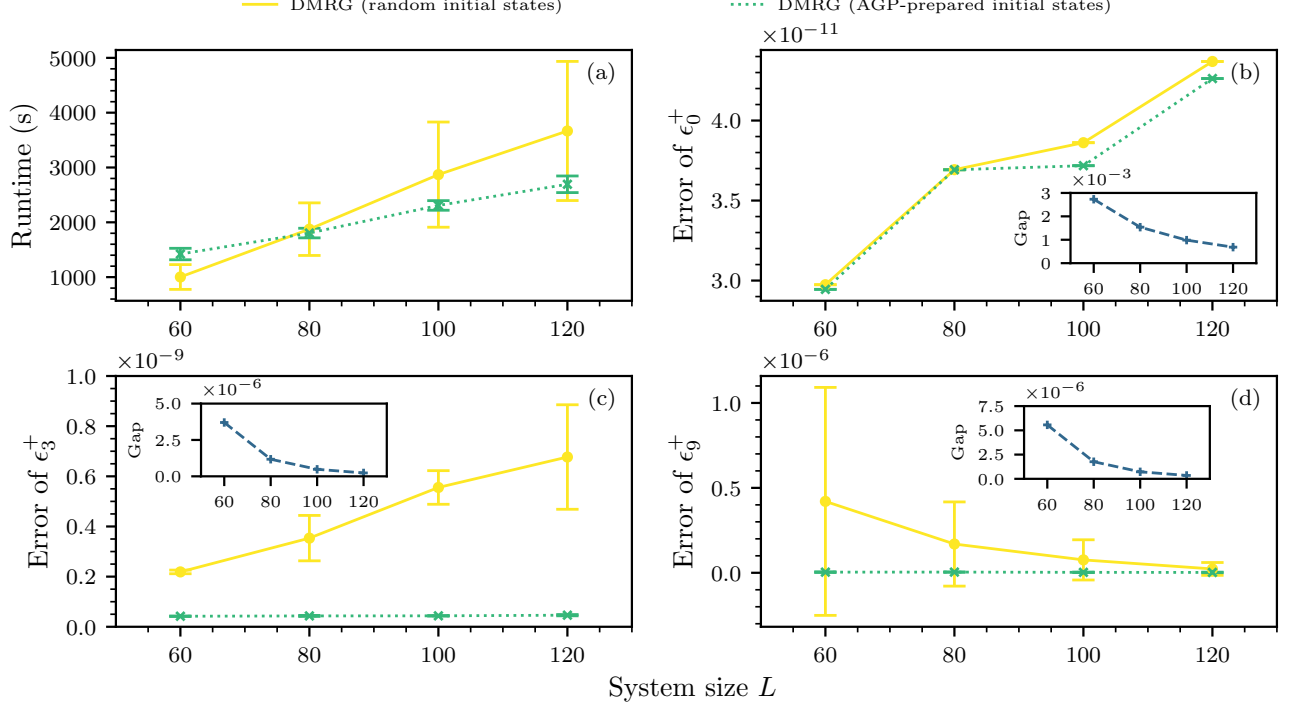


FIG. 5. Transverse field Ising model with $L = 120$ at transverse field strength $g = 1$. We compare the performance of DMRG with AGP-prepared initial states (dotted lines) against DMRG with random initial states (solid lines) at $g = 1$ for varying system sizes $L = 60, 80, 100, 120$. Both methods find the ten lowest-lying states in even and odd sectors at $g = 1$, respectively, until the relative energy difference between each sweep is less than 10^{-11} . AGP-initialized DMRG used 8 realizations of g , computed at $g = [2.0, 1.8, 1.6, 1.4, 1.2, 1.08, 1.02, 1.0]$. For the DMRG calculations, SVD truncation cutoffs between 10^{-11} and 10^{-8} are used for the MPS truncation while truncation cutoffs of 10^{-6} to 10^{-5} are used when solving for the AGP \mathcal{A}_λ . (a) shows the average runtime in seconds against system size. For AGP-initialized DMRG, this includes the time taken to obtain the AGP, perform the AGP evolutions, and execute the intermediary DMRG calculations. (b), (c), and (d) show the errors of ϵ_0^+ , ϵ_3^+ , and ϵ_9^+ , respectively. Insets in (b), (c), and (d) show the relative exact energy differences between ϵ_0^+ and ϵ_1^+ , ϵ_3^+ and ϵ_4^+ , and ϵ_8^+ and ϵ_9^+ , respectively, for all system sizes considered.

lution. Finally, we perform DMRG at $g = 1$ using the propagated states until the relative energy difference between sweeps for each eigenstate is less than 10^{-11} , which is less than the relative difference between the exact energies of the eigenstates for the system sizes considered (see insets of Fig. 5). This ensures that we have enough precision to properly find excited states at the critical point. In all, this procedure defines the AGP-initialized DMRG method.

For comparison, we run standard DMRG optimization at transverse field $g = 1$ using random initial wavefunctions. For this comparison, we used MPSs of bond dimension 10 that are generated with random circuits, which we find is a relatively good unbiased method for initializing the starting states of the optimization. Then we perform DMRG to compute the ten lowest-lying states in even and odd sectors, respectively, until the relative energy difference between sweeps for each eigenstate is less than 10^{-11} .

For DMRG with AGP-prepared initial states, we start with ten different realizations of random trial states at

$g = 2$. On the other hand, for DMRG with random initial states, we simply use twenty different realizations of random initial states at $g = 1$. Fig. 5a shows the average runtimes of DMRG using random initial states compared to DMRG with initial states prepared using AGP evolution (with their sample standard deviations represented by error bars) for system sizes $L = 60, 80, 100$, and 120. As shown, on average, AGP-initialized DMRG performs faster than the random-initialized DMRG for larger system sizes, as the runtime of the former increases slower with system size. Notably, the standard deviation of runtimes for standard DMRG drastically increases with system size, which further suggests that the AGP-initialized DMRG is a more reliable method. Next, we systematically compare the errors of the energies ϵ_0^+ , ϵ_3^+ , and ϵ_9^+ in Figs. 5b, 5c, and 5d, respectively, obtained using the two methods. As shown, on average, AGP-initialized DMRG outperforms random-initialized DMRG by achieving up to two orders of magnitude lower errors with much smaller fluctuations. As reference, the insets of Figs. 5b, 5c, and 5d show the relative

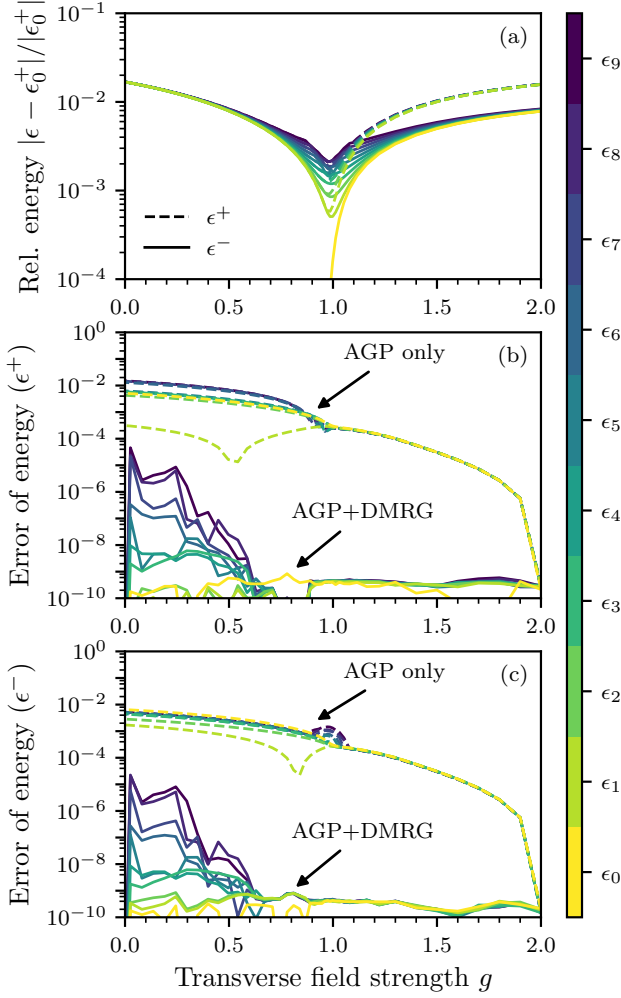


FIG. 6. Transverse field Ising model at $L = 120$. We present the accuracy of our method by comparing the ten lowest-lying states (refer to color bar for ordering) from transverse field strength $g = 2$ to $g = 0$, while passing the critical point at $g = 1$. We do this separately for even and odd parity sectors. In (a), we plot the relative energies in the even and odd sectors (denoted by dashed and solid lines, respectively) with respect to the ground state of the even sector (ϵ_0^+) against g obtained by using AGP+DMRG. In (b) and (c), we plot the error of the energy as a function of transverse field g , for even and odd sectors, respectively, where the dashed and solid lines represent data from using AGP evolution only and AGP+DMRG, respectively.

exact energy differences $|\epsilon_0^+ - \epsilon_1^+|/|\epsilon_0^+|$, $|\epsilon_3^+ - \epsilon_4^+|/|\epsilon_3^+|$, and $|\epsilon_9^+ - \epsilon_8^+|/|\epsilon_9^+|$, respectively, for varying system sizes. While not clearly shown, DMRG initialized with AGP-prepared states outperforms DMRG initialized with random states at $L = 120$ for the highest excited state studied in the even sector, where the errors are $2.2(16) \times 10^{-9}$ and $2.3(38) \times 10^{-8}$, respectively.

B. Scanning phase diagrams with AGP evolution of tensor networks

Understanding the landscape of the system's phases, including the locations of quantum critical points, can provide critical information on numerous physical phenomena such as high-temperature superconductivity [52–54], quantum hall states [55, 56], magnetic insulators [57], and much more. However, finding low-lying energy eigenstates (and thereby studying the phases) becomes particularly challenging near phase transitions, where the correlation length generally diverges. Hence, scanning phase diagrams is generally difficult if passing through different phases. While standard DMRG procedure can be employed, DMRG optimization can become expensive and difficult to converge near phase transitions. To prevent this, we augment our AGP evolution procedure by adding accurate DMRG calculations along the trajectory (coined AGP+DMRG) to find low-lying energy eigenstates. This makes finding phase diagrams more computationally efficient and robust, for using AGP-initialized states as ansatzes is more reliable than random trial states. Finally, using the AGP provides automated scanning of the phase diagram by automatically using more steps near the phase transitions.

Using AGP+DMRG, we compute the low-lying energy spectra of the transverse field Ising model and propagate them to scan its phase diagrams. In the transverse field Ising model, we pass through the critical point at $g = 1$, starting at $g = 2$ and terminating at $g = 0$, while propagating ten lowest-lying states in even and odd sectors, respectively, with system size $L = 120$. Specifically, 40 values of g 's are considered with similar spacing as shown in Fig. 4 but with δg capped at 0.1 instead (we find that allowing larger steps than that leads to large errors in the adiabatic evolution). In contrast to Sec. IV A, however, we perform more expensive intermediary DMRG calculations to ensure more accurate results for all values of g considered. The relative energies with respect to the ground state energy (ϵ_0^+) are shown in Fig. 6. As shown, when $g \lesssim 1$, $\epsilon_i^+ \approx \epsilon_i^-$ for all integer i 's such that $0 \leq i \leq 9$. Furthermore, as shown in Figs. 6b and 6c, incorporating intermediary DMRG calculations in between AGP evolutions drastically improves the results. Importantly, AGP+DMRG is able to correctly calculate the excited states even while passing through the critical point. While not shown, AGP calculation and evolution times only contributed to roughly 10% of the total computation time with the remainder taken up by DMRG. On average, intermediary DMRG calculations at each value of g took roughly 88% less time than the DMRG calculations at starting point $g = 2$. It is clear that AGP-related computation is hardly the bottleneck of AGP+DMRG.

Now, we consider the non-integrable Ising model by adding a longitudinal field with strength h to our Hamil-

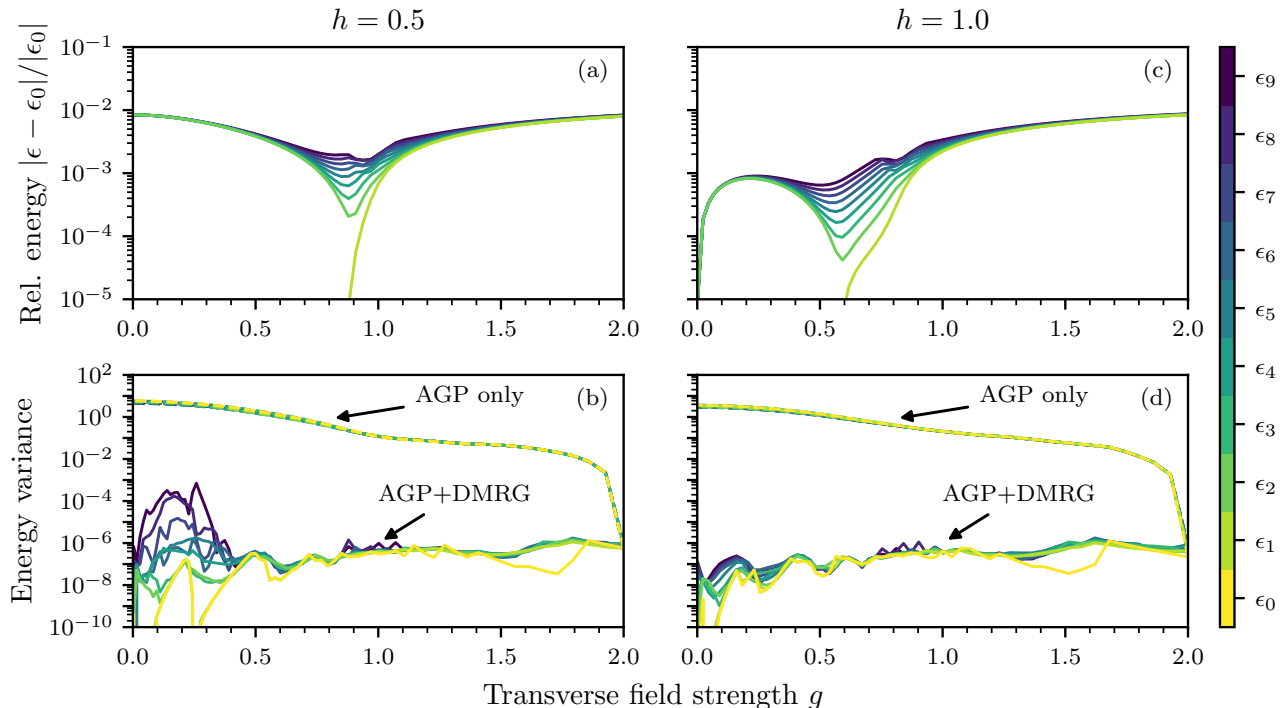


FIG. 7. Ising model with transverse and longitudinal fields at $L = 120$ with longitudinal field strengths $h = 0.5, 1$. We compute the ten lowest-lying states (refer to color bar for ordering) from transverse field strength $g = 2$ to $g = 0$ at $h = 0.5$ in (a) and (b) and $h = 1$ in (c) and (d). (a) and (c) show the relative energy against the ground state energy ϵ_0 for $h = 0.5$ and $h = 1$, respectively. (b) and (d) show the energy variance of the states ($\sigma_\epsilon^2 = \langle \mathcal{H}^2 \rangle - \langle \mathcal{H} \rangle^2$) for $h = 0.5$ and $h = 1.0$, respectively, where the solid lines represent data from using AGP+DMRG while dashed lines represent AGP evolution only.

tionian in Eq. (9):

$$\mathcal{H}_{\text{LTFIM}} = \sum_{i=1}^{L-1} \sigma_i^x \sigma_{i+1}^x + g \sum_{i=1}^L \sigma_i^z + h \sum_{i=1}^L \sigma_i^x. \quad (12)$$

Contrary to the previous model, this Hamiltonian is not exactly solvable, generically has no symmetry sectors (open boundary conditions are employed), and is generally chaotic [58]. This model has a more complicated phase structure and exhibits phase transitions at $0 \leq g \leq 1$ when $0 \leq h \leq 2$. Here, we employ AGP+DMRG to study the low-lying spectrum of $\mathcal{H}_{\text{LTFIM}}$ as we pass through phase transitions for system size $L = 120$ at two fixed values of $h = 0.5, 1$. In Fig. 7, we propagate ten lowest-lying states from $g = 2$ to $g = 0$ at $h = 0.5$ and $h = 1$. In Figs. 7a and 7c, we plot the energy spectrum of the states relative to the ground state energy for $h = 0.5$ and $h = 1$, respectively. Notably, the energy spectrum shown in Fig. 7a is similar to that shown in Fig. 6c, with slight differences in values of g at which the ground state becomes doubly degenerate. Similar to the integrable case, the error increases when $g \lesssim 0.5$ as the excited states become highly degenerate. However, in Fig. 7c, we have a noticeably different energy spectrum with all of the ten-lowest lying states becoming degenerate at $g = 0$. As we lack access to exact energies, we compute the energy variance $\sigma_\epsilon^2 = \langle \mathcal{H}^2 \rangle - \langle \mathcal{H} \rangle^2$ for

$h = 0.5$ and $h = 1$ in Figs. 7b and 7d, respectively, where the expectation values are taken with respect to computed states $|n\rangle$ such that $\mathcal{H}|n\rangle \approx \epsilon_n|n\rangle$. The variance provides a measure of how close $|n\rangle$ is to being an eigenstate of \mathcal{H} . As shown, a variance on the order of 10^{-6} is achievable using AGP+DMRG. Once again, the AGP calculation and evolution contributed to roughly 12% of the total runtime with DMRG calculations taking up the rest. Further, on average, intermediary DMRG calculations at each value of g took roughly 80% less time than the DMRG calculations at $g = 2$.

V. CONCLUSION

We presented a tensor network method for computing the adiabatic gauge potential, the generator of adiabatic eigenstate deformations, as a tensor network and tested the method on some paradigmatic models. For simplicity, a matrix product operator (MPO) ansatz was used, though generalizing to other tensor networks (or other ansatzes) is straightforward.

Future work and applications are abundant. Firstly, our choice of the particular excited state DMRG method used is arbitrary and, in principle, adiabatically evolving matrix product states with the adiabatic gauge poten-

tial has the possibility to improve other excited state (or ground state) DMRG algorithms. Here are more applications of the AGP in tensor networks:

1. Some recent excited state DMRG methods could directly benefit from our results such as for adiabatic transport of orbital wavefunctions in molecular crystals [11] and multi-target matrix product state ansatz [16].
2. The AGP could be applied to other variational tensor network methods: other excited state methods, 2D tensor network state (TNS) or projected entangled pair states (PEPS) methods, preparation of starting states for dynamics, and more.
3. Continuum systems of weakly interacting electrons can be slow to converge with DMRG [59–63]. This convergence could be improved by preparing a free fermion state as a tensor network [40–45], evolving it with the AGP to an interacting regime, and then performing DMRG using the AGP-prepared state.
4. State preparation for quantum computers could be implemented using the AGP. Namely, one could prepare an MPS or TNS, adiabatically evolve it to the desired parameter regime using the AGP, and then convert it to a low-depth circuit [64, 65]. This could help produce starting states for simulations with quantum chemistry [66, 67], quantum field theory [68, 69], variational quantum circuits [70–75], and more.
5. A natural extension of our work lies on the tensor network simulations of adiabatic quantum algorithms: quantum approximate optimization algorithms (QAOAs) [76], Grover’s algorithm [77, 78], and adiabatic quantum computation [79].
6. The AGP as an MPO can be generalized to more Hamiltonian parameters and be used to study, for example, geodesics in parameter space [30, 80] for larger system sizes.
7. In Ref. [81], they use a sampling of states in phase space, determined by computing the residuals of high-dimensional eigenvalue problems, to map out

the rest of the phase diagram. Instead, the AGP could be used to automate this sampling process to lower the computational cost.

8. Ref. [82] showed that adiabatic time evolution of highly excited states, particularly quantum scars, can be robustly performed. Using this, our work could be extended to adiabatically evolving highly excited states with the AGP.

Computing Resources and Software Packages

The code used to produce the numerical results in this paper was written using the **ITensorTDVP.jl** package [83] — a publicly available Julia [84] package for solving a variety of equations in terms of MPS including finding eigenstates with DMRG and DMRG-X [85], performing time evolution with TDVP, and solving linear equations of MPS. It is built on top of **ITensors.jl** [86], which provides the basic tensor operations and MPO/MPS types. For the MPS linear solver in **ITensorTDVP.jl**, we use the *linsolve* method from **KrylovKit.jl** [87] as the local solver, which implements the GMRES algorithm [88]. Calculations of the AGPs for all of the numerical computations were performed on a personal laptop while the rest of the numerics were performed on Flatiron Institute’s Rusty computing cluster. The code to construct the AGP as an MPO is available in **ITensorAGP.jl** [89].

The tensor network diagrams in this paper were produced using the publicly available package **GraphTikZ.jl** [90], a general-purpose Julia package for visualizing graphs, including tensor networks.

ACKNOWLEDGMENTS

The authors thank Miles Stoudenmire and Anatoli Polkovnikov for helpful discussions. M.F. and D.S. are grateful for ongoing support through the Flatiron Institute, a division of the Simons Foundation. H.K. acknowledges partial support from NSF (grant no. DMR-2103658). D.S. is partially supported by AFOSR (grant no. FA9550-21-1-0236) and NSF (grant no. OAC-2118310).

-
- [1] S. R. White, Phys. Rev. Lett. **69**, 2863 (1992).
 - [2] U. Schollwöck, Annals of Physics **326**, 96 (2011).
 - [3] F. Verstraete, V. Murg, and J. Cirac, Advances in Physics **57**, 143 (2008).
 - [4] I. P. McCulloch, Journal of Statistical Mechanics: Theory and Experiment **2007**, P10014 (2007).
 - [5] L. Wang and A. W. Sandvik, Phys. Rev. Lett. **121**, 107202 (2018).
 - [6] J. Yang, A. W. Sandvik, and L. Wang, Phys. Rev. B **105**, L060409 (2022).
 - [7] S. R. White, Phys. Rev. B **48**, 10345 (1993).
 - [8] S. Östlund and S. Rommer, Phys. Rev. Lett. **75**, 3537 (1995).
 - [9] J. Haegeman, B. Pirvu, D. J. Weir, J. I. Cirac, T. J. Osborne, H. Verschelde, and F. Verstraete, Phys. Rev. B **85**, 100408 (2012).
 - [10] M. C. Bañuls, K. Cichy, J. I. Cirac, and K. Jansen, Journal of High Energy Physics **2013**, 1 (2013).
 - [11] W. Hu and G. K.-L. Chan, Journal of Chemical Theory and Computation **11**, 3000 (2015), pMID: 26575737.

- [12] V. Khemani, F. Pollmann, and S. L. Sondhi, *Phys. Rev. Lett.* **116**, 247204 (2016).
- [13] B. Roberts, T. Vidick, and O. I. Motrunich, *Phys. Rev. B* **96**, 214203 (2017).
- [14] M. Van Damme, R. Vanhove, J. Haegeman, F. Verstraete, and L. Vanderstraeten, *Phys. Rev. B* **104**, 115142 (2021).
- [15] Q. Li, Y. Gao, Y.-Y. He, Y. Qi, B.-B. Chen, and W. Li, *Phys. Rev. Lett.* **130**, 226502 (2023).
- [16] X. Li, Z. Zhou, G. Xu, R. Chi, Y. Guo, T. Liu, H. Liao, and T. Xiang, Accurate determination of low-energy eigenspectra with multi-target matrix product states (2023), arXiv:2305.15868 [cond-mat.str-el].
- [17] M. Kolodrubetz, D. Sels, P. Mehta, and A. Polkovnikov, *Physics Reports* **697**, 1 (2017).
- [18] M. V. Berry, *Journal of Physics A: Mathematical and Theoretical* **42**, 365303 (2009).
- [19] M. Demirplak and S. A. Rice, *Journal of Physical Chemistry A* **107**, 9937 (2003).
- [20] D. Sels and A. Polkovnikov, *Proceedings of the National Academy of Sciences* **114**, E3909 (2017).
- [21] P. W. Claeys, M. Pandey, D. Sels, and A. Polkovnikov, *Phys. Rev. Lett.* **123**, 090602 (2019).
- [22] I. Čepaitė, A. Polkovnikov, A. J. Daley, and C. W. Duncan, *PRX Quantum* **4**, 010312 (2023).
- [23] M. Pandey, P. W. Claeys, D. K. Campbell, A. Polkovnikov, and D. Sels, *Phys. Rev. X* **10**, 041017 (2020).
- [24] M. Schwarz, K. Temme, and F. Verstraete, *Phys. Rev. Lett.* **108**, 110502 (2012).
- [25] M. Schwarz, K. Temme, F. Verstraete, D. Perez-Garcia, and T. S. Cubitt, *Phys. Rev. A* **88**, 032321 (2013).
- [26] Y. Ge, A. Molnár, and J. I. Cirac, *Phys. Rev. Lett.* **116**, 080503 (2016).
- [27] E. Cruz, F. Baccari, J. Tura, N. Schuch, and J. I. Cirac, *Phys. Rev. Res.* **4**, 023161 (2022).
- [28] Z.-Y. Wei, D. Malz, and J. I. Cirac, *Phys. Rev. Res.* **5**, L022037 (2023).
- [29] M. Ljubotina, B. Roos, D. A. Abanin, and M. Serbyn, *PRX Quantum* **3**, 030343 (2022).
- [30] S. Sugiura, P. W. Claeys, A. Dymarsky, and A. Polkovnikov, *Phys. Rev. Res.* **3**, 013102 (2021).
- [31] B. Pirvu, V. Murg, J. I. Cirac, and F. Verstraete, *New Journal of Physics* **12**, 025012 (2010).
- [32] G. K.-L. Chan, A. Keselman, N. Nakatani, Z. Li, and S. R. White, *The Journal of Chemical Physics* **145**, 014102 (2016).
- [33] C. Hubig, I. P. McCulloch, and U. Schollwöck, *Phys. Rev. B* **95**, 035129 (2017).
- [34] E. Jeckelmann, *Phys. Rev. B* **66**, 045114 (2002).
- [35] S. V. Dolgov and D. V. Savostyanov, *SIAM Journal on Scientific Computing* **36**, A2248 (2014).
- [36] J. Haegeman, J. I. Cirac, T. J. Osborne, I. Pizorn, H. Verschelde, and F. Verstraete, *Phys. Rev. Lett.* **107**, 070601 (2011).
- [37] J. Haegeman, C. Lubich, I. Oseledets, B. Vandereycken, and F. Verstraete, *Phys. Rev. B* **94**, 165116 (2016).
- [38] G. Vidal, *Phys. Rev. Lett.* **93**, 040502 (2004).
- [39] S. R. White and A. E. Feiguin, *Phys. Rev. Lett.* **93**, 076401 (2004).
- [40] C. V. Kraus, N. Schuch, F. Verstraete, and J. I. Cirac, *Phys. Rev. A* **81**, 052338 (2010).
- [41] G. Evenbly and G. Vidal, *Phys. Rev. B* **81**, 235102 (2010).
- [42] M. T. Fishman and S. R. White, *Phys. Rev. B* **92**, 075132 (2015).
- [43] N. Schuch and B. Bauer, *Phys. Rev. B* **100**, 245121 (2019).
- [44] Y.-H. Wu, L. Wang, and H.-H. Tu, *Phys. Rev. Lett.* **124**, 246401 (2020).
- [45] G. Petrica, B.-X. Zheng, G. K.-L. Chan, and B. K. Clark, *Phys. Rev. B* **103**, 125161 (2021).
- [46] L. Campos Venuti and P. Zanardi, *Phys. Rev. Lett.* **99**, 095701 (2007).
- [47] M. Kolodrubetz, V. Gritsev, and A. Polkovnikov, *Phys. Rev. B* **88**, 064304 (2013).
- [48] P. Zanardi and N. Paunković, *Phys. Rev. E* **74**, 031123 (2006).
- [49] P. Sierant, A. Maksymov, M. Kuś, and J. Zakrzewski, *Phys. Rev. E* **99**, 050102 (2019).
- [50] F. H. L. Essler and M. Fagotti, *Journal of Statistical Mechanics: Theory and Experiment* **2016**, 064002 (2016).
- [51] G. B. Mbeng, A. Russomanno, and G. E. Santoro, The quantum ising chain for beginners (2020), arXiv:2009.09208 [quant-ph].
- [52] E. Dagotto, *Rev. Mod. Phys.* **66**, 763 (1994).
- [53] M. Maple, *Journal of Magnetism and Magnetic Materials* **177-181**, 18 (1998), international Conference on Magnetism.
- [54] J. Orenstein and A. Millis, *Science* **288**, 468 (2000).
- [55] S. L. Sondhi, S. M. Girvin, J. P. Carini, and D. Shahar, *Rev. Mod. Phys.* **69**, 315 (1997).
- [56] V. Piazza, V. Pellegrini, F. Beltram, W. Wegscheider, T. Jungwirth, and A. H. MacDonald, *Nature* **402**, 638 (1999).
- [57] D. Bitko, T. F. Rosenbaum, and G. Aeppli, *Phys. Rev. Lett.* **77**, 940 (1996).
- [58] H. Kim, T. N. Ikeda, and D. A. Huse, *Phys. Rev. E* **90**, 052105 (2014).
- [59] E. M. Stoudenmire, L. O. Wagner, S. R. White, and K. Burke, *Phys. Rev. Lett.* **109**, 056402 (2012).
- [60] M. Dolfi, B. Bauer, M. Troyer, and Z. Ristivojevic, *Phys. Rev. Lett.* **109**, 020604 (2012).
- [61] E. M. Stoudenmire and S. R. White, *Phys. Rev. Lett.* **119**, 046401 (2017).
- [62] R. Haghshenas, Z.-H. Cui, and G. K.-L. Chan, *Phys. Rev. Res.* **3**, 023057 (2021).
- [63] S. Dutta, A. Buyskikh, A. J. Daley, and E. J. Mueller, *Phys. Rev. Lett.* **128**, 230401 (2022).
- [64] S.-J. Ran, *Phys. Rev. A* **101**, 032310 (2020).
- [65] L. Slattery and B. K. Clark, Quantum circuits for two-dimensional isometric tensor networks (2021), arXiv:2108.02792 [quant-ph].
- [66] I. Kassal, S. P. Jordan, P. J. Love, M. Mohseni, and A. Aspuru-Guzik, *Proceedings of the National Academy of Sciences* **105**, 18681 (2008).
- [67] 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, *et al.*, *Nature chemistry* **2**, 106 (2010).
- [68] A. Hamed Moosavian and S. Jordan, *Phys. Rev. A* **98**, 012332 (2018).
- [69] J. Preskill, Simulating quantum field theory with a quantum computer (2018), arXiv:1811.10085 [hep-lat].
- [70] W. Huggins, P. Patil, B. Mitchell, K. B. Whaley, and E. M. Stoudenmire, *Quantum Science and Technology* **4**, 024001 (2019).
- [71] J.-G. Liu, Y.-H. Zhang, Y. Wan, and L. Wang, *Phys.*

- Rev. Res. **1**, 023025 (2019).
- [72] D. A. Fedorov, B. Peng, N. Govind, and Y. Alexeev, *Materials Theory* **6**, 1 (2022).
 - [73] J. Dborin, F. Barratt, V. Wimalaweera, L. Wright, and A. G. Green, *Quantum Science and Technology* **7**, 035014 (2022).
 - [74] Z. Xu, Y. Fan, H. Shang, and C. Guo, Differentiable matrix product states for simulating variational quantum computational chemistry (2022), arXiv:2211.07983 [quant-ph].
 - [75] X. Hou, Q. Li, M.-H. Yung, X. Xu, Z. Wang, C. Guo, and X. Wang, A sequentially generated variational quantum circuit with polynomial complexity (2023), arXiv:2305.12856 [quant-ph].
 - [76] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm (2014), arXiv:1411.4028 [quant-ph].
 - [77] L. K. Grover, in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96 (Association for Computing Machinery, New York, NY, USA, 1996) p. 212–219.
 - [78] E. M. Stoudenmire and X. Waintal, Grover’s algorithm offers no quantum advantage (2023), arXiv:2303.11317 [quant-ph].
 - [79] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, Quantum computation by adiabatic evolution (2000), arXiv:quant-ph/0001106 [quant-ph].
 - [80] H. Kim and A. Polkovnikov, Integrability is attractive (2023), arXiv:2308.09745 [cond-mat.stat-mech].
 - [81] M. F. Herbst, B. Stamm, S. Wessel, and M. Rizzi, *Phys. Rev. E* **105**, 045303 (2022).
 - [82] H. Yarloo, H.-C. Zhang, and A. E. B. Nielsen, Adiabatic time evolution of highly excited states (2023), arXiv:2306.13967 [quant-ph].
 - [83] ITensorTDVP.jl, <https://github.com/ITensor/ITensorTDVP.jl> (2023).
 - [84] J. Bezanson, S. Karpinski, V. B. Shah, and A. Edelman, Julia: A fast dynamic language for technical computing (2012), arXiv:1209.5145 [cs.PL].
 - [85] V. Khemani, F. Pollmann, and S. L. Sondhi, *Phys. Rev. Lett.* **116**, 247204 (2016).
 - [86] M. Fishman, S. R. White, and E. M. Stoudenmire, *SciPost Phys. Codebases*, 4 (2022).
 - [87] KrylovKit.jl, <https://github.com/Jutho/KrylovKit.jl> (2022).
 - [88] Y. Saad and M. H. Schultz, *SIAM Journal on Scientific and Statistical Computing* **7**, 856 (1986).
 - [89] ITensorAGP.jl, <https://github.com/ITensor/ITensorAGP.jl> (2023).
 - [90] GraphTikZ.jl, <https://github.com/mtfishman/GraphTikZ.jl> (2023).