Faster Methods for Contracting Infinite Two-Dimensional Tensor Networks

M.T. Fishman,[1] L. Vanderstraeten,[2] V. Zauner-Stauber,[3] J. Haegeman,[2] and F. Verstraete[3, 2]

[1] *Institute for Quantum Information and Matter,*
California Institute of Technology, Pasadena, California 91125, USA
[2] *Ghent University, Faculty of Physics, Krijgslaan 281, 9000 Gent, Belgium*
[3] *Vienna Center for Quantum Technology, University of Vienna, Boltzmanngasse 5, 1090 Wien, Austria*

We revisit the corner transfer matrix renormalization group (CTMRG) method of Nishino and Okunishi for contracting two-dimensional (2D) tensor networks and demonstrate that its performance can be substantially improved by determining the tensors using an eigenvalue solver as opposed to the power method used in CTMRG. We also generalize the variational uniform matrix product state (VUMPS) ansatz for diagonalizing 1D quantum Hamiltonians to the case of 2D transfer matrices and discuss similarities with the corner methods. These two new algorithms will be crucial to improving the performance of variational infinite projected entangled pair state (PEPS) methods.

I. INTRODUCTION

Two-dimensional (2D) tensor networks are ubiquitous in many-body physics[1]. They occur naturally in the context 2D classical many-body systems as representations of partition functions[2–8] and can represent ground states, finite temperature states and the time evolution of 1D quantum systems, e.g. for systems with local interactions in terms of Trotter-Suzuki decompositions[9–17]. Additionally, they occur in the context of tensor product state (TPS)[18–21] or projected entangled pair state (PEPS)[22] representations of 2D quantum systems and boundaries of 3D classical systems. Most 2D tensor networks of interest do not allow exact solutions and can only be studied approximately, and a copious array of numerical tensor network methods have been developed over many decades for their study[2–8,13,15–17,22–36].

Methods for contracting 2D tensor networks fall roughly into two main categories, which we refer to as "coarse graining methods" and "boundary methods." Examples of coarse graining methods are tensor renormalization group (TRG)[27] and extensions such as second renormalization group (SRG)[30], higher order tensor renormalization group (HOTRG)[33], and tensor network renormalization (TNR)[34–36]. A common feature of these methods is that the local degrees of freedom are combined and truncated, so the Hilbert space of the network is explicitly changed at each step. For boundary methods, a matrix product state (MPS) is used as an ansatz for the environment, and this MPS is optimized in various ways. Boundary methods include the density matrix renormalization group (DMRG) algorithm[6,13,15,23,24,26,29], the corner transfer matrix renormalization group (CTMRG) algorithm[3–5,7,8], the time evolving block decimation (TEBD) algorithm[16,17,28,37], the time dependent variational principle (TDVP)[38,39], etc. Boundary methods have certain advantages: they are optimized iteratively instead of optimized layer by layer like most coarse graining methods, the form of the environments can make it much easier to calculate arbitrary correlation functions, and they appear to be very well-suited for performing PEPS calculations[22,40–44].

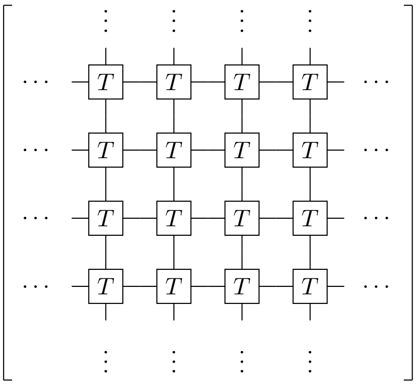
The history of modern boundary methods goes back to Nishino's application of DMRG to calculating fixed points of transfer matrices[6]. Soon after, Nishino and Okunishi created the CTMRG algorithm[7,8] by combining the corner transfer matrix (CTM) method of Baxter[3–5] and White's DMRG algorithm[23,24]. CTMRG was initially introduced as a powerful numerical tool for contracting 2D classical partition functions. In addition, it has been used extensively in TPS/PEPS calculations of 3D classical and 2D quantum systems, where it is used to approximate the contraction of 2D tensor networks that arise in those calculations. CTMRG was used as the contraction method in the original TPS calculations[19–21,45]. An MPS-based boundary method was used for the original finite PEPS calculation[22] while iTEBD, an MPS-based power method, was used to perform the original infinite PEPS[40,41] calculations. Since then, PEPS calculations in the thermodynamic limit have mostly been performed using CTMRG as the contraction method, and a variety of advancements have been made to the method over recent years in that context[43,46–53].

Here, we present two new approaches that improve upon the speed of CTMRG for contracting 2D tensor networks in the thermodynamic limit. First, we present a transfer matrix version of the recently introduced variational uniform matrix product state (VUMPS)[54] algorithm for contracting 2D tensor networks. We also present a new corner method analogous to CTMRG that better exploits translational invariance by solving for the environment tensors using a set of fixed point equations. We present benchmark results for VUMPS and our new corner method, showing remarkable speedups over CTMRG, particularly for systems near criticality. Our benchmarks include a variety of both 2D statistical mechanics models and 2D quantum systems represented as PEPS.

II. PROBLEM STATEMENT

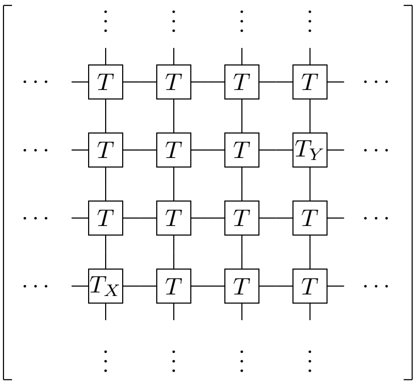
We are interested in the approximate numerical contraction of infinite 2D tensor networks. For simplicity, throughout the paper we will focus on tensor networks on an infinite square lattice with a single site unit cell. We are agnostic about where the tensor network comes from: it could be a 2D classical partition function, the norm of a PEPS, etc.

For concreteness, we are interested in evaluating the contraction of the following tensor network

$\kappa^{MN} \equiv Tr$  (1)

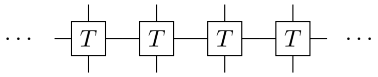
(for readers unfamiliar with tensor networks, we refer them to Ref. 55 for an introduction). In Eq. (1), we work directly in the thermodynamic limit, i.e. the number of lattice sites in the horizontal and vertical directions, M, N, approaches ∞. $Tr[...]$ denotes two traces, one over the open horizontal indices and another over the open vertical indices. If the network represents a 2D classical partition function, the fourth-order tensor T is related to the local Boltzmann weight (possibly up to a local tensor renormalization) and κ is defined to be the "partition function per site,"[5] related to the free energy per site. If the network is the evaluation of the norm of a PEPS, each tensor T is the bra and ket PEPS tensor at each site contracted over the physical index[56], and κ is the norm per site.

We are also interested in calculating observables such as expectation values of local operators or correlation functions. In terms of the tensor network, these are represented as impurity sites, such as:

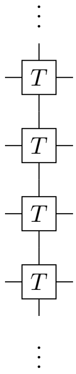
$\langle XY \rangle = Tr$  $/\ \kappa^{MN}$. (2)

We want a contraction method that makes it easy to calculate arbitrary correlation functions, since they show up in e.g. calculating structure factors or summing Hamiltonian terms in variational PEPS ground state optimizations[42–44]. For this reason we focus on MPS boundary methods, which make it much easier to calculate arbitrary correlation functions. It is more challenging in methods like TRG/TNR where all of the tensors at each layer must properly be kept track of, and calculating arbitrary correlation functions on the lattice is potentially very complicated.

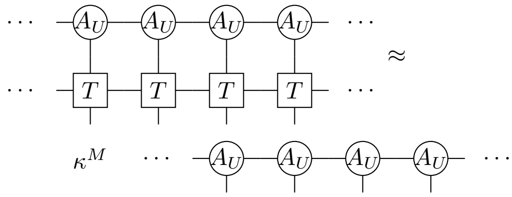
Here we will also define the row-to-row transfer matrix, which is simply a single infinite row of the tensor network:

 . (3)

The row-to-row transfer matrix is an infinite, translationally invariant matrix product operator (MPO). We also define the column-to-column transfer matrix as an infinite column of the tensor network:

 . (4)

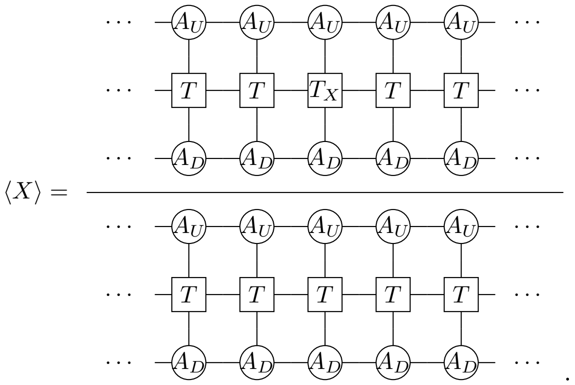
For MPS boundary methods, the evaluation of diagrams like Eq. (1)–(2) is performed by finding the leading up and down eigenvectors of the row-to-row transfer matrix portrayed in (3) and the leading left and right eigenvectors of column-to-column transfer matrix portrayed in (4). Exact MPS representations of these eigenvectors are in general infinitely large, but for many 2D tensor networks representing physical many-body systems good MPS approximations exist (in some cases provably[57]). We refer to these uniform MPS fixed points as the up, down, left and right boundary MPSs, and call their MPS tensors respectively A_U, A_D, A_L and A_R. As an example, the fixed point equation for the top MPS is as follows:



(5)

where κ is the partition function or norm per site defined in Eq. (1). At the fixed point, analogous equations to Eq. (5) should be satisfied by the other boundary MPSs.

Once the boundary MPSs are obtained, local observables and correlations functions can be computed efficiently. For example, the expectation value of a local observable can be calculated from the up and down boundary MPSs as follows:



(6)

These networks can then be contracted efficiently[28]. Arbitrary correlation functions can also be computed efficiently using fixed point boundary MPSs[42].

There are many different approaches to obtaining the four boundary MPS fixed points of the row-to-row transfer matrix (3) and column-to-column transfer matrix (4). In the next section, we review one very commonly used contraction method, the corner transfer renormalization group (CTMRG) algorithm of Nishino and Okunishi[3–5,7,8], and describe two new proposals, one based on the recently proposed variational uniform matrix product state (VUMPS) algorithm[54], and one that we refer to as the fixed point corner method (FPCM), which is like CTMRG but solves for the boundary tensors using a series of fixed point equations.

III. ALGORITHM OVERVIEW

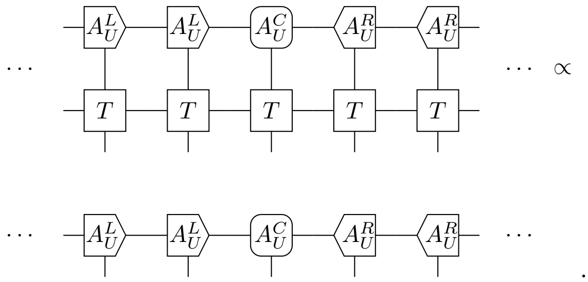
One strategy for evaluating Eq. (1)–(2) involves finding a single boundary MPS eigenvector at a time. The infinite time evolving block decimation (iTEBD) of Orús and Vidal[16,17,28] is an example of this strategy. In iTEBD, a power method is used to find the fixed point MPS eigenvector by repeatedly applying a row-to-row or column-to-column transfer matrix to a starting MPS. In this work, one of the strategies we propose also focuses on solving for a single MPS eigenvector for each direction at a time. Instead of iTEBD, we propose using the recently introduced variational uniform matrix product state (VUMPS) algorithm[54], which can be viewed as an improvement on the infinite density matrix renormalization group (iDMRG)[23,24,29] where an MPS is optimized directly in the thermodynamic limit instead of grown site-by-site to reach the thermodynamic limit. VUMPS was originally applied to finding ground state approximations of 1D and quasi-1D quantum states, where it was shown to substantially improve the computational performance over iTEBD and iDMRG. In analogy to how Nishino introduced the transfer matrix DMRG (TMRG) method as an extension of applying DMRG to finding the fixed point MPS approximation of the transfer matrices of partition functions[6], in Section III A we show how VUMPS can be applied to find the fixed point of infinite uniform transfer matrices.

Another strategy for finding the boundary MPSs is to attempt to find all four MPSs at once. An example of this approach is the corner transfer matrix (CTM) method of Baxter[3–5], and its improvement by Nishino and Okunishi called the corner transfer matrix renormalization group (CTMRG)[7,8]. In the CTMRG algorithm, all four boundary MPSs are iteratively optimized. We give a brief review of CTMRG in Section III B. One of the new methods we propose in this work, which we refer to as the fixed point corner method (FPCM) and is explained in Section III C, also solves for all four MPS fixed points at once. Like CTMRG, FPCM uses CTMs, but solves for the CTMs and MPS tensors using a series of fixed point equations (which were originally written down by Baxter[3–5]).

A. VUMPS for contracting infinite 2D tensor networks

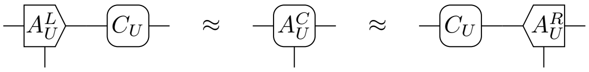
Here, we present the application of the recently proposed VUMPS algorithm[54] to finding MPS fixed points of infinitely large, translationally invariant transfer matrices. Essentially, we apply VUMPS to the problem of directly finding fixed points of the form shown in Eq. (5).

We now present VUMPS for obtaining the top fixed point MPS of the network. We would like to find the uniform MPS satisfying Eq. (5). In VUMPS, we use the mixed canonical form of the MPS, so Eq. (5) becomes:



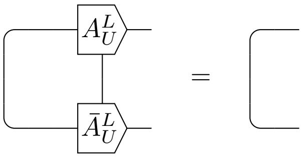
(7)

In the mixed canonical gauge, for the state to be (approximately) translationally invariant the tensors must satisfy the relations:

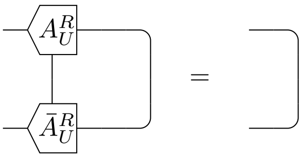


(8)

where the singular values of the matrix C_U are the Schmidt values of the uniform MPS. Note the inequalities in Eq. (8), since the relationships will not generally all simultaneously be satisfied exactly during the optimization. How accurately they are satisfied will relate to how translationally invariant the state is, and should be satisfied to very high accuracy at the fixed point of the VUMPS algorithm. Additionally, A_U^L and A_U^R are isometric tensors satisfying:



(9)

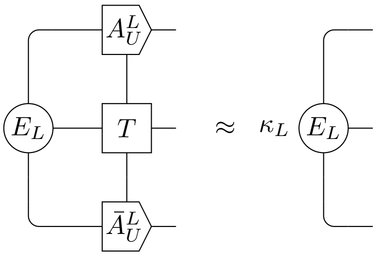


(10)

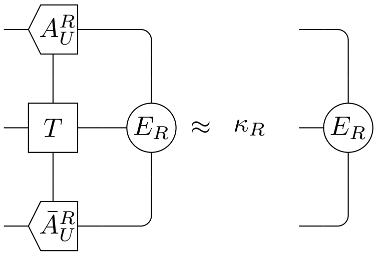
at all times. Any uniform MPS can be turned into this form, for example with the algorithm introduced in Ref. 28 in the context of iTEBD or with the algorithm introduced in Ref. 1 and expanded on in Appendix B 1.

The VUMPS algorithm proceeds by repeating the following steps until convergence:

1. Solve for the environments:



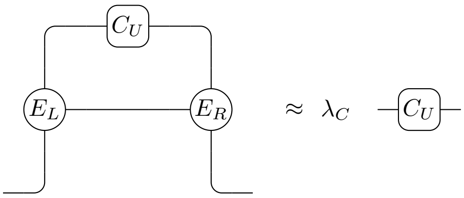
(11)



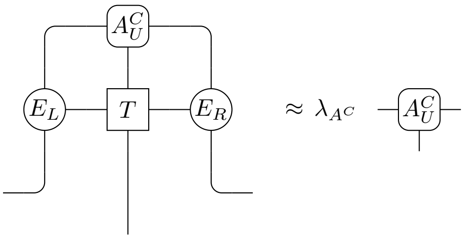
(12)

where $\kappa_L \approx \kappa_R$ up to errors in Eq. (8).

2. Solve for zero-site and single-site tensors:



(13)

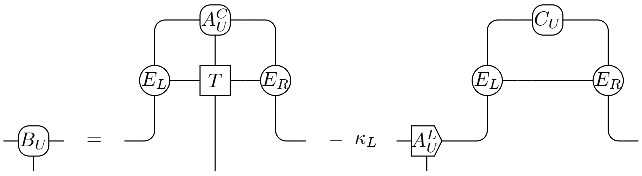


(14)

where $\lambda_{A^C}/\lambda_C \approx \kappa_{L/R}$ near or at the fixed point.

3. From A_U^C and C_U found in step 2, find new MPS tensors A_U^L and A_U^R satisfying Eq. (8). Techniques for numerically solving these equations are described in the original VUMPS proposal in Ref. 54.

The VUMPS algorithm proceeds by repeating steps 1–3 until convergence. Convergence can be measured, for example, by the change in the singular values of C from step to step. Another measure for the convergence that can be used is the norm of the residual B_U:



(15)

(or the analogous right version), similar to the gradient discussed in Ref. 54.

For finding the fixed point of a row-to-row or column-to-column transfer matrix Hermitian about the horizontal, this scheme maps directly to the original VUMPS proposal[54], and the algorithm solves for both the top and bottom fixed points, which are just Hermitian conjugates of each other. For a Hermitian row-to-row transfer matrix, the fixed points environments E_L, E_R are related to the fixed points of the boundary MPS tensors A_L, A_R used in the CTMRG ansatz (which we review in Section III B), the gauged MPS tensors A_U^L, A_U^R are related to the fixed points of isometric projectors used to renormalize the CTM environment (i.e. the eigenvectors of the product of the four CTMs), and the center tensor C_U is related to the product of CTMs $C_{LU}C_{UR}$. This correspondence is discussed in more detail in Ref. 1. A similar correspondence between the fixed point of CTMRG and the fixed point of DMRG applied to Hermitian transfer matrices was pointed out by Nishino and Okunishi[6–8].

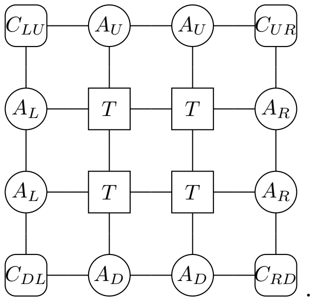
For contracting 2D statistical mechanics partition functions and calculating the norm of a PEPS, transfer matrix VUMPS is in fact simpler than the original proposal, because we do not in general have to be concerned about summing Hamiltonian terms which can lead to divergences if the fixed point is not calculated properly[54,58], and methods such as Arnoldi can be directly employed to find the fixed points. One may have to be more careful contracting networks that involve sums of local operators, such as when calculating structure factors or gradients of PEPS. See Ref. 42–44 for approaches to contracting such networks, where the environments calculated from the norm of a PEPS are used to aid in the contraction.

In general for a non-Hermitian network, to get the environment for calculating local observables, one must additionally solve for the bottom fixed point (and in order to calculate arbitrary correlation functions, the left and right fixed point MPSs as well). For a network that isn't "very asymmetric," the top fixed point can be used as a good starting point for the bottom fixed point MPS. It is also important to note that in the case of symmetry breaking, one should take care that the fixed points in different directions are all compatible, in the sense that they correspond to the same symmetry-broken states.

For non-Hermitian networks, the method we propose here is analagous to iTEBD, where each of the four boundary MPSs is solved for in separate optimizations (although in iTEBD the fixed points in each direction are obtained with power methods, which was shown to be slower than VUMPS in Ref. 54). An alternative approach from the one proposed here is to solve for two opposing fixed points in the same optimization (for example both the top and bottom fixed points of the row-to-row transfer matrix). This is approach has been used in the context of applying DMRG to non-Hermitian transfer matrices (TMRG)[13,15,26,31,32,59–63]. It would be interesting to generalize the transfer matrix VUMPS algorithm to solving for both fixed points at once, but we do not explore that here.

B. Corner transfer matrix renormalization group (CTMRG) review

In this section we review the corner transfer matrix renormalization group algorithm, which was originally introduced by Nishino and Okunishi and extended in a variety of other works in the context of PEPS calculations. The general ansatz used for the environment in the corner transfer matrix renormalization group (CTMRG) algorithm is as follows:



. (16)

The matrices $\{C_i\}$ in Eq. (16), known as the corner transfer matrices (CTMs), were originally introduced by Baxter for studying 2D classical statistical mechanics problems[3–5,7]. The CTMs represent approximations of the infinite corners of the tensor network. The boundary MPS tensors $\{A_i\}$ in Eq. (16) represent approximations of the half-row transfer matrices (HRTMs) and half-column transfer matrices (HCTMs). In our notation, C_{LU} denotes the CTM approximating the upper left corner of the network, A_L denotes the left HRTM of the network, A_U denotes the upper HCTM of the network, etc. We refer to the set of tensors $\{C_i, A_j\}$ as the environment of the 2D tensor network.

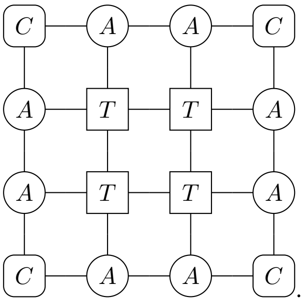
The CTMRG algorithm is thought of in terms of contracting row-to-row transfer matrices and/or column-to-column transfer matrices composed of tensor T into the environment, either simultaneously in multiple directions or sequentially in specified directions (depending on details of the renormalization scheme).

If the row-to-row and column-to-column transfer matrices of the network are absorbed into the environment indefinitely, then the environment tensors would grow exponentially in size, so some sort of truncation scheme is required. The truncation is referred to as renormalization. This renormalization of the enlarged environment is performed by introducing projectors into the network. There are multiple methods available for how to grow the lattice as well as how to choose the projectors. We will start by describing how these projectors are chosen for tensor networks with reflection symmetries, where the ansatz in Eq. (16) can be constrained.

1. *Symmetric CTMRG review*

To get some intuition for how CTMRG works, it is useful to discuss the case in which the network tensor T is Hermitian about all reflections (about the horizontal, vertical and diagonals, in other words $T_{lurd} = \bar{T}_{ruld} = \bar{T}_{ldru} = \bar{T}_{drul}$[64]). This is the case for many statistical mechanics models. In this case, we can constrain the environment tensors in the ansatz in Eq. (16) to satisfy

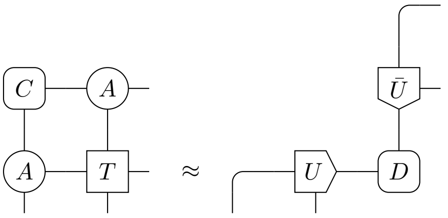
$A_U = A_R = A_D = A_L \equiv A$, $C_{LU} = C_{UR} = C_{RD} = C_{DL} \equiv C$, and additionally impose $C = C^\dagger$ and $A^s = (A^s)^\dagger$[65]. Eq. (16) becomes:



(17)

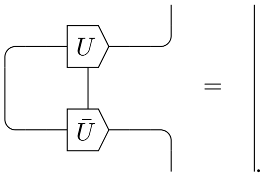
This is the CTMRG case that was covered in the initial proposal of Nishino and Okunishi[7,8] (though extensions to the asymmetric case were discussed). The CTMRG algorithm consists of obtaining the projector by "growing" the corner transfer matrices C by absorbing surrounding network and environment tensors and performing a Hermitian eigendecomposition, and we summarize the algorithm here:

1. Obtain the projector from a Hermitian eigendecomposition of the grown corner transfer matrix[66]:



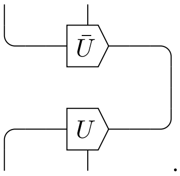
(18)

   where we use the convention that the indices of the tensor in the diagram are ordered clockwise, except when the complex conjugate is taken in which case the ordering is reversed. In Eq. (18), the tensor network on the left side is contracted, reshaped into a Hermitian matrix, a Hermitian eigendecomposition is performed, and the bond dimension is truncated according to the eigenvalues. D is a diagonal matrix storing the largest magnitude (real) eigenvalues. U is the matrix of the orthonormal eigenvectors associated with the largest eigenvalues D reshaped into an isometric tensor. U satisfies $(U^s)^\dagger U^s = I$ (using Einstein summation convention) or diagramatically:



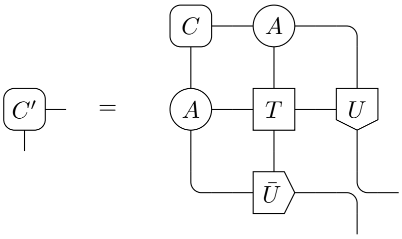
(19)

2. Renormalize the grown environment. The new CTMRG environment is obtained by absorbing a row and column of the tensor network in each direction into the environment. The renormalization is performed with the projector $U^s(U^{s'})^\dagger$, which diagramatically is:

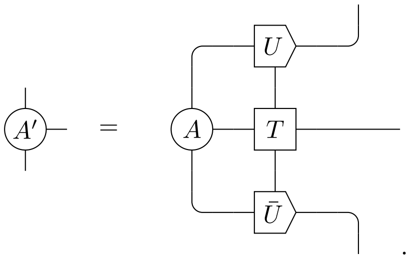


(20)

   The projector Eq. (20) is inserted into the grown boundary environment at every link in the environment, and grown environment tensors are renormalized to obtain the new environment tensors. The new environment tensors C' and A' are obtained as follows:



(21)



(22)

   Of course, from Eq. (18), we can trivially see that $C' = D$, but the more general form of Eq. (21) will be useful when we discuss generalizing to situations where the tensor network is comprised of asymmetric tensors T, and when we discuss our new fixed point corner method.

The CTMRG algorithm essentially involves iterating steps 1 and 2 until convergence (for example, measured by the difference in the eigenvalues of the corner transfer matrices between steps), where one must make sure to normalize the HRTMs, HCTMs, and CTMs at each steps.

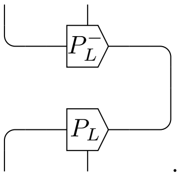
Extensions to networks with other types of symmetries are straightforward. If the network is Hermitian about the horizontal and vertical directions but not the diagonal directions, we can impose $A_U = A_D \equiv A$ (where $A = A^\dagger$) and $A_L = A_R \equiv B$ (where $B = B^\dagger$), and $C_{LU} = C_{UR}^\dagger = C_{RD} = C_{DL}^\dagger \equiv C$ (where $C = C^\dagger$ if the gauge is chosen properly). In that case, a generalization of Eq. (18) can be used to obtain projectors for the left/right direction and up/down direction using the left and right singular vectors respectively obtained from the SVD of one of the grown corners.

If the tensor network is not Hermitian about either a horizontal or vertical reflection, a simple Hermitian eigendecomposition like that shown in Eq. (18) will not suffice, and a more involved scheme must be invoked. In general, the projectors used will not be isometric, and there will be different projectors for renormalizing each direction (left, up, right, and down). We refer readers to Ref. 50 for a discussion of applying CTMRG to the case when the network is Hermitian about a single direction. In the next section, we will discuss strategies for generalizing CTMRG to fully asymmetric tensor networks.

2. *Asymmetric CTMRG review*

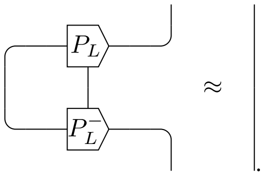
When the tensor network does not contain reflection symmetries as discussed in the previous section, in general there is not a unique way for choosing the projector and a variety of methods have been proposed. In the previous section, we discussed methods where the network was renormalized in all four directions at once. Here, we will focus on what is called the "directional" approach, where a single direction of the network is renormalized at a time, and a single CTMRG step constitutes cycling through the different directions. This approach makes the discussion easier, and is well-suited for PEPS calculations[49,51].

In the directional approach, the "left move" involves contracting just the column-to-column transfer matrix into the left environment, and renormalizing with a projector which we will call $P^s_L[P^-_L]^{s'}$ or diagramatically:



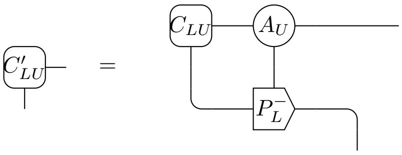
(23)

P^-_L denotes the approximate left inverse of P_L, in other words they satisfy $[P^-_L]^s P^s_L \approx I$ or diagramatically:

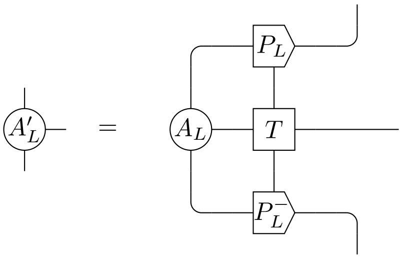


(24)

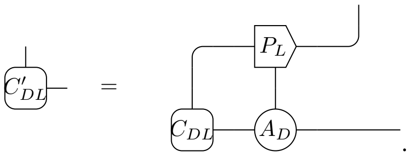
Using the projector in Eq. (23), the left move is shown below:



(25)



(26)

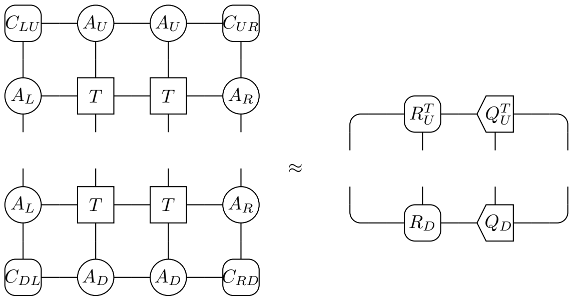


(27)

The diagrammatic notation we use for the tensors in the projector is suggestive, identifying them as MPS tensors. This notation will prove useful later on when we present our new algorithms. The up, right and down moves are simply rotated versions of the left move.

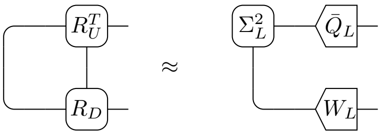
The projectors (i.e. Eq. (23)) are obtained from the current guess for the environment, and in general the choice is not unique. Many methods for obtaining these projectors have been proposed over the years[7,46–51]. In this work, for asymmetric CTMRG, we will use the method that is most commonly used in modern infinite PEPS calculations, the one proposed in Ref. 51. We summarize the left move of that asymmetric CTMRG method here:

1. The first step of the left move of the CTMRG method proposed in Ref. 51 is to perform the following QR decompositions:



(28)

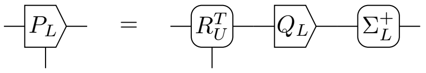
2. Next we obtain the tensors P_L, P^-_L from R_U and R_D obtained in previous step. This is done using a method we will refer to as "biorthogonalization" which was originally proposed in the context of transfer matrix DMRG (TMRG) in Ref. 31 and 32. The first step of the biorthogonalization procedure is to perform the following SVD:



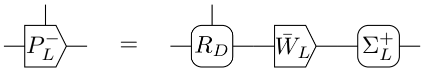
(29)

where the second line is obtained by taking the SVD $R_D^s[R_U^s]^T = W_L \Sigma_L^2 Q_L^\dagger$ and truncating according to the singular values to the desired bond dimension.

3. Finally, we obtain P_L, P_L^-:



(30)



(31)

Note that it may be necessary for stability to use a pseudoinverse of Σ_L, where we denote the pseudoinverse with Σ_L^+.

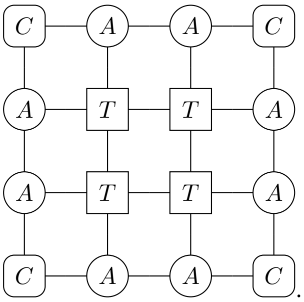
In Appendix A we discuss technical details about the stability of this method as well as some new alternatives. In the next section, we discuss a transfer matrix version of the variational uniform matrix product state (VUMPS) algorithm introduced in Ref. 54.

C. New fixed point corner method (FPCM)

Here we present a new corner method, which we refer to as the fixed point corner method (FPCM), which is similar to the CTMRG algorithm but solves for environment tensors in terms of fixed points.

1. *Symmetric FPCM*

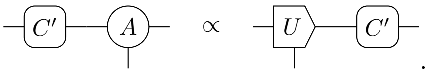
We start with the simplest version of our new fixed point corner method (FPCM), when the network is comprised of a tensor T that is Hermitian about all reflections. In this case, we use the same ansatz for the environment as we would use for the fully symmetric CTMRG algorithm, which we mentioned previously in Section III B 1 and repeated here:



(32)

As before, we also impose that $C = C^\dagger$ and $A^s = (A^s)^\dagger$. For this network, the FPCM proceeds as follows:

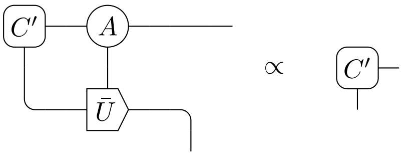
1. Isometrically gauge the uniform MPS composed of tensor A. Using A, find the isometric tensor U and (positive) symmetric matrix C' satisfying:



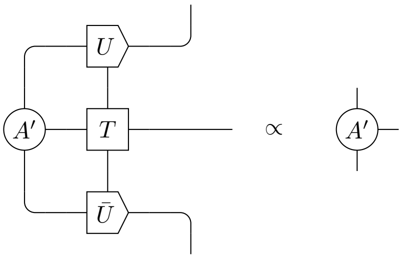
(33)

This is performed with a new uniform MPS gauging method described in Appendix B 1.

2. Obtain the new MPS boundary tensor A and the new CTM C using U found in step 1. This is done by numerically solving the following fixed point equations (in practice using an iterative method such as Arnoldi):



(34)



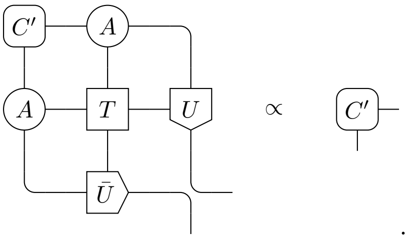
(35)

(note that Eq. 34 may or may not be different from how C' was solved for in step 1 but this alternative fixed point equation may give an improved CTM).

Then, steps 1 and 2 are repeated until convergence. We should point out that the boundary MPS tensor A solved for using the fixed point equation Eq. (35) may only be symmetric up to errors in the accuracy that the fixed point is solved to, and it may be useful to symmetrize the tensor explicitly during the optimization.

Note that a similar set of fixed point equations for numerically solving for boundary MPSs were discussed previously in Ref. 1. We would also like to point out that obtaining the isometry as proposed in Eq. (33) can be viewed as a translationally invariant version of the so-called "simplified one-directional 1D method" discussed in Ref. 50. More generally, the tensor U can be viewed as a translationally invariant version of the projector obtained in CTMRG.

Note that the following fixed point equation can be used to obtain a more accurate CTM C:



(36)

A similar fixed point equation for the CTM was discussed previously in Ref. 42 and 44.

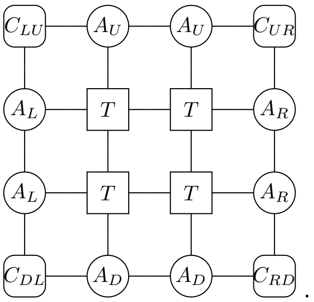
We also note that in practice, we find performing a few steps of CTMRG per step of the FPCM can help improve the convergence of the algorithm and obtain a more accurate fixed point environment. One can therefore think of FPCM as a way to speed up a CTMRG implementation, by performing a step of FPCM periodically during the CTMRG algorithm to help speed up convergence. In that case, it is important to solve for an improved CTM with Eq. (36) so that the best possible CTM is used for the CTMRG step.

In the next section, we will describe a generalization of this algorithm to asymmetric tensor networks.

2. *Asymmetric FPCM*

The asymmetric version of FPCM is not as straightforward as the symmetric version, analogous to the case for CTMRG. Our strategy is to determine translational invariant analogues of the CTMRG projectors shown in Eq. (25)–(27), and then determine the environment tensors $\{C_i, A_j\}$ from fixed point equations.

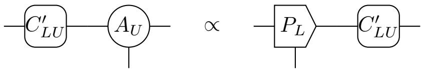
We use the same ansatz as that used for the asymmetric CTMRG algorithm (as presented in Sec. III B 2):



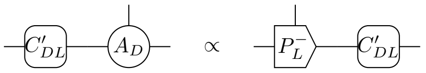
(37)

Using this ansatz for the environment, the left move of FPCM consists of the following steps:

1. "Biorthogonalize" the top and bottom MPSs comprised of MPS tensors A_U and A_D. Using $A_{U/D}$, we find P_L, P_L^- along with a new set of C'_{LU}, C'_{DL} satisfying:



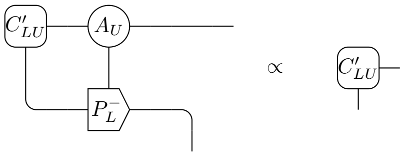
(38)



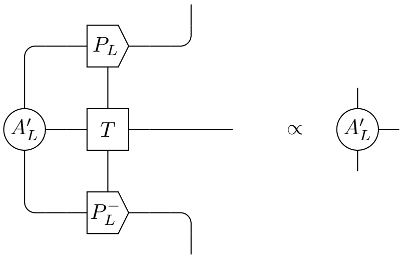
(39)

where P_L, P_L^- satisfy Eq. (24). There are multiple possible methods for finding tensors P_L, P_L^- and C'_{LU}, C'_{DL} that satisfy Eq. (38)–(39), and the choices are not unique. The method we use is described in detail in Appendix B 2.

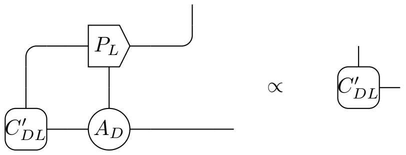
2. Obtain the left HRTM and CTMs using the gauged MPS tensors P_L, P_L^- found in step 1. This is done by numerically solving the following fixed point equations (in practice using an iterative method such as Arnoldi):



(40)



(41)



(42)

(Eq. (40) and (42) may be redundant, depending on what method for obtaining P_L, P_L^- is used).

Steps 1–2 constitute the left move of the asymmetric FPCM algorithm. For a single step of FPCM, the lattice is rotated, and the other directional moves are performed. For example, one could follow a conventional ordering of the directional CTMRG and next perform the up move, then the right move, and then the down move. In practice, we don't find that the ordering makes a noticeable difference in the performance of the algorithm. Note that a similar set of fixed point equations for the CTMs, HRTMs, and HCTMs were discussed by Baxter in the context of his CTM method[5].

The CTMs can be obtained in an alternative way from the corner transfer fixed point equations show in Fig. 1 (which are generalizations of Eq. (36), the symmetric corner transfer fixed point equation). We find that obtaining the CTMs with these equations leads to a more accurate environment than those found in Eq. (40) and (42), but are of course more computationally expensive. Finding more accurate CTMs is important for calculating accurate observables. In addition, like in the symmetric FPCM case, we find that in practice the accuracy of this procedure is improved by alternating between steps of FPCM and CTMRG, which we will discuss more in Section IV. In that case, it is particularly important to get the most accurate CTMs possible by solving for them with the fixed point equation in Fig. 1.

The algorithm looks very similar to the VUMPS algorithm when the network is Hermitian about a certain

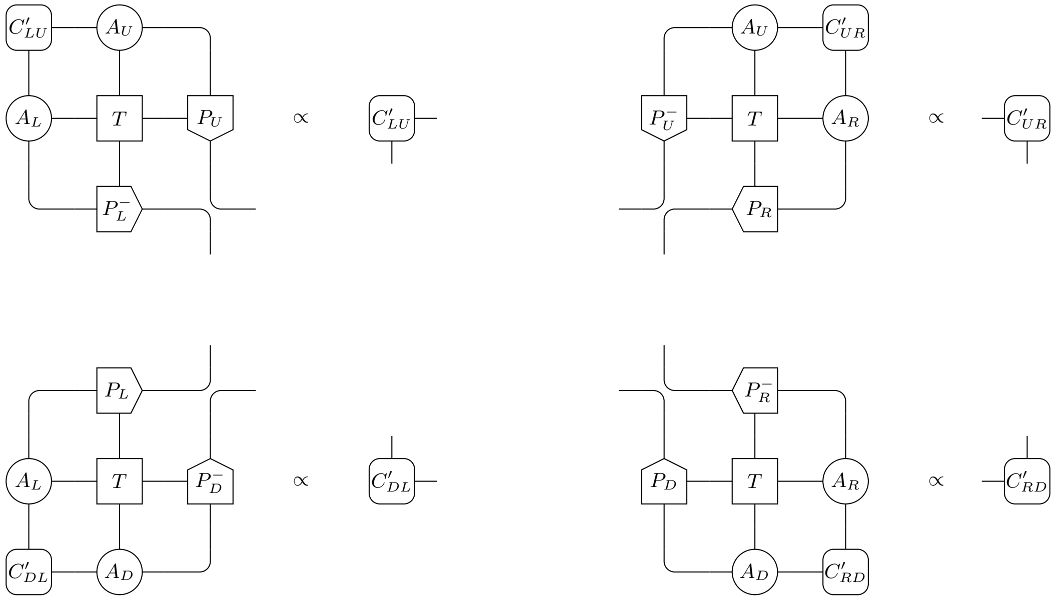


Figure 1. Fixed point equations for the CTMs.

direction (horizontal or vertical), in which case a pair of tensors P_L, P_L^- can be chosen to be isometric. However, like in CTMRG, the CTMs are used explicitly, not the center matrix of VUMPS/iDMRG, and the corners can be seen roughly as “square roots” of the center matrix. This is discussed in more detail in Ref. [1].

The leading cost of this algorithm, the calculation of the new boundaries, is $O(\chi^3 d^2)$ where χ is the bond dimension of the boundary, and d is the bond dimension of the network (assuming the fixed point is calculated in a sparse way with an iterative method such as Arnoldi and for simplicity assuming a large χ limit). This is the same leading cost as single-site VUMPS or single-site iDMRG. The cost of CTMRG, following the most standard schemes, is generally a full eigendecomposition, singular value decomposition, or QR decomposition of some part of the grown boundary. Since the boundary is grown from a bond dimension χ to a bond dimension χd, these decompositions lead to a scaling of the algorithm of $O(\chi^3 d^3)$, so asymptotically both (single site) VUMPS and our new corner method scale better than traditional CTMRG in the network bond dimension.

Even so, each step of traditional CTMRG can be much faster than the new schemes presented, because of the fixed points that we must calculate (note that avoiding the use of fixed point equations was one of the original motivations for the development of CTMRG as an alternative to DMRG[7,8]). However, we will see in Section IV that solving for the environment tensors with fixed point equations leads to a large speedup in total convergence time, because substantially fewer steps are needed for convergence.

The speedup of VUMPS and FPCM over CTMRG is particularly pronounced for networks with small gaps. One way to understand this is that the original CTMRG can be viewed as a power method, where only a single (or pair of) row-to-row and/or column-to-column transfer matrices are absorbed into the environment at a time, and the projectors are only determined in a local way. The new schemes properly exploit the translational invariance of the system, and iterative methods such as Arnoldi are known to be much faster than power methods for finding eigenvectors of matrices with small gaps (and the gaps of the transfer matrices are expected to be related to the gap of the system[67]). In addition, the projectors that are used for renormalization in the FPCM are obtained from the current guess for the entire (translationally invariant) boundary, not just a set of local tensors.

IV. RESULTS

Here, we present benchmark results for the methods described in the previous section: CTMRG, transfer matrix VUMPS, and the new fixed point corner method (FPCM). We benchmark the 2D classical ferromagnetic Ising model in Section IV A, the 2D classical XY model in Section IV B, the 2D quantum spin-1/2 Heisenberg model in Section IV C, and the chiral Resonating Valence Bond (RVB) PEPS in Section IV D.

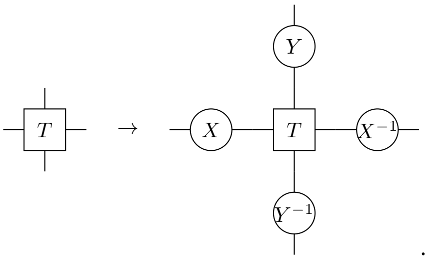
For all of the examples shown, the networks are on the square lattice and have a single-site unit cell, and all tensors used are dense. Calculations were performed with a single BLAS thread. To obtain a consistent comparison between different methods, the starting boundary states are chosen to be small (usually with bond dimension 2), the methods are run until convergence with the small bond dimension, and then the bond dimension is increased to the final one (CTMRG is used to grow the bond dimension for the FPCM, and the bond dimension growth scheme introduced in Ref. 54 is used for VUMPS). Most of the calculations were performed using the Extreme Science and Engineering Discovery Environment (XSEDE)[68] with Intel Math Kernel Library (MKL), except calculations in Fig. 3, which were performed on a laptop with OpenBLAS. Fixed points are calculated using the Arnoldi method as implemented in ARPACK.

A. 2D classical Ising model

In Fig. 2, we present benchmark results for the isotropic 2D ferromagnetic classical Ising model. The MPO comprising the partition function for this model has a link bond dimension of $d = 2$, and the tensor can be taken to be real and symmetric about all rotations and reflections. The environment tensors we use for all methods are restricted to being real. For CTMRG and the FPCM, in the ansatz in Eq. (16), we impose $A_U = A_R = A_D = A_L \equiv A$ and $C_{LU} = C_{UR} = C_{RD} = C_{DL} \equiv C$, and additionally impose $A^s = (A^s)^T$ and $C = C^T$. For VUMPS, in Eq. (7), we impose $[A_U^R]^s = ([A_U^L]^s)^T$ and $C_U = C_U^T$. Additionally, when we calculate observables, we set the bottom fixed point MPS equal to the top fixed point MPS. For CTMRG, we find the projector to renormalize the boundary using a symmetric eigendecomposition, which is fast and numerically very stable.

From Fig. 2, we see that as we approach the critical point of the 2D classical Ising model, the performance improvement of VUMPS and our new fixed point corner method (FPCM) over CTMRG increases. This can be understood by the fact that the boundary tensors for the new methods are obtained by solving fixed point equations (in practice with Arnoldi and Lanczos methods), which are known to be faster than power methods for finding extremal eigenvectors of matrices with small gaps. This indicates that these new methods are better suited for studying systems close to or at criticality, e.g. in combination with the theory of "finite entanglement scaling"[69–72].

In Fig. 3, we present results for the 2D ferromagnetic classical Ising model in a non-unitary basis. In other words, we introduce the following "gauge transformations" on the links of the tensor network:



. (43)

These gauge transformations, for random complex non-unitary matrices X and Y, artificially break the rotation and reflection symmetries of the Ising model partition function. Gauge transformations like these can be introduced during a PEPS optimization if explicit symmetries are not enforced, even if the state being targeted is expected to be rotationally symmetric. The environments we use for all methods are complex.

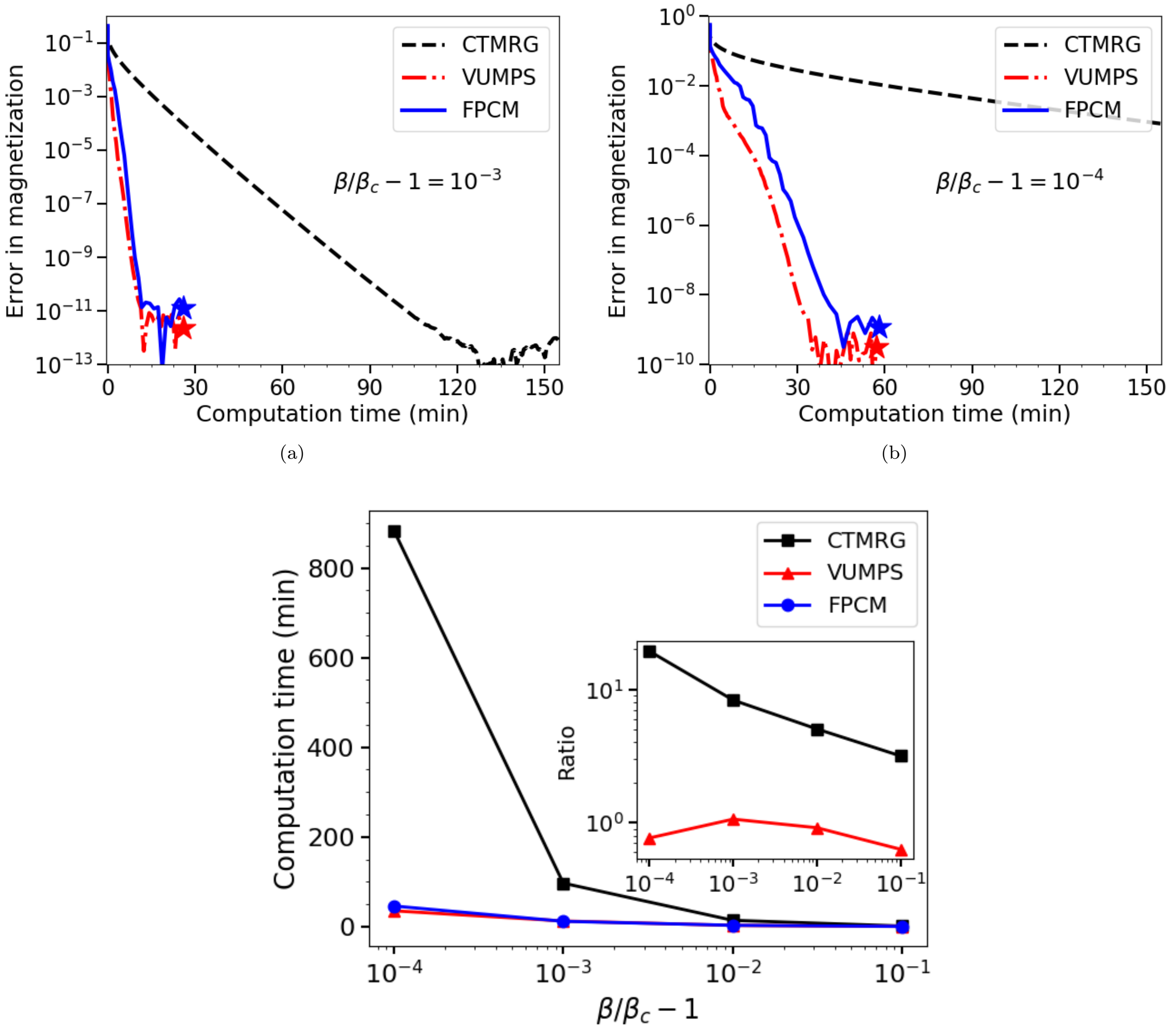
In Fig. 3(a), we show results for the asymmetric CTMRG method proposed by Corboz et al. in Ref. 51 and reviewed in Section III B 2. In Fig. 3(b), we show results for performing a step of the FPCM introduced in Section III C 2 every five steps of the CTMRG method of Corboz et al., which leads to substantially improvements in convergence time. Here, for the FPCM steps we use a pseudoinverse cutoff of 1×10^{-7} to ensure stability of the algorithm, which we find is only necessary in early steps when the bond dimension is being increased. Note that to obtain full precision for the fixed point environments for this example, it was important to calculate SVD in the biorthogonalization step of CTMRG with the LAPACK routine gesvd as opposed to the routine gesdd. We discuss this point and other details about the stability of the CTMRG method for this example in Appendix A.

Note that we found that the asymmetric FPCM method as presented in Section III C 2 alone has a tendency to "get stuck," i.e., not find the most accurate fixed point environment for a given bond dimension. It is possible that a modification of the method itself can fix this problem, but we find that as presented, the method is very simple and numerically stable, and combining with CTMRG is very effective and robust. We speculate about the reason why combining FPCM with CTMRG may help improve the accuracy of the fixed point found with FPCM at the end of Appendix B 2.

We also note that the asymmetric versions of CTMRG and FPCM are more computationally demanding and the convergence in general is not as good as the symmetric versions of the algorithms. Improving these methods by finding better techniques for obtaining the projectors is an interesting area of further research.

B. 2D classical XY model

In Fig. 4(a), we present results for contracting the partition function for the 2D classical XY model. Because the lattice degree of freedom is continuous for this model, the MPO tensor comprising the partition function can only be constructed approximately, though to high accuracy. The XY model has been studied previously with transfer matrix DMRG (TMRG)[73]. Additionally, related models have been studied previously with CTMRG[74,75]. The construction we use for the partition function is described in Ref. 76 and 77, where HOTRG was used to contract the partition function, and we refer readers to those references for details on constructing the MPO for this model. We use an inverse critical temperature 10% below the critical point estimated in that reference, and use an approximation for the MPO with a link bond dimension of $d = 25$. We use zero applied magnetic field,



(a)

(b)

(c)

Figure 2. Plots (a) and (b) show the error in the magnetization for the isotropic 2D classical Ising model as a function of computation time at two temperatures near criticality, where (b) is closer to criticality than (a). The network has a bond dimension of $d = 2$, and a boundary MPS bond dimension of $\chi = 600$ is used. A fully symmetric CTM ansatz is used for CTMRG and the FPCM, and full symmetry is exploited in VUMPS. The speedup of VUMPS and the corner method over CTMRG increases as one gets closer to criticality. Stars indicate the environment tensors have reached a fixed point, and data points beyond those points are numerical fluctuations and were not shown in order to simplify the plot. Plot (c) shows convergence time as a function of inverse temperature above criticality, $\beta/\beta_c - 1$, for the 2D classical Ising model. For all data points, a boundary MPS bond dimension of $\chi = 600$ is used. All data is converged to an error in the magnetization of $< 2 \times 10^{-9}$. The inset shows the ratio of the convergence time of CTMRG and VUMPS with respect to the FPCM convergence time (note the log scale).

and at this temperature the model is expected to be gapped. Since the U(1) symmetry cannot be broken at any finite temperature, we expect the magnetization to be zero.

The MPO tensor comprising the partition function is real and symmetric about reflections about the diagonals of the network, but not symmetric about the x and y axes. The environment we use for all methods is restricted to being real. For CTMRG and the FPCM, in the ansatz in Eq. (16), we impose $A_U = A_R^T = A_D = A_L^T \equiv A$, $C_{LU} = C_{UR} \equiv C$, and $C_{DL} = C_{RD} \equiv D$, and additionally impose $C = C^T$ and $D = D^T$. For VUMPS,

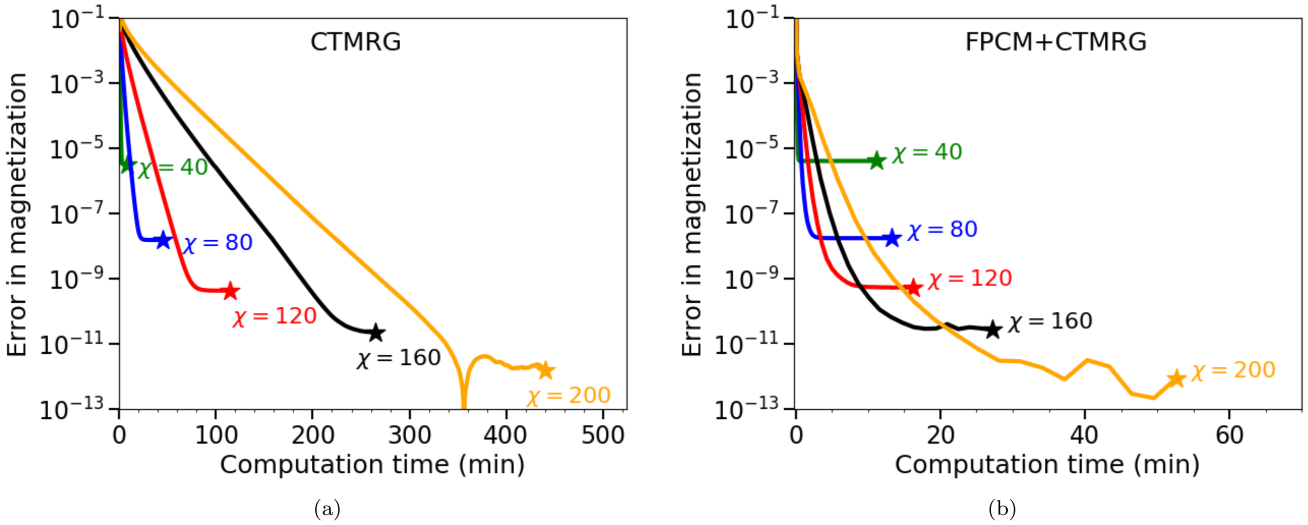


Figure 3. Plots of error in magnetization for the isotropic ferromagnetic 2D classical Ising model at $\beta/\beta_c - 1 = 10^{-3}$ with random non-unitary gauge transformations introduced on the horizontal and vertical links, as shown in Eq. (43). This artificially breaks the lattice symmetry in order to test each method on an asymmetric network. Plot (a) shows results for the asymmetric CTMRG algorithm by Corboz et al. in Ref. 51 and plot (b) shows results for the FPCM introduced in this work combined with the CTMRG algorithm used in (a).

in Eq. (7) we don't impose any symmetries, but when we calculate observables, we set the bottom fixed point MPS equal to the transpose of the top fixed point MPS (such that the environment is invariant under a rotation by π).

For CTMRG, we obtain the projectors using a symmetric diagonalization of the grown corner. For FPCM, we obtain the fixed point projectors by isometrically gauging the boundary MPS. Additionally, like for the asymmetric FPCM calculation performed in Section IV A, we find it is best to alternate between steps of FPCM and CTMRG instead of performing FPCM alone, and the results shown are obtained by performing a few steps of CTMRG per step of FPCM. We see that VUMPS performs noticeably worse than the FPCM, likely because the ansatz we use for the VUMPS calculation does not exploit the lattice symmetry as well as the CTM ansatz. Like with the Ising model, we expect the improvement of the FPCM compared to CTMRG to become even more pronounced closer to the critical point.

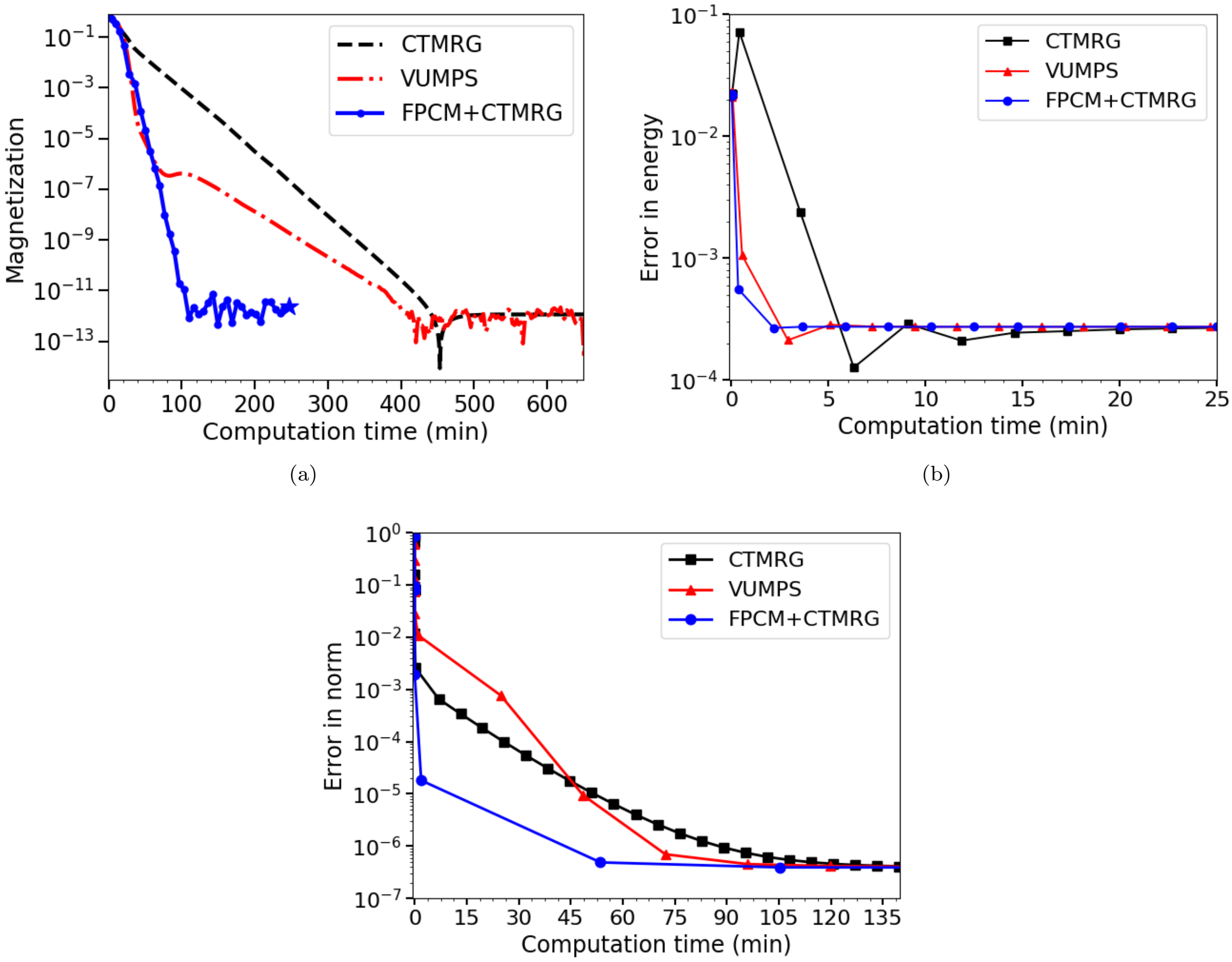
C. 2D quantum Heisenberg model

In Fig. 4(b), we present results for contracting a PEPS approximation ($D = \sqrt{d} = 5$) to the ground state of the 2D quantum Heisenberg model. The PEPS tensor was optimized using the conjugate gradient method described in Ref. 44. We plot the error in the energy relative to the energy obtained from Monte Carlo simulations[78].

The PEPS tensor is complex and symmetric (not Hermitian) about all rotations and reflections, which was a symmetry imposed in the optimization. Therefore, the MPO tensor that comprises the tensor network for the norm of the PEPS is also complex and symmetric about all rotations and reflections. The environments we use for all methods are necessarily complex. For CTMRG and the FPCM, in the ansatz in Eq. (16), we impose $A_U = A_R = A_D = A_L \equiv A$, $C_{LU} = C_{UR} = C_{RD} = C_{DL} \equiv C$, and additionally impose $C = C^T$. For VUMPS, in Eq. (7) we don't impose any symmetries[79], but when we calculate observables, we set the bottom fixed point MPS equal to the top fixed point MPS (not the conjugate of the top fixed point, as we would do if the MPO was Hermitian as opposed to complex symmetric). The CTMRG algorithm we use is a modification of the one from Ref. 51, where the symmetry of the network is exploited wherever possible. The FPCM method we use is a modification of the asymmetric version presented in Section III C 2 where the symmetry of the network is exploited wherever possible. Additionally, we use a modification of the uniform MPS biorthogonalization procedure in Appendix B 2, where we first gauge the MPSs isometrically before we biorthogonalize them. As previously mentioned in Sections IV A–IV B, we find for the FPCM that it is best to perform a few steps of CTMRG per step of the FPCM, which we find improves the accuracy of the fixed point environment.

D. Chiral resonating valence bond PEPS

In Fig. 4(c), we present results for contracting a chiral resonating valence bond (RVB) PEPS. The chiral RVB PEPS state was introduced as a chiral extension of the traditional nearest neighbor RVB PEPS[80,81]. As in the



(a)

(b)

(c)

Figure 4. (a) Plot of magnetization for the 2D classical XY model, for network bond dimension $d = 25$ and boundary MPS bond dimension $\chi = 50$. (b) Plot of error in energy (compared to Monte Carlo results) for the 2D quantum Heisenberg model. The network bond dimension is $d = 25$ (or PEPS bond dimension $D = \sqrt{d} = 5$), and the MPS boundary bond dimension $\chi = 100$. (c) Plot of error in the norm (where the "exact" results is taken to be an extrapolation of the norm in the limit of a large environment bond dimension) of the chiral RVB PEPS. The network bond dimension is $d = 9$ (or PEPS bond dimension $D = \sqrt{d} = 3$), and the boundary MPS bond dimension is $\chi = 800$.

previous works on this model, we choose $\lambda_1 = \lambda_2 = \lambda_{\text{chiral}} = 1$, where $\lambda_2 = \lambda_{\text{chiral}} = 0$ would correspond to the non-chiral nearest neighbor RVB state. We refer readers to those previous works on this model for details on its derivation and physics.

The PEPS tensor (and therefore double layer MPO tensor) for this model is complex and Hermitian about the horizontal, vertical and diagonal reflections of the lattice. For CTMRG and the FPCM, in the ansatz in Eq. (16), we impose $A_U = A_R = A_D = A_L \equiv A$ and $C_{LU} = C_{UR} = C_{RD} = C_{DL} \equiv C$, and additionally impose $A^s = (A^s)^\dagger$ and $C = C^\dagger$. For VUMPS, in Eq. (7) we do not impose any symmetries[82]. When we calculate observables, we set the bottom fixed point MPS obtained from VUMPS equal to the complex conjugate of the top fixed point MPS. For CTMRG, the projectors are obtained with a Hermitian diagonalization of the grown corner, and for FPCM, the fixed point projectors are obtained by isometrically gauging the boundary MPS.

Again, we see an improvement in performance of the FPCM and VUMPS over CTMRG, but the FPCM performs better than VUMPS (we believe for this case because the symmetry of the network is exploited better in the CTM ansatz). Again, we perform a few steps of CTMRG per step of the FPCM, which we find improves the convergence time.

V. CONCLUSION AND OUTLOOK

We presented two new approaches for contracting infinite 2D tensor networks, such as 2D classical partition functions and 2D quantum states represented as a PEPS. One approach uses the recently proposed VUMPS algorithm to obtain boundary MPSs that approximate the infinite environment of the tensor network. The other approach uses the CTM ansatz like CTMRG, but improves upon CTMRG by solving for the boundary tensors with fixed point equations, which we refer to as the fixed point corner method (FPCM). With careful benchmarking, we compared these new approaches to CTMRG for a variety of systems, which is currently the most widely used method for contracting 2D tensor networks in infinite PEPS calculations. We found that both methods improve upon the performance of CTMRG, though for certain models, the improvement is more pronounced for FPCM as opposed to VUMPS.

We showed that the improvement upon CTMRG is particularly pronounced as models approach criticality, as exemplified by our benchmarking of the 2D classical Ising model. This can be explained by the fact that, as the gap of the model closes, so too does the gap of the transfer matrix. By solving for the boundary tensors with fixed point equations, methods such as Arnoldi and Lanzos can be used, which are known to perform better than power methods for finding extremal eigenvectors of matrices with small gaps. Even though each step of the new approaches we present can be slower than each step of CTMRG, substantially fewer steps are required to reach fixed points leading to an overall improvement in the performance.

We are convinced that these new methods directly improve the performance of current state of the art infinite PEPS optimization techniques, where the contraction of the network is the most computationally expensive step. When combined with recently introduced variational methods for optimizing PEPS[43,44], we expect that significant improvements can still be made to existing PEPS algorithms.

ACKNOWLEDGMENTS

M.F. acknowledges helpful feedback from P.R. Corboz and T. Nishino. M.F. would also like to thank S.R. White and E.M. Stoudenmire for useful input on the presentation of the results. The authors gratefully acknowledge support from the National Science Foundation Graduate Research Fellowship Program (NSF GRFP) under Grant No. DGE-1144469 (M.F.), the Austrian Science Fund (FWF): F4104 SFB ViCoM and F4014 SFB FoQuS (V.Z.-S. and F.V.), and the European Research Council (ERC) under Grant No. 715861 (J.H.). J.H. and L.V. are supported by the Research Foundation Flanders (FWO). This project has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant agreement No 647905). This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1548562.

[1] J. Haegeman and F. Verstraete, Annual Review of Condensed Matter Physics **8**, 355 (2017).

[2] H. A. Kramers and G. H. Wannier, Phys. Rev. **60**, 263 (1941).

[3] R. J. Baxter, Journal of Mathematical Physics **9**, 650 (1968).

[4] R. J. Baxter, Journal of Statistical Physics **19**, 461 (1978).

[5] R. J. Baxter, *Exactly solved models in statistical mechanics* (Academic Press, London, 1982) Chap. 13.

[6] T. Nishino, Journal of the Physical Society of Japan **64**, 3598 (1995).

[7] T. Nishino and K. Okunishi, Journal of the Physical Society of Japan **65**, 891 (1996).

[8] T. Nishino and K. Okunishi, Journal of the Physical Society of Japan **66**, 3040 (1997).

[9] H. F. Trotter, Proceedings of the American Mathematical Society **10**, 545 (1959).

[10] M. Suzuki, Communications in Mathematical Physics **51**, 183 (1976).

[11] M. Suzuki, Phys. Rev. B **31**, 2957 (1985).

[12] M. Suzuki, Physics Letters A **146**, 319 (1990).

[13] X. Wang and T. Xiang, Phys. Rev. B **56**, 5061 (1997).

[14] M. Suzuki, Journal of Mathematical Physics **32**, 400 (1991).

[15] R. J. Bursill, T. Xiang, and G. A. Gehring, Journal of Physics: Condensed Matter **8**, L583 (1996).

[16] G. Vidal, Phys. Rev. Lett. **91**, 147902 (2003).

[17] G. Vidal, Phys. Rev. Lett. **98**, 070201 (2007).

[18] T. Nishino, K. Okunishi, Y. Hieida, N. Maeshima, and Y. Akutsu, Nuclear Physics B **575**, 504 (2000).

[19] T. Nishino, Y. Hieida, K. Okunishi, N. Maeshima, Y. Akutsu, and A. Gendiar, Progress of Theoretical Physics **105**, 409 (2001).

[20] A. Gendiar, N. Maeshima, and T. Nishino, Progress of Theoretical Physics **110**, 691 (2003).

[21] Y. Nishio, N. Maeshima, A. Gendiar, and T. Nishino, “Tensor Product Variational Formulation for Quantum Systems,” arXiv:cond-mat/0401115.

[22] F. Verstraete and J. J. I. Cirac, “Renormalization algorithms for Quantum-Many Body Systems in two and higher dimensions,” arXiv:cond-mat/0407066.

[23] S. R. White, Phys. Rev. Lett. **69**, 2863 (1992).

[24] S. R. White, Phys. Rev. B **48**, 10345 (1993).

[25] T. Nishino and K. Okunishi, Journal of the Physical Society of Japan **64**, 4084 (1995).

[26] N. Shibata, Journal of the Physical Society of Japan **66**, 2221 (1997).

[27] M. Levin and C. P. Nave, Phys. Rev. Lett. **99**, 120601 (2007).

[28] R. Orús and G. Vidal, Phys. Rev. B **78**, 155117 (2008).

[29] I. P. McCulloch, “Infinite size density matrix renormalization group, revisited,” arXiv:0804.2509.

[30] Z. Y. Xie, H. C. Jiang, Q. N. Chen, Z. Y. Weng, and T. Xiang, Phys. Rev. Lett. **103**, 160601 (2009).

[31] Y.-K. Huang, Phys. Rev. E **83**, 036702 (2011).

[32] Y.-K. Huang, Journal of Statistical Mechanics: Theory and Experiment **2011**, P07003 (2011).

[33] Z. Y. Xie, J. Chen, M. P. Qin, J. W. Zhu, L. P. Yang, and T. Xiang, Phys. Rev. B **86**, 045139 (2012).

[34] G. Evenbly and G. Vidal, Phys. Rev. Lett. **115**, 180405 (2015).

[35] S. Yang, Z.-C. Gu, and X.-G. Wen, Phys. Rev. Lett. **118**, 110504 (2017).

[36] M. Bal, M. Mariën, J. Haegeman, and F. Verstraete, Phys. Rev. Lett. **118**, 250602 (2017).

[37] M. P. Zaletel, R. S. K. Mong, C. Karrasch, J. E. Moore, and F. Pollmann, Phys. Rev. B **91**, 165112 (2015).

[38] J. Haegeman, J. I. Cirac, T. J. Osborne, I. Pižorn, H. Verschelde, and F. Verstraete, Phys. Rev. Lett. **107**, 070601 (2011).

[39] J. Haegeman, C. Lubich, I. Oseledets, B. Vandereycken, and F. Verstraete, Phys. Rev. B **94**, 165116 (2016).

[40] J. Jordan, R. Orús, G. Vidal, F. Verstraete, and J. I. Cirac, Phys. Rev. Lett. **101**, 250602 (2008).

[41] J. Jordan, R. Orús, and G. Vidal, Phys. Rev. B **79**, 174515 (2009).

[42] L. Vanderstraeten, M. Mariën, F. Verstraete, and J. Haegeman, Phys. Rev. B **92**, 201111 (2015).

[43] P. Corboz, Phys. Rev. B **94**, 035133 (2016).

[44] L. Vanderstraeten, J. Haegeman, P. Corboz, and F. Verstraete, Phys. Rev. B **94**, 155123 (2016).

[45] N. Maeshima, Y. Hieida, Y. Akutsu, T. Nishino, and K. Okunishi, Phys. Rev. E **64**, 016705 (2001).

[46] R. Orús and G. Vidal, Phys. Rev. B **80**, 094403 (2009).

[47] P. Corboz, R. Orús, B. Bauer, and G. Vidal, Phys. Rev. B **81**, 165104 (2010).

[48] P. Corboz, J. Jordan, and G. Vidal, Phys. Rev. B **82**, 245119 (2010).

[49] P. Corboz, S. R. White, G. Vidal, and M. Troyer, Phys. Rev. B **84**, 041108 (2011).

[50] R. Orús, Phys. Rev. B **85**, 205117 (2012).

[51] P. Corboz, T. M. Rice, and M. Troyer, Phys. Rev. Lett. **113**, 046402 (2014).

[52] H. N. Phien, J. A. Bengua, H. D. Tuan, P. Corboz, and R. Orús, Phys. Rev. B **92**, 035142 (2015).

[53] P. Corboz, Phys. Rev. B **93**, 045116 (2016).

[54] V. Zauner-Stauber, L. Vanderstraeten, M. T. Fishman, F. Verstraete, and J. Haegeman, Phys. Rev. B **97**, 045145 (2018).

[55] R. Orús, Annals of Physics **349**, 117 (2014).

[56] The PEPS tensors can of course be left uncontracted to allow for a more efficient ordering of contraction later on, but for now we will think of it as a single larger tensor.

[57] M. Hastings, J. Stat. Mech. Theor. Exp. **2007**, P08024 (2007).

[58] L. Michel and I. P. McCulloch, "Schur Forms of Matrix Product Operators in the Infinite Limit," arXiv:1008.4667.

[59] A. Kemper, A. Schadschneider, and J. Zittartz, Journal of Physics A: Mathematical and General **34**, L279 (2001).

[60] T. Enss and U. Schollwöck, Journal of Physics A: Mathematical and General **34**, 7769 (2001).

[61] G. K.-L. Chan and T. V. Voorhis, The Journal of Chemical Physics **122**, 204101 (2005).

[62] U. Schollwöck, Rev. Mod. Phys. **77**, 259 (2005).

[63] Y.-K. Huang, P. Chen, and Y.-J. Kao, Phys. Rev. B **86**, 235102 (2012).

[64] Note that this is equivalent to the MPO tensor being invariant under $\pi/2$ rotations.

[65] Note that we use the notation A^s to denote the matrix obtained by setting the physical index of MPS tensor A to s.

[66] The original proposal actually involved a symmetric eigendecomposition of a product of four of the grown corners in Eq. (18) which has the interpretation of a density matrix, but the eigenbasis is the same as that of a single corner.

[67] V. Zauner, D. Draxler, L. Vanderstraeten, M. Degroote, J. Haegeman, M. Rams, V. Stojevic, N. Schuch, and F. Verstraete, New J. Phys. **17**, 053002 (2015).

[68] J. Towns, T. Cockerill, M. Dahan, I. Foster, K. Gaither, A. Grimshaw, V. Hazlewood, S. Lathrop, D. Lifka, G. D. Peterson, R. Roskies, J. R. Scott, and N. Wilkins-Diehr, Computing in Science & Engineering **16**, 62 (2014).

[69] L. Tagliacozzo, T. R. de Oliveira, S. Iblisdir, and J. I. Latorre, Physical review b **78**, 024410 (2008).

[70] F. Pollmann, S. Mukerjee, A. M. Turner, and J. E. Moore, Physical review letters **102**, 255701 (2009).

[71] B. Pirvu, G. Vidal, F. Verstraete, and L. Tagliacozzo, Physical review b **86**, 075117 (2012).

[72] V. Stojevic, J. Haegeman, I. P. McCulloch, L. Tagliacozzo, and F. Verstraete, Physical Review B **91**, 035120 (2015).

[73] S. G. Chung, Phys. Rev. B **60**, 11761 (1999).

[74] D. P. Foster and C. Pinettes, Phys. Rev. E **67**, 045105 (2003).

[75] D. P. Foster and C. Pinettes, Journal of Physics A: Mathematical and General **36**, 10279 (2003).

[76] Y. Liu, Y. Meurice, M. P. Qin, J. Unmuth-Yockey, T. Xiang, Z. Y. Xie, J. F. Yu, and H. Zou, Phys. Rev. D **88**, 056005 (2013).

[77] J. F. Yu, Z. Y. Xie, Y. Meurice, Y. Liu, A. Denbleyker, H. Zou, M. P. Qin, J. Chen, and T. Xiang, Phys. Rev. E **89**, 013308 (2014).

[78] A. W. Sandvik, Phys. Rev. B **56**, 11678 (1997).

[79] One may expect that we could set $[A_U^R]^s = ([A_U^L]^s)^T$ and $C_U = C_U^T$. However, we were unable to get VUMPS to converge with these constraints imposed, and it would likely require a nontrivial modifications of the VUMPS algorithm.

[80] D. Poilblanc, J. I. Cirac, and N. Schuch, Phys. Rev. B **91**, 224431 (2015).

[81] D. Poilblanc, N. Schuch, and I. Affleck, Phys. Rev. B **93**, 174414 (2016).

[82] One may expect that we could set $[A_U^R]^s = ([A_U^L]^s)^\dagger$ and $C_U = C_U^\dagger$. However, we found in practice these relations only held up to diagonal phases. This could possibly be fixed by some modification of the VUMPS algorithm.

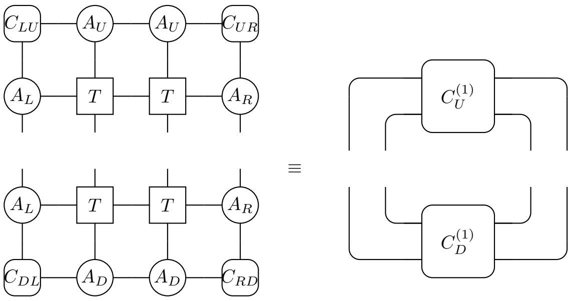
Appendix A: Some comments on the numerical stability of the asymmetric CTMRG method

In this section, we discuss the stability of the CTMRG algorithm introduced in Ref. 51 (and reviewed in this work in Section III B 2). We also discuss slight alternatives to that method and discuss their relevance for the efficiency and stability of the asymmetric CTMRG algorithm.

We begin by pointing out an equivalent form of the algorithm of Ref. 51:

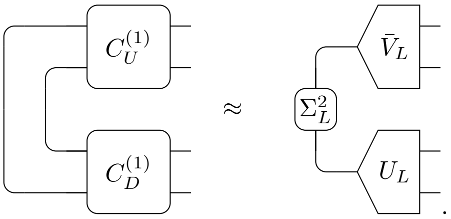
1. We first define the half system transfer matrices

$C_U^{(1)}$ and $C_D^{(1)}$:



(A1)

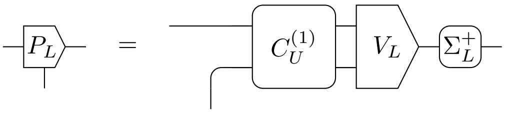
To obtain the projectors, we use the same biorthogonalization procedure as was used in Eq. (29)–(31). We take the following SVD of $C_D^{(1)}C_U^{(1)}$:



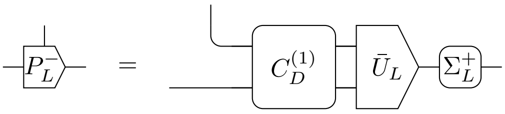
(A2)

In other words, we take the SVD $C_D^{(1)}C_U^{(1)} = U_L\Sigma_L^2 V_L^\dagger$. Eq. (A2) is approximate because we truncate according to the singular values down to the desired bond dimension for the renormalized environment.

2. Now we obtain P_L, P_L^- as follows:



(A3)



(A4)

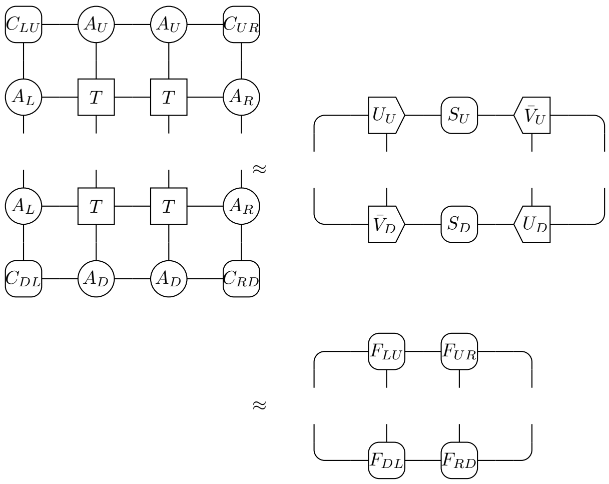
This alternative for the left move obtains the same tensors P_L,P_L^- but skips the step of calculating the QR decomposition of $C_U^{(1)}$, $C_D^{(1)}$, making it computationally more efficient.

For both the method of Ref. 51 and the equivalent form above, it may be important to perform a pseudoinverse of the (sqaure root) of the singular values obtained in the biorthogonalization. Because of the (pseudo)inverse, it is important to calculate the singular values to high accuracy. However, this can become challenging because to obtain an environment with high accuracy, a large bond dimension must be used and therefore small singular values will appear. In practice, we found that a higher precision for the environment could be obtained by calculating the SVD with the LAPACK routine gesvd as opposed to the routine gesdd.

A comparison of the fixed point spectrum that needs to be inverted during the asymmetric CTMRG method of Ref. 51 when the SVDs are performed with gesvd and gesdd is shown in Fig. 5. In that figure, we plot $\Sigma_{L,i}$, the diagonal entries of Σ_L calculated in Eq. (29) or Eq. (A2), for cases when CTMRG is run using either gesvd or gesdd. The spectrums are calculated from fixed point environments of the 2D classical Ising model with a non-unitary change of basis (as described in Section IV A) with a bond dimension for the environment of $\chi = 160$. In practice, we found that using gesdd to calculate the SVD could lead to a less precise fixed point environment when many small singular values were present, because the small singular values were not calculated accurately and therefore a pseudoinverse cutoff (i.e. 5×10^{-8}) was required to ensure the CTMRG algorithm was numerically stable.

Alternative methods can be used to improve the conditioning of the matrix that must be inverted during the biorthogonalization procedure. If the conditioning is improved, the biorthogonalization is less sensitive to the way that the SVD is computed. One method was mentioned in Ref. 51, which is to biorthogonalize the grown CTMs of the environment (as opposed to the half system transfer matrices). As mentioned in Ref. 51, this method is not as accurate as using the half system transfer matrices. An alternative method to obtain the CTMRG projectors which is as accurate as using the half system transfer matrices but improves the conditioning of the inverse required in the biorthogonalization is as follows:

1. We start by taking the SVD of the half system transfer matrices:

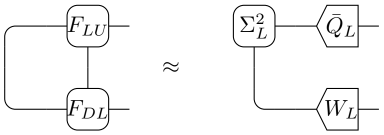


(A5)

where we define $F_{LU}^s \equiv U_U^s S_U^{1/2}$, $F_{UR}^s \equiv S_U^{1/2}(V_U^s)^\dagger$, $F_{RD}^s \equiv U_D^s S_D^{1/2}$, and $F_{DL}^s \equiv S_D^{1/2}(V_D^s)^\dagger$. Here, one can truncate these equations according to the singular values, which can improve the computational cost of the algorithm.

2. The next step is to obtain the tensors P_L, P_L^-. Again, we use the same biorthogonalization procedure as was used in Eq. (29)–(31). However, this

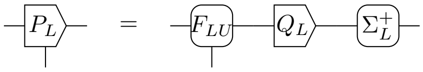
time we biorthogonalize the tensors F_{LU}, F_{DL}:



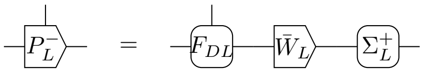
(A6)

where the second line is obtained by taking the SVD $F^s_{DL} F^s_{LU} = W_L \Sigma^2_L Q^\dagger_L$ and truncating according to the singular values to the desired bond dimension.

3. Again, following the biorthogonalization procedure, we obtain P_L, P_L^- as follows:



(A7)



(A8)

For stability of the algorithm, it may be necessary to set a pseudoinverse cutoff for Σ_L, where we use the notation Σ_L^+ to denote the pseudoinverse. We found this was particularly relevant in the early steps of the algorithm, during bond dimension growth.

We found it could be important to reorthogonalize the tensors P_L,P_L^- (i.e. repeat the biorthogonalization process on P_L,P_L^- a few times) in order to improve the accuracy of the projectors obtained. Note that with this new CTMRG method, it is natural to compute both the left and right projectors in a single step to save computation time (by directly biorthogonalizing the tensors F_{UR} and F_{RD} to obtain the right move projector).

The advantage of this alternative method for obtaining the projectors is that it improves the conditioning of the inverse required in the biorthogonalization, which in practice means the method is more numerically stable (not as sensitive to the choice of SVD algorithm that is used). The improvement of the conditioning is shown in Fig. 5. The new CTMRG method is more computationally expensive than the one from Ref. 51 (and the simplified version shown at the beginning of this section) because of the extra SVDs that are computed. Alternating between steps of this new asymmetric CTMRG method and steps of FPCM leads to similar speedups in convergence times as those shown in Section IV A.

Appendix B: Algorithms for gauging uniform MPSs

1. New algorithm for isometrically gauging a uniform MPS

Starting with a uniform MPS comprised of the MPS tensor A, we would like to find the gauge in which the left fixed point of the MPS transfer matrix is identity from bra to ket (the "canonical" gauge). In other words, we would like to find U and C which satisfy:

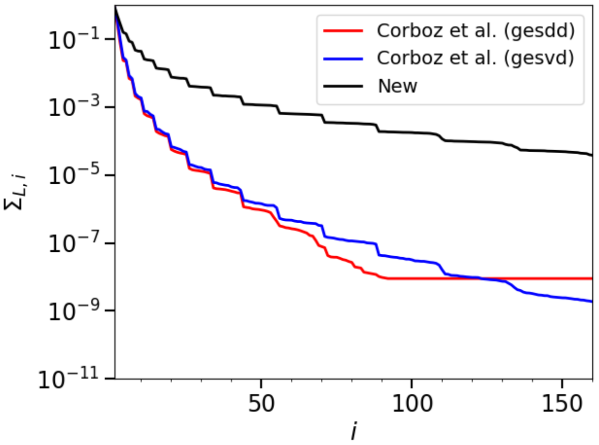
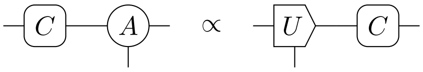
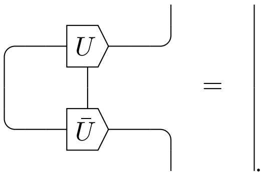


Figure 5. Comparison of spectrums that need to be inverted to perform the biorthogonalization in different versions of the asymmetric CTMRG algorithm. Note that what is plotted, $\Sigma_{L,i}$, are the square root of the singular values that are calculated during the biorthogonalization. See the main text for details.



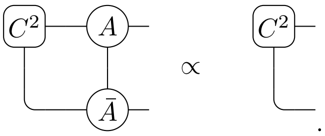
(B1)



(B2)

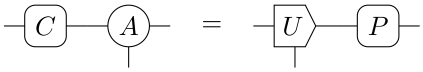
A method for finding the orthogonal gauge for a uniform MPS was first proposed in the context of the iTEBD algorithm[28] and involves a pseudoinverse of the matrix C to solve for U. Unfortunately, this means that U is generally only approximately isometric, and the accuracy up to which this "pulling through" equation can be satisfied may be limited for a MPS with small singular values.

We now present a fast, robust and highly accurate alternative, where U is constrained to be isometric and no explicit matrix inversions are used. Similar to previous methods, we start by finding the left fixed point which we suggestively call C^2 of the MPS transfer matrix:



(B3)

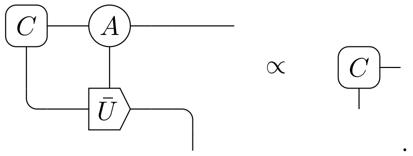
From properties of the transfer matrix, we know that C^2 is a positive Hermitian matrix (up to numerical errors). We obtain C by taking the square root of C^2 (for example by performing a Hermitian eigendecomposition of C^2 and taking the square roots of the positive eigenvalues). We now obtain our initial U by performing the following polar decomposition:



(B4)

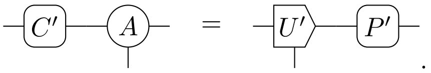
where U is read off as the isometry obtained from the polar decomposition, and P is the positive Hermitian matrix obtained from the polar decomposition. $|C - P|$ is taken to be our initial error in the gauging. Because we took a square root of C^2, our initial error may be limited to the square root of machine precision, i.e. $O(10^{-8})$. If higher precision is required, we repeat the following steps until convergence:

1. Get a new corner matrix C from the mixed transfer matrix of A and $\bar{U}$ by approximately solving for the leading eigenvector of the fixed point equation:



(B5)

2. Get a new U' using the new C from step 1. First, take the (left) polar decomposition of C to get $C = QC'$, where C' is positive and Hermitian. Then, obtain the new U' from a polar decomposition similar to before, i.e.,

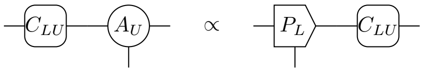


(B6)

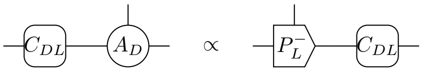
We measure the error of the current iteration as $|C' - P'|$, and repeat steps 1 and 2 until a desired tolerance is met.

2. New algorithm for "biorthogonalizing" two MPSs

We now describe how to "biorthogonalize" two uniform MPSs (with single-site unit cells) that are respectively comprised of MPS tensors A_U and A_D. By biorthogonalize, we mean that we wish to gauge transform A_U and A_D to gauges in which in one direction the fixed point of the mixed MPS transfer matrix formed from the two uniform MPSs is the identity matrix from bra to ket. In other words, we would like to gauge transform A_U and A_D so that they satisfy:

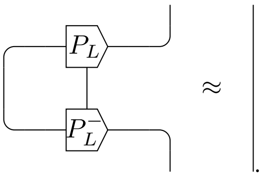


(B7)



(B8)

where P_L and P_L^- are tensors satisfying

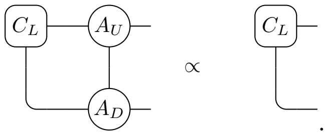


(B9)

If $A_D^s = (A_U^s)^\dagger$ (possibly after fixing a gauge degree of freedom), one can use the approach introduced in Appendix B 1, or use the iTEBD algorithm from Ref. 28. In that case, P_L and P_L^- can be chosen to be isometries, such that $[P_L^-]^s = (P_L^s)^\dagger$ and $(P_L^s)^\dagger P_L^s = I$.

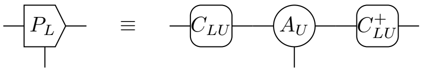
If A_D is not the conjugate of A_U, then in general P_L and P_L^- won't be isometries, and there are multiple possible approaches for satisfying Eq. (B7)–(B8). The approach we use is the following:

1. We start by getting the left fixed point C_L of the mixed transfer matrix of A_U and A_D:

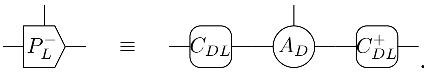


(B10)

2. We now take the SVD of $C_L = U_L \Sigma_L^2 V_L^\dagger$. We define $C_{LU} \equiv \Sigma_L V_L^\dagger$ and $C_{DL} \equiv U_L \Sigma_L$ and define P_L, P_L^- as follows:



(B11)



(B12)

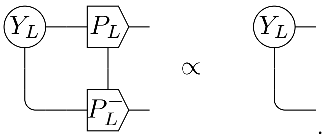
Here, $C_{LU}^+ = V_L \Sigma_L^+$ and $C_{DL}^+ = \Sigma_L^+ U_L^\dagger$, where Σ_L^+ denotes the pseudoinverse of Σ_L.

This procedure can be viewed as a simple fixed point formulation of biorthogonalization procedure introduced by Huang[31,32,63] in the context of non-Hermitian transfer matrix DMRG (TMRG) and originally applied to CTMRG by Corboz et al.[51].

In practice, we find that P_L, P_L^- from Eq. (B7)–(B8) may not satisfy Eq. (B9) well enough. If this is the case, we can perform a procedure that we refer to as "reorthogonalization." For reorthogonalizing P_L, P_L^- we perform the following steps (which are essentially just the steps listed above applied to biorthogonalizing P_L, P_L^-):

1. Calculate the dominant left fixed point of the mixed MPS transfer matrix of P_L, P_L^- calculated from

Eq. (B7)–(B8) above:



(B13)

2. Take the SVD of $Y_L = U\Sigma^2 V^\dagger$, defining $Y_{LU} = \Sigma V^\dagger$ and $Y_{DL} = U\Sigma$. Then, update $P_L^s \rightarrow Y_{LU} P_L^s Y_{LU}^+$, $[P_L^-]^s \rightarrow Y_{DL}^+ [P_L^-]^s Y_{DL}$, $C_{LU} \rightarrow Y_{LU} C_{LU}$, and $C_{DL} \rightarrow Y_{DL} C_{DL}$.

These steps can be repeated a number of times. Typically, a small number of repetitions (5 to 10) is advantageous for improving the accuracy of the biorthogonalization. It is interesting to point out that a similar concept of reorthogonalization is used in standard Krylov subspace methods.

Additionally, we will describe an alternative biorthogonalization method that we have tested with the asymmetric FPCM. One could first gauge transform the MPSs comprised of A_U and A_D isometrically from the left (for example using the method described in Appendix B 1) to obtain the isometries which we will call A_U^L and A_D^L. Then, the biorthogonalization procedure can be applied to the isometries A_U^L, A_D^L to obtain what would in general be a different set of P_L, P_L^- and C_{LU}, C_{DL} in Eq. (B11)–(B12). Although this seems to improve the conditioning of the inverse, we found that using this method in the FPCM led to numerical instabilities of the FPCM at larger bond dimensions that were not fixed by setting a pseudoinverse. However, we found that this isometric gauging can work when there are symmetry constraints between the top and bottom MPSs (for example in the Heisenberg model example in Section IV C). A similar instability was noticed by Huang in Ref. 31, where a subspace expansion technique was proposed in the context of TMRG to improve the stability. It would be interesting to see if an analgous subspace expansion could be used in the context of FPCM.

In general, it is clear that gauge transforming the MPSs comprised of A_U and A_D before performing the biorthogonalization can lead to different fixed point projectors P_L, P_L^-, which can affect the accuracy and stability of FPCM. As we have mentioned in the main text, in practice we found it worked well to alternate between steps of FPCM and CTMRG. We believe that the steps of CTMRG may help to find "good" gauges of the HRTMs, HCTMs and CTMs, leading to better projectors obtained from the biorthogonalization. For a more in-depth discussion of biorthogonalizing uniform MPSs and other possible strategies for fixing the gauge, we refer readers to Ref. 32.