

Development of Tensor Network Algorithms for Studying Classical and Quantum Many-Body Systems

Thesis by
Matthew Theodore Fishman

In Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy

The logo for the California Institute of Technology (Caltech), featuring the word "Caltech" in a bold, orange, sans-serif font.

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2018
Defended May 24, 2018

© 2018

Matthew Theodore Fishman

All rights reserved

ACKNOWLEDGEMENTS

First and foremost I would like to thank my parents, without whom none of this would be possible. Their commitment to my success in life is unwavering, and I owe them for everything I have.

Of course I would like to thank my advisors, Steve White and John Preskill, as well as the other members of my committee, Garnet Chan and Lesik Motrunich.

Steve took me under his wing when I was looking for research guidance. His kindness, generosity and patience are unmatched, and he has always been available to talk and to give insight and advice for any problem I might have. It was a great privilege and honor to work with him and observe his scientific process. He is able to pinpoint the exact source of a problem and immediately propose a clever and original solution. Working with and talking to him has been humbling, and has shaped the way I approach problems I encounter.

John creates an environment in his group for exploration, and this work would not have been possible without the freedom he allowed. It was inspiring to see the wide range of problems members of his group work on, and the surprising range of technical topics with which John could engage. John is known for asking deep and insightful questions on research topics that are not directly in his area of expertise, and his reputation preceded him. Throughout my PhD I had a small voice in the back of my head with questions John might ask me about my research, and that was a guiding force for me to ask deeper questions or change research directions.

I am grateful to Garnet for encouraging me to talk to Steve early on in my PhD. Garnet is an incredibly ambitious and energetic scientist, and it has been great to get to discuss research with him over the years.

I would like to thank Frank Verstraete, who kindly hosted me during my year abroad working with his group in Austria and Belgium and took me in as if I was one of his own students. Frank approaches both science and life like a force of nature, and his excitement, confidence and ingenuity are truly a sight to behold. I would also like to thank Jutho Haegeman for his role in mentoring me while I was visiting Frank. Jutho delves into incredible depth into every topic he comes into contact with, and it was a great opportunity to get to work with these two amazing scientists. They introduced me to their unique techniques and approaches which were the basis for my research at the end of my PhD.

I would particularly like to thank Glen Evenbly, who as a postdoc at Caltech in John's group introduced me to this field of research as well as to Steve. Glen approaches the messy world of computational physics with an unmatched clarity and care, and I can only hope to approach problems with his level of precision. While he was at Caltech and afterward as a postdoc at UCI with Steve, most of the research ideas I had passed by him, and I quickly learned to treat his answers to questions I had as the ultimate authority on the topic.

I am indebted to Micheal Beverland, a former graduate student of John's who convinced me to speak to Glen when I was looking for topics of research. His friendship and guidance throughout my PhD helped get me through some of my most difficult times here. He is wise beyond his years, and one of the most thoughtful people I have ever met. I would also like to thank Sam Johnson for her friendship and support when I was starting out here at Caltech.

I would like to thank Frank's students including but not limited to Valentin Stauber, Burak Sahinoglu, Matthias Bal, Laurens Vanderstraeten, etc. for their friendship and scientific collaborations while I was visiting them in Europe. I would also like to thank all of my friends in Austria and Belgium, who made me feel at home when I was away from my friends and family.

It was a privilege to have the opportunity to engage with so many scientists at Caltech, UCI, and elsewhere, such as Miles Stoudenmire, Mike Zaletel, Martin Ganahl, Ashley Milsted, Julian Rincon, Philippe Corboz, Chris White, Brenden Roberts, Guifre Vidal, Zhenyue Zhu, Chia-Min Chung, Thomas Baker, etc. I have found that the scientists in this field of research are passionate, curious and supportive, and I appreciate the chance I have had to get to interact with them.

Finally, I would like to express my appreciation for all of the friends I've made here at Caltech, including but not limited to, and in no particular order: Sarah, Will, Mickey, Marius, Thom, Lori, Teddy, Adina, Kat, Stefan, Vatsal, Dave, Kes, Elise, Nick, Cody, etc. I wouldn't be who I am today without them, and I wouldn't have wanted to spend the majority of my free time throughout this process with anyone else. I am convinced that they are among the smartest, funniest, weirdest, coolest and most adventurous people I will ever get a chance to meet. The idea that I will not get to interact with them on a regular basis is difficult for me to imagine, and I hope we will be able to remain friends for a long time to come, wherever we may end up.

ABSTRACT

The field of tensor networks, kicked off in 1992 by Steve White's invention of the spectacularly successful density matrix renormalization group (DMRG) algorithm, has exploded in popularity in recent years. Tensor networks are poised to play a role in helping us solve some of the greatest open physics problems of our time, such as understanding the nature of high-temperature superconductivity and illuminating a theory of quantum gravity. DMRG and extensions based on a class of variational states known as tensor network states have been indispensable tools in helping us understand both numerically and theoretically the properties of complicated classical and quantum many-body systems. However, practical challenges to these techniques still remain, and algorithmic developments are needed before tensor network algorithms can be applied to more physics problems. In this thesis we present a variety of recent advancements to tensor network algorithms.

First we describe a DMRG-like algorithm for noninteracting fermions. Noninteracting fermions, naturally being gapless and therefore having high levels of entanglement, are actually a challenging setting for standard DMRG algorithms, and we believe this new algorithm can help with tensor network calculations in that setting.

Next we explain a new algorithm called the variational uniform matrix product state (VUMPS) algorithm that is a DMRG-like algorithm that works directly in the thermodynamic limit, improving upon currently available MPS-based methods for studying infinite 1D and quasi-1D quantum many-body systems.

Finally, we describe a variety of improvements to algorithms for contracting 2D tensor networks, a common problem in tensor network algorithms, for example for studying 2D classical statistical mechanics problems and 2D quantum many-body problems with projected entangled pair states (PEPS). One is a new variant of the corner transfer matrix renormalization group (CTMRG) algorithm of Nishino and Okunishi that improves the numerical stability for contracting asymmetric two-dimensional tensor networks compared to the most commonly used method. Another is the application of the VUMPS algorithm to contracting 2D tensor networks. The last is a new alternative to CTMRG, where the tensors are solved for with eigenvalue equations instead of a power method, which we call the fixed point corner method (FPCM). We present results showing the transfer matrix VUMPS algorithm and FPCM significantly improve upon the convergence time of CTMRG. We expect

these algorithms will play an important role in expanding the set of 2D classical and 2D quantum many-body problems that can be addressed with tensor networks.

PUBLISHED CONTENT AND CONTRIBUTIONS

- ¹M. T. Fishman and S. R. White, “Compression of correlation matrices and an efficient method for forming matrix product states of fermionic gaussian states”, [Phys. Rev. B **92**, 075132 \(2015\)](#).
- ²V. Zauner-Stauber, L. Vanderstraeten, M. T. Fishman, F. Verstraete, and J. Haegeman, “Variational optimization algorithms for uniform matrix product states”, [Phys. Rev. B **97**, 045145 \(2018\)](#).
- ³M. T. Fishman, L. Vanderstraeten, V. Zauner-Stauber, J. Haegeman, and F. Verstraete, “Faster Methods for Contracting Infinite 2D Tensor Networks”, [arXiv:1711.05881](#).

Chapter 2 is based on work published with Steve White in Ref. [1]. The work was initiated by the insight from Steve to diagonalize a correlation matrix with a local set of unitary gates and his idea that a discrete orthogonal wavelet transformation can be seen as a single-particle MERA. I was the main contributor (I performed all of the calculations, developed the technique to form a many-body MPS from the single-particle gates, developed the technique to produce a free fermion MERA, and developed the DMRG-like algorithm for obtaining the free fermion MPS from a quadratic Hamiltonian).

Chapter 3 is adapted from work published with Valentin Zauner-Stauber, Laurens Vanderstraeten, Frank Verstraete, and Jutho Haegeman in Ref. [2], of which I was a minor contributor. The chapter focuses on parts of the algorithm proposed in that reference that I contributed most to (helping to determine the preferred technique for obtaining the uMPS from the zero-site and one-site wavefunctions and the parallel algorithm for extending to multi-site unit cells).

Chapter 4 is primarily my own work. Some of it appears in Ref. [3] (posted to the arXiv and accepted for publication in Phys. Rev. B). I wrote the review of currently available CTMRG methods as well as developed the new CTMRG algorithm for contracting asymmetric tensor networks. Jutho pointed out the simplified version of the CTMRG method of Corboz et al.

Chapter 5 is based on work with Laurens Vanderstraeten, Valentin Zauner-Stauber, Jutho Haegeman, and Frank Verstraete posted to the arXiv in Ref. [3] and which has been accepted for publication in Phys. Rev. B. Part of the work is based on the VUMPS algorithm introduced in Ref. [2]. The work was initiated by insights from

Frank and Jutho for applying VUMPS to transfer matrices, as well as an original formulation of a fixed point corner method by Frank (and discussed in Ref. [4]). I am the main contributor (I performed all of the calculations, formulated the asymmetric form of the fixed point corner method, and contributed the idea to improve the fixed point corner method by combining it with CTMRG). Frank suggested the models for benchmarking. Laurens and Valentin performed previous calculations applying VUMPS to find fixed points of transfer matrices of various models and Laurens supplied the optimized iPEPS tensor for the Heisenberg model.

CONTENTS

Acknowledgements	iii
Abstract	v
Published Content and Contributions	vii
Contents	ix
List of Figures	xi
Chapter I: Introduction	1
Chapter II: Free Fermion Density Matrix Renormalization Group	4
2.1 Introduction	4
2.2 Background on Fermionic Gaussian States and Correlation Matrices	7
2.3 Algorithms	9
2.4 Numerical Results	22
2.5 Conclusion	29
Appendices	30
2.A Calculation of the Entanglement Entropy of a Fermionic Gaussian State	30
2.B GDMRG, an Algorithm to Obtain a Compressed Ground State Correlation Matrix as a GMPS	31
Chapter III: Variational Algorithms for Matrix Product States Directly in the Thermodynamic Limit	34
3.1 Introduction	34
3.2 A Variational Algorithm for Matrix Product States in the Thermodynamic Limit	36
3.3 Uniform MPS	37
3.4 Effective Hamiltonian	39
3.5 Updating the state	43
3.6 The Algorithm: VUMPS	45
3.7 Multi Site Unit Cell Implementations	48
3.8 Sequential Algorithm	49
3.9 Parallel Algorithm	50
3.10 Comparison of the two approaches	51
3.11 Conclusion and Outlook	51
Chapter IV: Revisiting the Corner Transfer Matrix Renormalization Group Algorithm for Asymmetric Lattices	53
4.1 Introduction	53

4.2	Corner transfer matrix (CTM) formalism	56
4.3	The corner transfer matrix renormalization group (CTMRG) algorithm	61
4.4	Results	72
4.5	Conclusion	75
Chapter V: Faster Methods for Contracting Infinite Two-Dimensional Tensor		
	Networks	77
5.1	Introduction	77
5.2	Problem Statement	79
5.3	Algorithm overview	81
5.4	Results	95
5.5	Conclusion and Outlook	102
Appendices 104		
5.A	New algorithm for isometrically gauging a uMPS	104
5.B	New algorithm for “biorthogonalizing” two uMPS	105
Bibliography 108		

LIST OF FIGURES

- 21 Fig. 21(a) shows the occupations n_b and corresponding entanglement $S_1(n_b)$ from diagonalizing a block of $B = 16$ sites in the middle of a system of free gapless fermions on $N = 1000$ sites at half filling. The minimum and maximum eigenvalues, n_1 and n_{16} , differ from 0 and 1 by $\approx 1.74 \times 10^{-11}$. The eigenvalues closest to $1/2$, $1/2 - n_8 = n_9 - 1/2 \approx 0.21$, have entropies $S_1(n_8) = S_1(n_9) \approx 0.60$, which are close to the maximum of $S_1(1/2) = \log(2) \approx 0.69$. Fig. 24(b) shows examples of eigenvectors from the same diagonalization. The eigenvectors with eigenvalues near 0 and 1, which contribute very little to the entanglement, are localized in the middle of the block, while the eigenvectors with eigenvalues closer to $1/2$ which contribute most to the entanglement have large support on the edges of the block. 8
- 24 In Fig. 24(a) we show schematically the procedure to obtain, given an approximate eigenvector \vec{v} of the correlation matrix Λ , the set of local rotation gates that make up our compressed correlation matrix. The example shown is for a block size $B = 4$ and system size $N = 8$. Fig. 24(b) shows that, by conjugating the correlation matrix by the gates obtained, the correlation matrix is approximately partially diagonalized. 13
- 25 Fig. 25(a) shows the overall gate structure obtained by the diagonalization procedure. These gates form the total $N \times N$ unitary V which approximately diagonalizes our correlation matrix Λ . By conjugating a diagonal matrix with the appropriate occupations of 0 or 1 found in the diagonalization procedure by this set of gates, we get an approximation for the correlation matrix. Fig. 25(b) shows an example of the correlations allowed by representing the correlation matrix Λ with a diagonal matrix conjugated by a finite depth circuit of depth $< N/2$. The grey area (the “light cone”) represents sites where there can be nonzero correlations with the first site. The circles in the middle represent a diagonal matrix with 1’s and 0’s on the diagonal, which is conjugated by a unitary change of basis approximated here by a finite depth circuit. For the circuit depth shown, there can’t be correlations with the last two sites. A circuit of depth $\geq N/2$ is required to allow for the possibility of nontrivial correlations across the entire system. . 14

- 26 An example of an alternative diagonalization scheme resulting in a MERA-like gate structure. Here we show a section of the first two renormalization steps, with 12 sites shown in the first layer and 6 renormalized sites shown in the second. A block size of $B = 4$ is used. For this block size there are two layers of disentanglers and one layer of isometries per level of the MERA. Open legs at the top of each layer correspond to diagonal modes of the correlation matrix (with eigenvalues 0 or 1) and are ignored at the next layer. 16
- 27 Here we show an example of a discrete wavelet transform written in the gate notation introduced in this paper. We show the D4 wavelet, which corresponds to a fermionic Gaussian MERA with one layer of disentanglers and one layer of isometries per layer. w_1 and s_1 (w_2 and s_2) label wavelet and scaling functions for the first (second) layer. Taking $\theta_1 = \pi/6$ and $\theta_2 = 5\pi/12$, we reproduce the conventional scaling coefficients for the D4 WT, $\vec{a}^T = (a_1, a_2, a_3, a_4) = (1 + \sqrt{3}, 3 + \sqrt{3}, 3 - \sqrt{3}, 1 - \sqrt{3})/(4\sqrt{2})$ 18
- 28 Here we show explicitly how to obtain the scaling and wavelet coefficients of the D4 WT from the circuit construction. Taking $\theta_1 = \pi/6$ and $\theta_2 = 5\pi/12$, in (a) and (b) we reproduce the conventional scaling coefficients for the D4 WT, $\vec{a}^T = (a_1, a_2, a_3, a_4) = (1 + \sqrt{3}, 3 + \sqrt{3}, 3 - \sqrt{3}, 1 - \sqrt{3})/(4\sqrt{2})$, up to a trivial reversal in the order. In (c) with the same choice of angles we reproduce the conventional wavelet coefficients $(a_4, -a_3, a_2, -a_1)$, again up to a trivial reversal and sign. 19
- 212 Examples of occupied and unoccupied modes found in the diagonalization process. Fig. 212(a) shows occupied/unoccupied modes for $\delta = 0.4$ (energy gap $\approx 0.806135t$). Fig. 212(b) shows occupied/unoccupied modes for $\delta = 0$ (energy gap ≈ 0.146088). 24
- 213 Examples of deviations in occupations at the end of the diagonalization procedure for $N = 128$ sites. Fig. 213(a) shows errors in the occupations for $\delta = 0.4$ (energy gap $\approx 0.806135t$). Fig. 213(b) shows errors in the occupations for $\delta = 0.0$ (energy gap $\approx 0.146088t$). 25

- 214 Block size B needed for a relative error in the energy of $< 10^{-6}$ as a function of number of sites N for spinless, gapless fermions with open boundary conditions at half filling. As expected from arguments about the entanglement of a critical system, we find $B \sim \log(N)$, tested up to $N = 2^{16} = 65536$ sites (note the log scale on the x axis). To study systems of this size and avoid the $O(N^3)$ diagonalization of the hopping Hamiltonian, we obtain the correlation matrix using the GDMRG algorithm as explained in Appendix 2.B. 26
- 215 Relative error in the energy for the proposed GMERA construction for increasing number of sites for a block size $B = 10$. The system analyzed is the ground state of free fermions hopping on a lattice with open boundary conditions. All errors are below 10^{-6} . As expected for a MERA, the error is seen to saturate for large N , indicating a fixed block size is sufficient to obtain an accuracy $< 10^{-6}$ up to very large system sizes. 27
- 216 A plot of the time to form the MPS approximation of gapped and gapless free fermion ground states at half filling as a function of sites N using gates from a GMPS. The bond dimensions are chosen large enough such that the relative errors in the energy of the MPS are below 10^{-6} . The block size of the GMPS used to form the MPS are the minimum required to obtain a GMPS with a relative error in the energy of 10^{-6} . A cutoff in the singular values of the SVD of 10^{-11} was used when applying the gates to form the MPS using the method described in Section 2.3.4. For the gapped case, the SSH model with $\delta = 0.1$ is used, corresponding to an energy gap of $\Delta \approx 0.2t$ (exact as $N \rightarrow \infty$). 28
- 4.41 Plots comparing our new method for obtaining the CTMRG projector compared to the method proposed in Ref. [108]. We apply these methods to the isotropic ferromagnetic 2D classical Ising model at inverse temperature $\beta/\beta_c - 1 = 10^{-3}$ with a non-unitary change of basis. Here we plot a comparison of: (a) the error in the magnetization as a function of bond dimension, (b) the correlation length as a function of bond dimension, and (c) the spectrum that needs to be inverted in creating the CTMRG projectors. See the main text for details about the computations and analysis of the results. 73

- 5.41 Plots of the error in the magnetization for the isotropic 2D classical Ising model 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. 97
- 5.42 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. 98
- 5.43 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. (5.35). This artificially breaks the lattice symmetry in order to test each method on an asymmetric network. (a) shows results for the asymmetric CTMRG algorithm by Corboz et al. in Ref. [108], (b) shows results for the FPCM introduced in this work combined with the CTMRG algorithm used in (a), (c) shows results for a newly introduced CTMRG algorithm introduced in Chapter 4, and (d) shows results for the FPCM introduced in this work combined with the CTMRG algorithm used in (c). For more details about the methods used, we refer readers to the main text. 99

- 5.44 (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 $\sqrt{d} = 5$), 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 $\sqrt{d} = 3$), and the boundary MPS bond dimension is $\chi = 800$ 100

Chapter 1

INTRODUCTION

The field of tensor networks, kicked off in 1992 by Steve White's invention of the spectacularly successful density matrix renormalization group (DMRG) algorithm, has exploded in popularity in recent years. Tensor networks are poised to play a role in helping us solve some of the greatest open physics problems of our time, such as understanding the nature of high-temperature superconductivity and illuminating a theory of quantum gravity. White's DMRG algorithm is best suited for calculating ground states of gapped, one-dimensional (1D) quantum many-body systems, a setting in which it is by far the most effective tool.

Unfortunately our most interesting open physics problems are in the real world, where there is oftentimes more than one dimension and a finite temperature. Although it has exponential scaling when applied to dimensions higher than one, DMRG is such a reliable algorithm that it is still the tool to beat for many 2D problems. DMRG, based on a variational state known as the matrix product state (MPS), can also be generalized for direct use in higher dimensions. It has become clear since the invention of DMRG that MPSs are simply the simplest variational state of a more general class of states known as tensor network states. The most popular of these higher-dimensional formulations for practical calculations makes use of a variational class of states known as tensor product states (TPS) or projected entangled pair states (PEPS), and in terms of those states polynomial-scaling algorithms can be formulated for problems in two and higher dimensions. Unfortunately these generalizations of the DMRG algorithm are more challenging to work with in practice, though a lot of progress has been made over the years to turn them into competitive numerical tools.

Many of the reasons why DMRG and related variational tensor network methods aren't used to solve more physics problems are ultimately algorithmic in nature. Much progress in improving variational tensor network algorithms, both those based on MPSs and higher-dimensional analogues, have been made and continue to be made to this day. In this thesis, we present on a variety of new developments to tensor network algorithms, which we hope will extend the use of DMRG and higher-dimensional analogues to new open problems in physics.

In Chapter 2, we present on a DMRG-like algorithm for noninteracting fermions. Tensor network states can be thought of as efficient data compressions: the amount of classical information in a quantum many-body state naively scales exponentially with the number of degrees of freedom in the system. In practice, however, most physical states we would encounter appear to not contain this much information, and tensor network states can be thought of as efficient compressions of the general quantum state into a much more efficient form. We show that, even though they already have an efficient representation, noninteracting quantum many-body states can be compressed into even more efficient forms, and we present simple and intuitive algorithm for performing that compression by performing local diagonalizations of the correlation matrix. Noninteracting fermions, naturally being gapless and therefore having high levels of entanglement, are actually a challenging setting for standard DMRG algorithms, and we believe these new algorithms can help with DMRG calculations in that setting.

In Chapter 3, we present on a new algorithm called the variational uniform matrix product state (VUMPS) algorithm that is a DMRG-like algorithm that works directly in the thermodynamic limit. The algorithm uses the ansatz of a uniform matrix product state (uMPS), and explicitly optimizes that variational state. This is in contrast to the infinite DMRG (iDMRG) algorithm, which is a DMRG algorithm that reaches the thermodynamic limit by growing the system size at each step, and the infinite time evolving block decimation (iTEBD) algorithm, which works with a uMPS in the thermodynamic limit but optimizing the state with a power method instead of variationally. Benchmark results presented in Ref. [2] show that this algorithm performs better than the state-of-the-art algorithms for a variety of 1D and quasi-1D systems in the thermodynamic limit, which is the exact limit of interest for many tensor network algorithms.

In Chapter 4, we present a short review of the corner transfer matrix renormalization group (CTMRG) algorithm of Nishino and Okunishi. CTMRG is one way to extend DMRG for studying 2D classical statistical mechanics problems. It also in practice plays a fundamental role in the most challenging part of infinite PEPS (iPEPS) calculations. In this chapter, we also present a new CTMRG approach that improves the numerical stability for contracting asymmetric two-dimensional tensor networks compared to the most commonly used method.

In Chapter 5, we apply the VUMPS algorithm from Ref. [2] and Chapter 3 to the study of transfer matrices, such that it can be used both for studying 2D classical statistical

mechanics problems as well as be used in iPEPS calculations. We also present a new approach to CTMRG where the tensors are solved for with eigenvalue equations, which we call the fixed point corner method (FPCM). With benchmarks on a variety of systems, we show that these two new algorithms perform better than CTMRG, and the improvements are particularly pronounced near critical points where standard DMRG algorithms tend to have trouble. These two new algorithms will be crucial to improving the performance of variational infinite projected entangled pair state (iPEPS) methods, a leading algorithm that generalizes DMRG to 2D.

*Chapter 2*FREE FERMION DENSITY MATRIX RENORMALIZATION
GROUP

¹M. T. Fishman and S. R. White, “Compression of correlation matrices and an efficient method for forming matrix product states of fermionic gaussian states”, [Phys. Rev. B **92**, 075132 \(2015\)](#).

Here we present an efficient and numerically stable procedure for compressing a correlation matrix into a set of local unitary single-particle gates, which leads to a very efficient way of forming the matrix product state (MPS) approximation of a pure fermionic Gaussian state, such as the ground state of a quadratic Hamiltonian. The procedure involves successively diagonalizing subblocks of the correlation matrix to isolate local states which are purely occupied or unoccupied. A small number of nearest neighbor unitary gates isolates each local state. The MPS of this state is formed by applying the many-body version of these gates to a product state.

We treat the simple case of compressing the correlation matrix of spinless free fermions with definite particle number in detail, though the procedure is easily extended to fermions with spin and more general BCS states. We also present a DMRG-like algorithm to obtain the compressed correlation matrix directly from a hopping Hamiltonian. In addition, we discuss a slight variation of the procedure which leads to a simple construction of the multiscale entanglement renormalization ansatz (MERA) of a fermionic Gaussian state, and present a simple picture of orthogonal wavelet transforms in terms of the gate structure we present in this paper. As a simple demonstration we analyze the Su-Schrieffer-Heeger model (free fermions on a 1D lattice with staggered hopping amplitudes).

2.1 Introduction

One of the strengths of the density matrix renormalization group (DMRG) [5, 6], and tensor network states in general, is that their power to simulate strongly correlated systems does not require the interactions to be weak. In fact, in fermion systems such as the Hubbard model, DMRG is *more* accurate for larger interactions. The matrix product state (MPS) representation of the wavefunction, which DMRG implicitly

uses, more efficiently compresses the wavefunction when interactions are strong, due to lower entanglement in a real-space basis.

In this paper, we introduce a new algorithm for efficiently producing an MPS representation for ground states of *noninteracting* fermion systems. Why is this useful, when DMRG is most useful in the opposite regime? This would be a valuable tool in a number of situations. For example, a powerful and widely used class of variational wavefunctions for strongly interacting systems begin with a mean-field fermionic wavefunction, and then one applies a Gutzwiller projection to reduce or eliminate double occupancy[7]. It could be very useful to find the overlaps of a DMRG ground state with a variety of such Gutzwiller states to help understand and classify the ground state. Once one has the MPS representation of the mean field state, the Gutzwiller projection is very easy, fast, and exact, whereas in other approaches it usually must be implemented with Monte Carlo. One might also begin a DMRG simulation with such a variational state, or in some cases with a mean field state without the Gutzwiller projection. Being able to represent fermion determinantal states as MPSs in a very efficient way also opens the door to using DMRG ground states as minus-sign constraints in determinantal quantum Monte Carlo, in particular in Zhang's constrained path Monte Carlo (CPMC) method[8, 9]. In this case one would hope that, for systems too big for accurate DMRG, at least the qualitative structure of the ground state could be captured by DMRG, and then the results could be made quantitative with the Monte Carlo method.

The basis of our approach shares ideas with DMRG. Matrix product state representations exploit a property of the state (low entanglement) to compress the information in the state. Fermionic Gaussian states (the general class of states which includes both fermion determinants, BCS states, and free fermion thermal states) are also compressible, as we will show. The properties of a Gaussian state are completely defined by its correlation matrix. For the case of a fermion determinant, the correlation matrix has eigenvalues which are either 0 or 1, i.e. they carry only a limited amount of information, indicating that the state can be compressed. In particular, one can perform an arbitrary single-particle change of basis within the occupied states, or within the unoccupied states, without changing the determinantal state. Tensor network methods in the context of fermionic Gaussian states have been studied previously in the context of the multiscale entanglement renormalization ansatz (MERA)[10] and projected entangled pair states (PEPS)[11], however here we present a simple and easily generalizable formalism and construction starting

with an efficient method for forming the MPS of a fermionic Gaussian state. We also present a new and simpler method for obtaining a fermionic Gaussian MERA (GMERA), the MERA of a fermionic Gaussian state, as a simple extension.

Our approach to producing the MPS of a fermionic Gaussian state also produces a compressed form of the correlation matrix itself, which we call a fermionic Gaussian MPS (GMPS), which might be useful in very different contexts where the single-particle matrices are very large. This compressed form expresses the $N \times N$ correlation matrix in terms of $O(BN)$ real angles which parametrize nearest neighbor rotation gates, where $B \ll N$ for states with low entanglement. The compressed form can be utilized directly. For example, ordinarily multiplying an arbitrary vector by the correlation matrix, which is not sparse, requires $O(N^2)$ operations, but by using the compressed form only $O(BN)$ operations are needed. For simplicity, the algorithm we introduce first utilizes the correlation matrix as the initial input. However, in Appendix 2.B we present a DMRG algorithm in the single particle context, which we call fermionic Gaussian DMRG (GDMRG), that starts with a single particle Hamiltonian matrix and outputs the ground state correlation matrix in compressed form as a GMPS at a greatly reduced cost compared to directly diagonalizing the Hamiltonian matrix, $O(B^3N)$ as opposed to $O(N^3)$. This algorithm exploits the close relationship between the correlation matrix and the density matrix of a many particle state, and many tensor network algorithms can similarly be translated into a single particle framework.

The paper is organized as follows. Section 2.2 gives a brief overview of fermionic Gaussian states and correlation matrices, including an introduction to the entanglement of these states. In Section 2.3, we give detailed descriptions of the new algorithms. Section 2.3.1 covers our new procedure for compressing a correlation matrix as a GMPS. Section 2.3.2 presents a variation of this method to obtain a GMERA. In Section 2.3.3 we give a brief introduction to how the GMERA gate structure relates to wavelet transforms. Section 2.3.4 covers the procedure for turning the gates obtained from compressing the correlation matrix into a many-body MPS approximation of the Gaussian state. Finally, Section 2.4 shows numerical results for the algorithms covered in the paper.

2.2 Background on Fermionic Gaussian States and Correlation Matrices

Consider the Hamiltonian for a 1D system of noninteracting fermions

$$\hat{H} = \sum_{i,j=1}^N \hat{a}_i^\dagger H_{ij} \hat{a}_j, \quad (2.1)$$

where a_i and a_i^\dagger are fermion operators and $H = [H_{ij}]$ is a Hermitian matrix ($H = H^\dagger$). We assume that the Hamiltonian terms are local (so the matrix H is band-diagonal).

Diagonalizing the matrix H , we have $H = UDU^\dagger$ where U is unitary and D is diagonal such that $D_{kk'} = \epsilon_k \delta_{kk'}$. The Hamiltonian can then be put into diagonal form,

$$\hat{H} = \sum_{k=1}^N \epsilon_k \hat{a}_k^\dagger \hat{a}_k, \quad (2.2)$$

where the operators which create the single particle energy eigenstates are

$$\hat{a}_k^\dagger = \sum_{i=1}^N U_{ik} \hat{a}_i^\dagger. \quad (2.3)$$

Assuming $\epsilon_k \leq \epsilon_{k'}$ if $k < k'$, the ground state is

$$|\psi_0\rangle = \prod_{k=1}^{N_F} \hat{a}_k^\dagger |\Omega\rangle, \quad (2.4)$$

where N_F is the number of particles in the system.

The *correlation matrix* is

$$\Lambda_{ij} = \langle \hat{a}_i^\dagger \hat{a}_j \rangle = \sum_{k=1}^{N_F} U_{ik}^* U_{jk}. \quad (2.5)$$

The correlation matrix fully characterizes $|\psi_0\rangle$ because all correlation functions, and therefore all observables, can be factorized into two-point correlators using Wick's theorem. Note that the eigenstates of H are also the eigenstates of Λ (the same U that diagonalizes H also diagonalizes Λ). However, the eigenvalues of Λ are either 1 (occupied) or 0 (unoccupied). The massive degeneracy of Λ means that we can make arbitrary changes of basis among the eigenstates of Λ as long as we do not mix occupied and unoccupied states.

In our procedure, we will be interested in finding localized eigenvectors of the correlation matrix which are (approximately) fully occupied or unoccupied. By

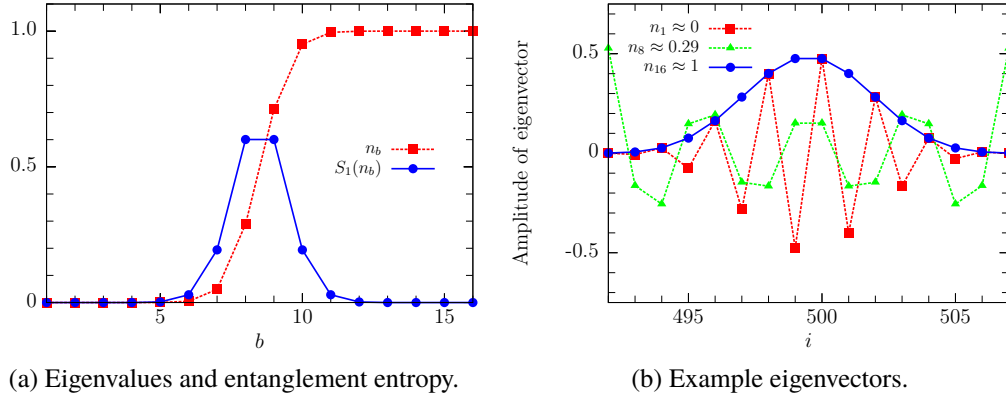


Figure 21: Fig. 21(a) shows the occupations n_b and corresponding entanglement $S_1(n_b)$ from diagonalizing a block of $B = 16$ sites in the middle of a system of free gapless fermions on $N = 1000$ sites at half filling. The minimum and maximum eigenvalues, n_1 and n_{16} , differ from 0 and 1 by $\approx 1.74 \times 10^{-11}$. The eigenvalues closest to $1/2$, $1/2 - n_8 = n_9 - 1/2 \approx 0.21$, have entropies $S_1(n_8) = S_1(n_9) \approx 0.60$, which are close to the maximum of $S_1(1/2) = \log(2) \approx 0.69$. Fig. 24(b) shows examples of eigenvectors from the same diagonalization. The eigenvectors with eigenvalues near 0 and 1, which contribute very little to the entanglement, are localized in the middle of the block, while the eigenvectors with eigenvalues closer to $1/2$ which contribute most to the entanglement have large support on the edges of the block.

rotating into the basis of these eigenvectors, we can locally diagonalize the correlation matrix, which will lead to a compression of the state. These eigenvectors have eigenvalues near 1 or 0, which makes them (approximate) eigenvectors of the entire correlation matrix and therefore uncorrelated with the rest of the system. What makes it possible to find a localized eigenvector?

The answer is the limited entanglement structure of the states we are interested in (ground states of local Hamiltonians). Consider the entanglement entropy of our fermionic Gaussian state, which can be calculated directly from the correlation matrix. Divide the system into an arbitrary subblock \mathcal{B} of B sites (with the corresponding submatrix of Λ , which we call $\Lambda_{\mathcal{B}}$) and the rest of the system. We would like to know how large of a block size B we need to find a localized eigenvector. If the matrix $\Lambda_{\mathcal{B}}$ has eigenvalues $\{n_b\}$ for $b \in \mathcal{B}$, with $0 \leq n_b \leq 1$, the entanglement entropy of the subblock \mathcal{B} , $S_B \equiv -\text{Tr}[\hat{\rho}_{\mathcal{B}} \log(\hat{\rho}_{\mathcal{B}})]$ (where $\hat{\rho}_{\mathcal{B}}$ is the reduced density matrix of the state in subblock \mathcal{B}), is

$$S_B(\{n_b\}) = \sum_{b \in \mathcal{B}} S_1(n_b), \quad (2.6)$$

where $S_1(n_b) = -[n_b \log(n_b) + (1 - n_b) \log(1 - n_b)]$. This expression has been shown elsewhere[12–15]. We show a simple, self-contained derivation of it in Appendix 2.A. Note that $S_1(n_b)$ vanishes for both $n_b \rightarrow 0$ and $n_b \rightarrow 1$.

The maximum amount of entanglement a block of size B can contain is when $n_b = 1/2$ for all $b \in \mathcal{B}$, so $S_B \leq B \log(2)$. This reflects a volume law entanglement in the “volume” B . However, ground states of 1D local Hamiltonians have entanglement that is much smaller, either of order unity (if the system is gapped), or the entanglement grows as $\log(B)$ if the system is gapless. To avoid the volume entanglement, most of the block eigenvalues n_b must be exponentially close to 0 or 1. In other words, as soon as we make B big enough so that the entanglement begins to saturate, except for a possible slow logarithmic growth, we should find at least one eigenvalue very close to 0 or 1. For gapless free fermions in 1D on $N = 1000$ sites, we show example eigenvalues, eigenvectors, and corresponding entanglements of a block of $B = 16$ in the middle of the correlation matrix in Fig. 21. Even for gapless free Fermions, with a block size of only $B = 16$ we find many eigenvalues near 0 or 1 (many localized eigenvectors). We use this observation next to develop algorithms to locally diagonalize correlation matrices and in the process find a very compressed form.

2.3 Algorithms

2.3.1 Compressing a Correlation Matrix as a GMPS

We begin the procedure by diagonalizing the upper left $B \times B$ subblock of a correlation matrix Λ of a pure fermionic Gaussian state. Assume that the state has some local entanglement structure, for example it is the ground state of a local Hamiltonian in 1D. For now, we imagine our system has open boundary conditions. For simplifying the discussion, from here on we assume our Hamiltonian is real (and therefore symmetric and diagonalized by an orthogonal matrix). We discuss the more general complex case at the end of the section. Let \mathcal{B} be the group of sites $1, \dots, B$ on the left end of the system, and $\Lambda_{\mathcal{B}}$ be the associated subblock of Λ . Also, let $\{n_b\}$ be the eigenvalues of $\Lambda_{\mathcal{B}}$ for $b \in \mathcal{B}$ where $0 \leq n_b \leq 1$. (This constraint on the eigenvalues of the subblock follows from the fact that both Λ and $1 - \Lambda$ are positive semi-definite.)

We increase B until we find some n_b that is nearly 1 or 0 within a specified tolerance, e.g. 10^{-6} . The closer the eigenvalue is to 1 or 0, the more accurate the compression, but a larger block size translates to more gates and a bigger bond dimension of

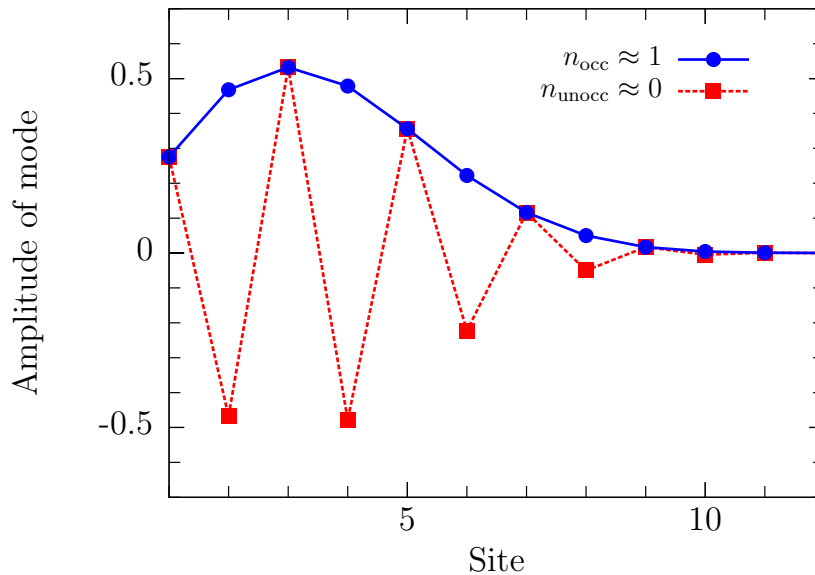


Figure 22: Examples of approximate occupied and unoccupied eigenvectors of Λ obtained from diagonalizing $\Lambda_{\mathcal{B}}$ where subblock \mathcal{B} are sites $1, \dots, B$. Λ is formed from the ground state of $\hat{H} = -t \sum_{i=1}^{N-1} (\hat{a}_i^\dagger \hat{a}_{i+1} + h.c.)$ for $N = 1024$ at half filling ($N_F = N/2$). A block size of $B = 12$ is used. Eigenvectors with highest (n_{occ}) and lowest (n_{unocc}) eigenvalues found from diagonalizing subblock \mathcal{B} are shown. We find $1 - n_{\text{occ}} = 2.4 \times 10^{-15}$ and $n_{\text{unocc}} = 7.3 \times 10^{-16}$, so the occupations are accurate to nearly machine double precision. $1 - n_{\text{occ}}$ and n_{unocc} should be equal at half filling (because of particle-hole symmetry), but are different in this case as a result of roundoff errors.

the MPS we will form. In Fig. 22 we show the most occupied and unoccupied eigenvectors of $\Lambda_{\mathcal{B}}$ for $B = 12$ for a system of gapless free fermions in 1D with $N = 1024$ sites. We see that $B = 12$ is sufficient to give deviations from occupancies of 0 or 1 to nearly machine double precision. The eigenvalues found in the bulk likely will not be as accurate, because states in the bulk will generally be more entangled than the ones on the edge. The smooth fall-off to zero at the right edge of the block is characteristic of these modes and is a result of diagonalizing the block on the left-most boundary of the system. The localized states we find here are *least* entangled with the rest of the system. This is in contrast to the dominant Schmidt states that are utilized within DMRG which have degrees of freedom that are localized at the edge of the block.

The eigenvector \vec{v} which is least entangled is also an approximate eigenvector of the total correlation matrix Λ , i.e. $\Lambda \vec{v} \approx n_1 \vec{v}$. Any $N \times N$ unitary matrix that has \vec{v} as its first column represents a change of basis that puts \vec{v} on the first site. The associated

transformation of Λ will make $\Lambda_{11} = n_1$, and zero out the rest of row 1 and column 1. The matrix of eigenvectors of $\Lambda_{\mathcal{B}}$ would produce such a matrix (expanding it to $N \times N$ by putting ones on the diagonal), but this $B \times B$ matrix does not translate well to many-particle gates to use in constructing an MPS.

We now introduce gate/circuit diagrams which apply equally well to simple matrix manipulations of Λ *and* to many-particle tensor networks. The basic ingredient of the diagrams are two site nearest neighbor unitary gates. In Figure 23 we show the relation between a gate and a matrix. In a later section we show how a gate is interpreted in the many-particle context of a tensor network. We consider nearest neighbor gates because these translate to fast MPS algorithms—typically, a non-nearest neighbor gate is implemented as a set of swap gates to bring the sites together, a nearest neighbor gate, followed by swaps to return to the original ordering of the sites, which is much slower than a single nearest neighbor gate. In the special case that the intermediate sites are in product states, i.e. bond dimension 1, nonlocal gates are also inexpensive, and we use these in our MERA algorithm.

Returning to the task of moving the least entangled state \vec{v} to the first site, a set of $B - 1$ two-site gates suffices. The first gate acts on sites $(B - 1, B)$, and we label it V_{B-1} . In general, we take

$$V_i = V(\theta_i) = \begin{pmatrix} \cos \theta_i & -\sin \theta_i \\ \sin \theta_i & \cos \theta_i \end{pmatrix}. \quad (2.7)$$

We choose $\theta_{B-1} = \tan^{-1}(v_B/v_{B-1})$, where v_i is the i^{th} component of the (un)occupied eigenvector of interest \vec{v} . With this choice, V_{B-1} acting on $\vec{v}^T = \begin{pmatrix} v_1 & \dots & v_{B-1} & v_B \end{pmatrix}$ sets the last component, v_B , to zero, and produces a new value of $v_{B-1} \rightarrow v'_{B-1}$. In other words, we solve for θ_{B-1} so that $\vec{v}^T V_{B-1} = \begin{pmatrix} v_1 & \dots & v_{B-1} & v_B \end{pmatrix} V_{B-1} = \begin{pmatrix} v_1 & \dots & v'_{B-1} & 0 \end{pmatrix}$. Next we rotate sites $(B-2, B-1)$, with $\theta_{B-2} = \tan^{-1}(v'_{B-1}/v_{B-2})$, and continue in this fashion. The action of all these gates on \vec{v}^T gives $\delta_{i,1}$, so they act to change the basis into the one containing \vec{v} .

We take $V_{\mathcal{B}} = V(\theta_{B-1})V(\theta_{B-2}) \dots V(\theta_1)$. This procedure is shown schematically for a simple case in Fig. 24(a). We then apply the gates to Λ . The transformed correlation matrix $V_{\mathcal{B}}^\dagger \Lambda V_{\mathcal{B}}$ will have $n_1 \approx 1$ or 0 as the top left entry (and nearly 0 in the rest of the entries in the first row and first column). A schematic of this transformation is shown in Fig. 24(b). We will call the first block $\mathcal{B}_1 \equiv \mathcal{B}$. We repeat this procedure for \mathcal{B}_2 , sites $2, \dots, B + 1$, now simply ignoring the first site. For half-filled systems, the modes found are likely to alternate between occupied

$$\left| \right| \left| \right| \left| \right| \left| \right| \equiv \left(\begin{array}{cccc} 1 & & & \\ & 1 & & \\ & & V(\theta_3) & \\ & & & 1 \\ & & & & 1 \end{array} \right)$$

$$V(\theta_3) = \begin{pmatrix} \cos \theta_3 & -\sin \theta_3 \\ \sin \theta_3 & \cos \theta_3 \end{pmatrix}$$

Figure 23: Definition of a gate used throughout the paper. Example for $N = 8$ sites for a gate at site $i = 3$. Unless specified otherwise, circuits are in a direct sum space. We take the convention that multiplying a matrix from the top by a vector corresponds to multiplying the matrix on the right by a column vector.

and unoccupied because occupied and unoccupied modes will generally be found in pairs when diagonalizing a block of the correlation matrix. Of course, B does not have to stay the same from one block to the next, and in general it is better to set it dynamically to make n_k sufficiently close to 1 or 0. For the last blocks, B is decreased to the remaining number of sites. After N blocks, we will have approximately diagonalized Λ .

The overall unitary transformation is $V = V_{\mathcal{B}_1} V_{\mathcal{B}_2} \dots V_{\mathcal{B}_{N-1}}$. The matrix V decomposed into the 2×2 rotation gates $\{V(\theta_i)\}$ for $N = 8$ and $B = 4$ is shown in Fig. 25(a). The $N \times N$ unitary approximately rotates our single particle basis from real space to what we refer to as the *occupation basis*, which is one of the highly degenerate eigenbases of the correlation matrix. Conjugating Λ by V gives us a matrix $V^\dagger \Lambda V$ that is nearly diagonal, with N_F values on the diagonal close to 1 corresponding to occupied modes and $N - N_F$ values on the diagonal close to 0 corresponding to unoccupied modes. In total, the procedure as described would require $O(BN)$ nearest neighbor rotations, where B is the largest block size needed for the desired accuracy of the representation of the correlation matrix.

Writing the 2×2 rotations as gates is very convenient for understanding the matrix transformations, but more importantly it makes it easy to connect to many-body gates and to quantum circuits in general. As a quantum circuit, these gates have a slightly peculiar structure. Because of how the diagonalizations overlapped, the

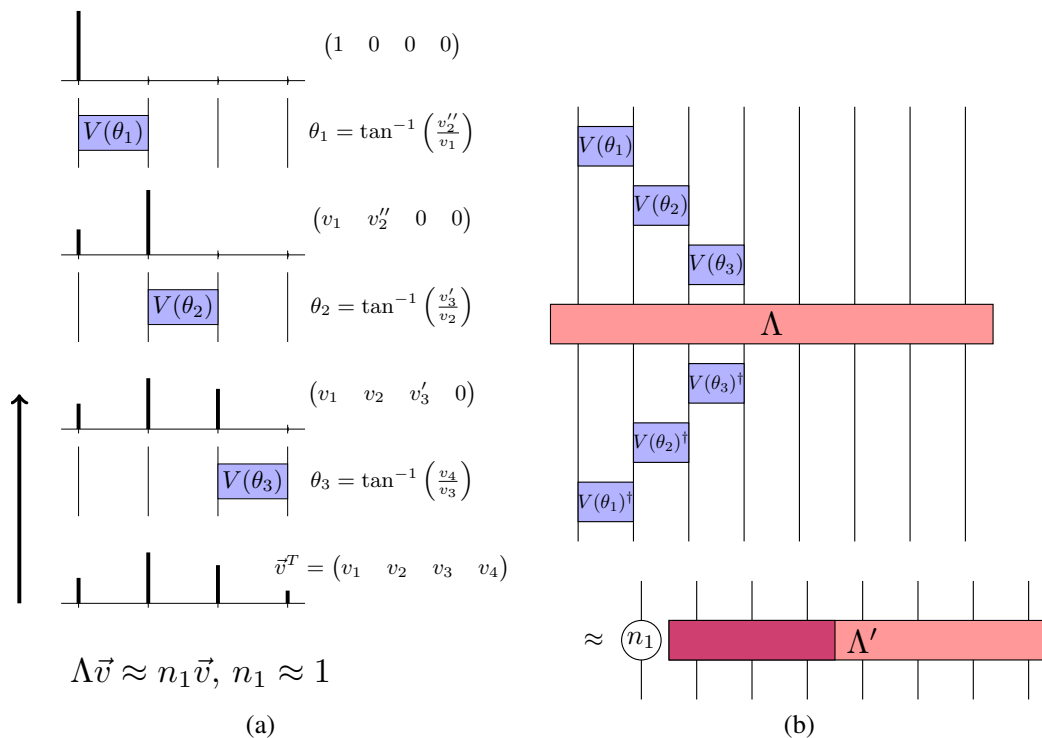


Figure 24: In Fig. 24(a) we show schematically the procedure to obtain, given an approximate eigenvector \vec{v} of the correlation matrix Λ , the set of local rotation gates that make up our compressed correlation matrix. The example shown is for a block size $B = 4$ and system size $N = 8$. Fig. 24(b) shows that, by conjugating the correlation matrix by the gates obtained, the correlation matrix is approximately partially diagonalized.

circuit has a depth of $O(N)$. However, a vertical cut through the circuit only passes through $O(B)$ gates. This construction and gate structure is in a certain sense optimal if we limit ourselves strictly to circuits with local gates. If we want to represent a correlation matrix in a compact way with nearest neighbor gates, we would like to be able to represent arbitrary correlations in the system (correlations at all lengths), and in particular, correlations between the first and last site. In Fig. 25(b), we show a circuit which cannot connect the first and last sites because its depth is less than $N/2$. Although our gate structure, shown in Fig. 25(a), has a depth $\sim N$, in fact we can adjust our diagonalization procedure slightly to obtain a depth of $\sim N/2$ so our circuit can capture correlations of all lengths. This is done by beginning the diagonalization procedure from both the left and right side of the system until the blocks meet in the middle. This freedom in where to start the diagonalization is similar to the choice of gauge of an MPS. Choosing one gauge over another can be useful if we have already performed this procedure for a correlation matrix and want

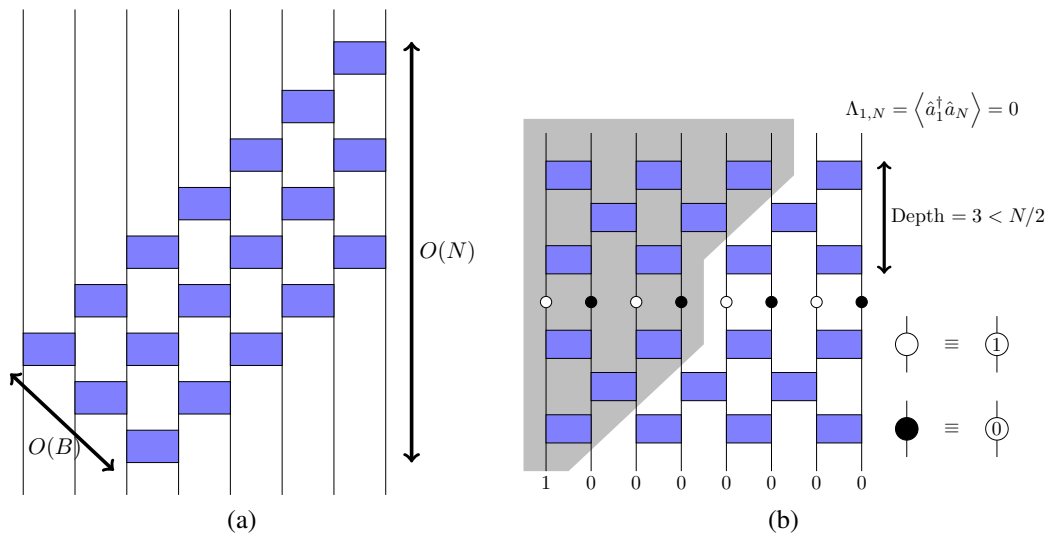


Figure 25: Fig. 25(a) shows the overall gate structure obtained by the diagonalization procedure. These gates form the total $N \times N$ unitary V which approximately diagonalizes our correlation matrix Λ . By conjugating a diagonal matrix with the appropriate occupations of 0 or 1 found in the diagonalization procedure by this set of gates, we get an approximation for the correlation matrix. Fig. 25(b) shows an example of the correlations allowed by representing the correlation matrix Λ with a diagonal matrix conjugated by a finite depth circuit of depth $< N/2$. The grey area (the “light cone”) represents sites where there can be nonzero correlations with the first site. The circles in the middle represent a diagonal matrix with 1’s and 0’s on the diagonal, which is conjugated by a unitary change of basis approximated here by a finite depth circuit. For the circuit depth shown, there can’t be correlations with the last two sites. A circuit of depth $\geq N/2$ is required to allow for the possibility of nontrivial correlations across the entire system.

to perform it again for another correlation matrix which is only locally different from the first one. If we choose the gauge center where the correlation matrix has changed, we only need to change a local set of gates.

A generic local circuit of depth $O(N)$ contains $O(N^2)$ gates, and can represent an arbitrary $N \times N$ single-particle unitary change of basis. The low entanglement of physical ground states allows us to represent an $N \times N$ matrix with $O(BN)$ one-parameter gates, with $B \ll N$. For a gapless system, we know from conformal field theory that the entanglement of a subblock \mathcal{B} of B sites varies as $S_B \sim \log(B)$. This means that we should be able to capture the entanglement of a critical system of N sites with a block size $B \sim \log(N)$. If we find that $B \sim \log(N)$, this means that our construction is roughly optimal. Fig. 214 in Section 2.4.1 shows numerical evidence that this is indeed the case.

2.3.2 Compressing a Correlation Matrix as a GMERA

A MERA tensor network[16] can represent a 1D critical system using a constant bond dimension, unlike an MPS. In our MPS construction, this is reflected in that $B \sim \log(N)$. However, we can adjust the diagonalization procedure slightly to obtain a MERA-like gate structure with a B which does not grow with N . The MERA for fermionic Gaussian states was first studied in [10], but was only used to study infinite translationally invariant systems and required a subtle optimization scheme. Here we will show a simpler construction only requiring the tools we have explained so far.

We begin the procedure in the same way as we did for the GMPS, by diagonalizing the block corresponding to sites $1, \dots, B$ of the correlation matrix. Just as before, for a large enough block size we find an occupied or unoccupied mode and rotate into the basis containing that mode with $B-1$ local 2×2 gates. Next, instead of diagonalizing the block starting at site 2, we instead diagonalize the block corresponding to sites $3, \dots, B+2$, again finding an occupied or unoccupied mode and rotating into that basis. The state at site 2 is “left behind”—it is not a low entangled state, so we cannot ignore it, but we leave it for a later stage of the algorithm. We continue in this manner, diagonalizing blocks starting at odd sites of size B to obtain $\sim BN/2$ nearest neighbor gates. Approximately half of the modes are fully occupied or unoccupied and are projected out (meaning the associated rows and columns in the correlation matrix are ignored in later stages). The other half were left behind, and continue as the sites of the next layer of the gate structure. By only trying to get $N/2$ unentangled modes in the first layer, the size of B does not need to grow with N , as we show below.

The left-behind sites pass through to the next layer and are interpreted as a course-grained version of our original state on only $N/2$ sites. We repeat the same procedure for this new course-grained system of $N/2$ sites, starting by diagonalizing the subblock of the first $1, \dots, B$ sites of the new course-grained lattice, finding an occupied or unoccupied mode of the course-grained system, and projecting it out. Here, however, the gate we use to rotate into the basis of the (un)occupied mode are 2×2 nearest neighbor gates in the course-grained lattice, but are actually next-nearest neighbor gates acting on the original lattice (if we project out every other site). Ordinarily, using next-nearest neighbor gates (or longer range gates at higher levels of the MERA) would be costly in the many-body case, requiring swap gates to make them effectively nearest neighbor. However, the projected-out sites are now

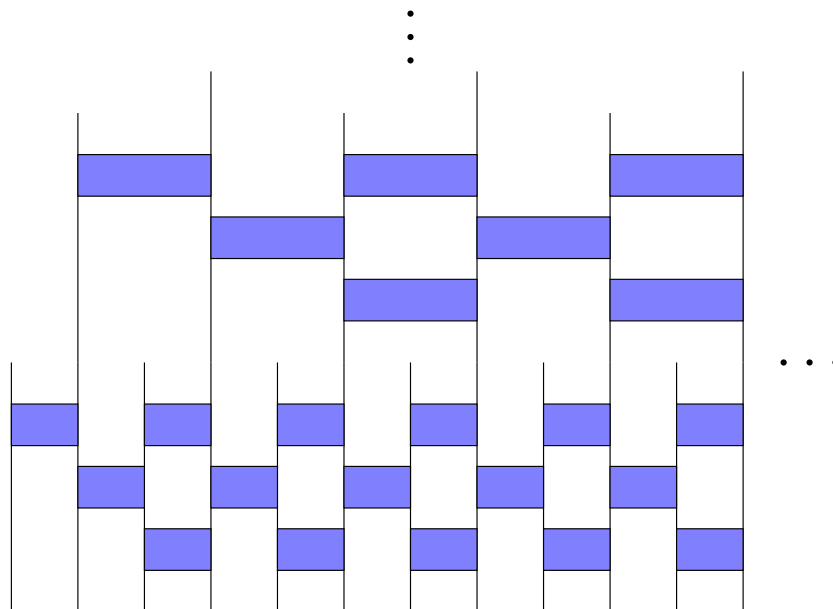


Figure 26: An example of an alternative diagonalization scheme resulting in a MERA-like gate structure. Here we show a section of the first two renormalization steps, with 12 sites shown in the first layer and 6 renormalized sites shown in the second. A block size of $B = 4$ is used. For this block size there are two layers of disentanglers and one layer of isometries per level of the MERA. Open legs at the top of each layer correspond to diagonal modes of the correlation matrix (with eigenvalues 0 or 1) and are ignored at the next layer.

in product states, meaning that swapping does not require significant time.

We repeat the above procedure of projecting out every other effective site and course graining to a lattice of half the size. All of the sites will be projected out after this course-graining is repeated $O(\log_2(N))$ times. Fig. 26 shows the first two layers of the resulting gate structure, which looks like a MERA with $B - 2$ layers of nearest neighbor 2-site disentanglers and a layer of nearest neighbor 2-site isometries. The total number of gates in the construction is $\sim B(N/2 + N/4 + \dots + 1) = BN$, the same gate count for a fixed block size B as for the GMPS. We call this gate structure, which like our GMPS construction is a compression of an $N \times N$ correlation matrix into $\sim BN$ gates, a fermionic Gaussian MERA or GMERA. In this figure, open legs at the top of each layer are modes that are uncorrelated with the rest of the sites and can be ignored in the next layer. Some extra gates will be required to project out the leftover sites at the right end of the system (not shown in Fig. 26), and there is some flexibility in how to do this which will change the accuracy of the compression slightly. For example, one could use a gate structure similar to the

GMPS construction to project out all of the leftover sites at the end.

How does the block size B of the GMERA compare to that in our GMPS construction? We show numerically in Section 2.4.2 that for a simple gapless Hamiltonian the GMERA does indeed produce accurate results with a block size $B = O(1)$, independent of the system size, making it much more efficient in the large N limit.

2.3.3 Discrete Wavelet Transforms and Fermionic Gaussian MERA

We would like to point out the similarity between the MERA gate structure and orthogonal wavelet transforms (WT), such as the WTs that produce the well-known Daubechies wavelets [17, 18]. Of course, the development of wavelets has not been in a many particle context, and, for now, we restrict ourselves to the matrix interpretation of the diagrams. For compact wavelets, an orthogonal wavelet transform is a local unitary transformation. It is not usually represented in terms of two-site gates, but this representation turns out to be particularly convenient. To be specific, we start with the simplest nontrivial WT, the D4 Daubechies WT. This WT is defined by four coefficients $\{a_j\}$ for $j = 1, \dots, 4$ which characterize how the D4 scaling function is related to itself at different scales through the equation $s(x) = \sum_j a_j \sqrt{2} s(2x - j)$. The matrix form of the WT is given by

$$\begin{pmatrix} a_1 & a_2 & a_3 & a_4 & 0 & 0 & 0 \\ a_4 & -a_3 & a_2 & -a_1 & 0 & 0 & 0 \\ 0 & 0 & a_1 & a_2 & a_3 & a_4 & 0 \\ 0 & 0 & a_4 & -a_3 & a_2 & -a_1 & 0 \\ & & & & & & \ddots \end{pmatrix}. \quad (2.8)$$

The $\{a_j\}$ are carefully chosen to ensure orthogonality between scaling functions centered at different sites, and to make the scaling functions have desirable completeness properties. For example, linear combinations of the D4 scaling functions centered at different sites, $\{s(x - k)\}$ for integer k , fit any linear function, so the resulting coefficients are $\vec{a}^T = (1 + \sqrt{3}, 3 + \sqrt{3}, 3 - \sqrt{3}, 1 - \sqrt{3})/(4\sqrt{2})$. The orthogonality requirement results in nonlinear equations to solve for the $\{a_j\}$ which becomes complicated for higher order. The second row of the matrix gives the coefficients that produces wavelets, designed to represent high momentum degrees of freedom. In terms of our MERA procedure, the wavelets are left behind, while the scaling functions propagate to the next level.

The D4 WT has a very simple gate structure, identical to our MERA structure with $B = 3$, shown for two layers in Fig. 27. In each horizontal layer of gates, all

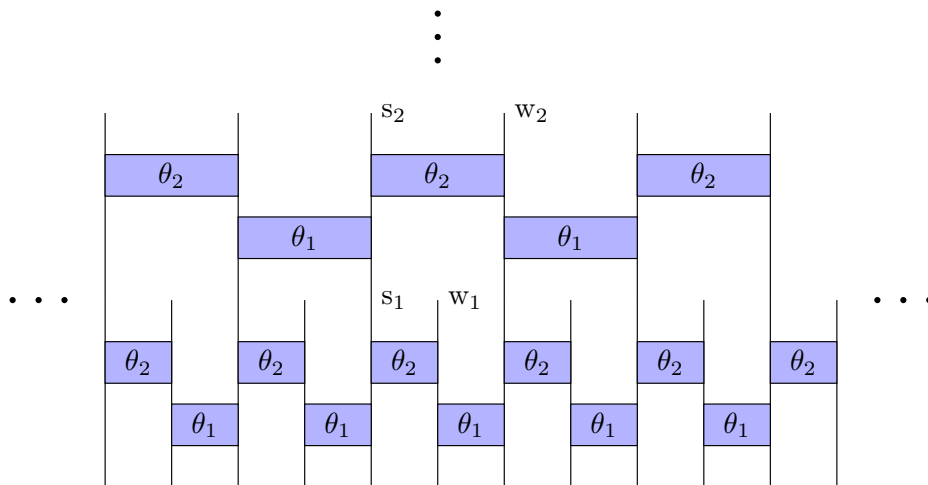
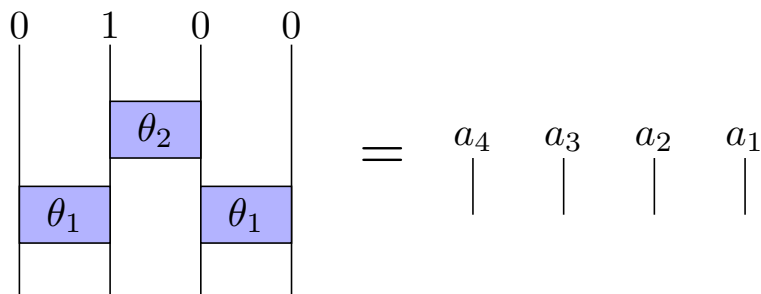


Figure 27: Here we show an example of a discrete wavelet transform written in the gate notation introduced in this paper. We show the D4 wavelet, which corresponds to a fermionic Gaussian MERA with one layer of disentanglers and one layer of isometries per layer. w_1 and s_1 (w_2 and s_2) label wavelet and scaling functions for the first (second) layer. Taking $\theta_1 = \pi/6$ and $\theta_2 = 5\pi/12$, we reproduce the conventional scaling coefficients for the D4 WT, $\vec{a}^T = (a_1, a_2, a_3, a_4) = (1 + \sqrt{3}, 3 + \sqrt{3}, 3 - \sqrt{3}, 1 - \sqrt{3})/(4\sqrt{2})$.

gates have the same angle. The D4 WT is specified by only two angles, θ_1 for the bottom layer and θ_2 for the next. Higher order WTs of this type (e.g. D6, D8, etc.) correspond to larger B . (For example, the D6 WT looks like Fig. 6). Given the angles, one gets the $\{a_j\}$ by setting all the top values of the circuit to zero except a 1 on one site and applying the 2×2 rotations in the layers below. The support of the scaling functions is made obvious using the gate structure, as there will be $2L$ nonzero values at the bottom of the circuit for L layers of gates. For the D4 WT, one finds that $\theta_1 = \pi/6$ and $\theta_2 = 5\pi/12$ reproduces the D4 $\{a_j\}$, up to a trivial reversal of the coefficients. (A single layer with $\theta_1 = \pi/4$ gives the trivial Haar wavelets, which have been used previously as a basis for transforming fermionic Gaussian states by Qi [19].) The scaling functions at the larger scales are found by performing the same transformation of L layers of gates on the scaling functions found at the previous scale.

In Fig. 28 we show how scaling coefficients $\{a_j\}$ come from the gate structure, applying a vector to the top of the circuit with 1 at the site of a scaling function and 0's elsewhere. In simple wavelet treatments, the wavelet coefficients are obtained from the scaling coefficients $\{a_j\}$ as $\{(-1)^{j-1}a_{2L-j+1}\}$ for $j = 1, \dots, 2L$. Here, they



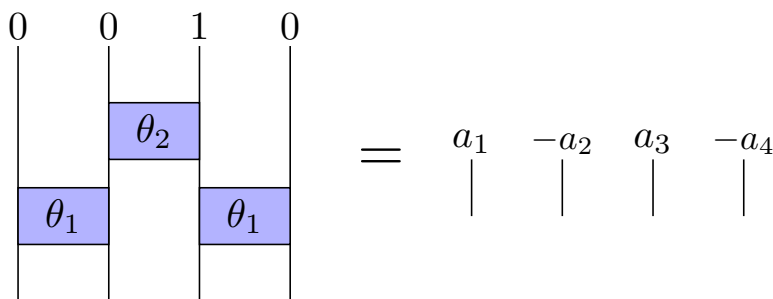
(a) Scaling coefficients from gate structure.

$$\begin{pmatrix} c_1 & -s_1 & & \\ s_1 & c_1 & & \\ & & c_1 & -s_1 \\ & & s_1 & c_1 \end{pmatrix} \begin{pmatrix} 1 & & & \\ & c_2 & -s_2 & \\ & s_2 & c_2 & \\ & & & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} a_4 \\ a_3 \\ a_2 \\ a_1 \end{pmatrix}$$

where $(a_4 \ a_3 \ a_2 \ a_1) = (-s_1 c_2 \ c_1 c_2 \ c_1 s_2 \ s_1 s_2)$

$$\text{and } c_i = \cos(\theta_i), \quad s_i = \sin(\theta_i)$$

(b) Gate structure in (a) written in terms of matrices and vectors.



(c) Wavelet coefficients from gate structure.

Figure 28: Here we show explicitly how to obtain the scaling and wavelet coefficients of the D4 WT from the circuit construction. Taking $\theta_1 = \pi/6$ and $\theta_2 = 5\pi/12$, in (a) and (b) we reproduce the conventional scaling coefficients for the D4 WT, $\vec{a}^T = (a_1, a_2, a_3, a_4) = (1 + \sqrt{3}, 3 + \sqrt{3}, 3 - \sqrt{3}, 1 - \sqrt{3})/(4\sqrt{2})$, up to a trivial reversal in the order. In (c) with the same choice of angles we reproduce the conventional wavelet coefficients $(a_4, -a_3, a_2, -a_1)$, again up to a trivial reversal and sign.

are obtained by shifting the location of the 1 at the top of the circuit, but we can show in general that this gives the same result. This is done by noting that the shift of the 1 to get the wavelet coefficients looks like a swap at the top of the circuit. We can “pull through” this swap by conjugating each layer of the WT with a transformation that reverses the order of the sites. This conjugation also negates the angles in the circuit. It leaves a site reversal at the bottom of the circuit, reversing the order of the coefficients. The angle negation negates the sine terms, leading to the same coefficients except with every other one negated, since every other site will have an even or odd number of $\sin(\theta_i)$ multiplied together.

Given an arbitrary set of $\{a_j\}$, we can use the same procedure that brought \vec{v} to the first site in our GMPS procedure to find all the angles defining the WT, i.e $\vec{v} = \vec{a}$. Thus, any compact orthogonal WT of this general type can be represented by a simple gate structure. Because wavelets are much easier to understand than generic many particle wavefunctions, the connection between MERA and wavelets may help provide intuition that helps one understand MERA.

2.3.4 Forming the Many-Body MPS from the GMPS (or GMERA)

For a number conserving real Hamiltonian H , the many particle unitary gate \hat{V}_i corresponding to the single particle rotation V_i , in the basis

$$\{|\Omega\rangle, \hat{a}_i^\dagger |\Omega\rangle, \hat{a}_{i+1}^\dagger |\Omega\rangle, \hat{a}_i^\dagger \hat{a}_{i+1}^\dagger |\Omega\rangle\}, \quad (2.9)$$

is

$$[\hat{V}_i] = [\hat{V}(\theta_i)] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta_i & \sin \theta_i & 0 \\ 0 & -\sin \theta_i & \cos \theta_i & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (2.10)$$

This reinterpretation of the gates is the only change needed to make our matrix gate structures act on the many particle Hilbert space.

Say we have compressed the correlation matrix of a pure fermionic Gaussian state as a GMPS. To create the MPS representation of this state, we begin with a product state, with each site being occupied or unoccupied, with the occupations given by n_k obtained in our diagonalization procedure (set to 1 or 0 for $n_k \approx 1$ or 0). We then apply, one by one, all of the nearest neighbor gates $\{\hat{V}_i\}$ (the many-body gates corresponding to the gates $\{V_i\}$ obtained with Eq. 2.10) in the opposite order in which they were obtained with our diagonalization procedure. The repeated application

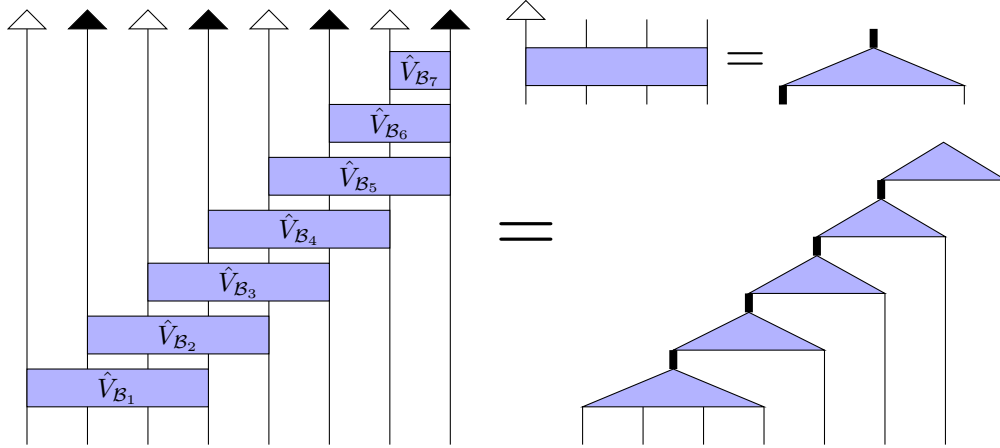


Figure 29: Tensor diagram showing the structure of gates $\{\hat{V}_{\mathcal{B}_i}\}$ for $i = 1, \dots, N - 1$ obtained in our procedure and how they contract to form an MPS. The white (black) triangles represent projectors onto the occupied (unoccupied) state. The ordering of the occupied and unoccupied states is determined by the ordering of the occupations found in the diagonalization procedure, one particular example at half filling is shown here. Here we show a system with $N = 8$ sites and a block size $B = 4$. The diagram on the right shows that once the sites are rotated into a basis where one of the modes is occupied or unoccupied (generally with some alternating pattern), the fully occupied or unoccupied modes can be projected out. The transformations $\{\hat{V}_{\mathcal{B}_i}\}$, including the projections, can be directly interpreted as the tensors composing the MPS representation of our many-body state if we do an exact contraction, or we can apply them as a set of gates as explained in the text.

of gates is similar to the time-evolving block decimation (TEBD) algorithm[20] or the time dependent DMRG algorithm[21], but the pattern of gates and ordering is different. We apply the two body gates by moving the center of the MPS to the location of the gate, contract the gate with the two relevant tensors in the MPS, and then form the new MPS by performing a singular value decomposition (SVD), with possible truncation of states by throwing out states with small singular values.

We can also form the MPS from our GMERA construction in a similar manner. However, instead of starting with a full product state, we start with the gates at the top of the MERA and work our way down, including only the sites that have been touched by a gate at that level or above. When a site is added, it starts as a completely occupied or unoccupied state, and is immediately mixed with another site by a gate. The number of sites involved roughly doubles with each layer, and after $O(\log_2(N))$ layers of the MERA we have our MPS approximation for the entire system. Again, we can truncate as needed by throwing out low weight states after the SVD as we work our way down.

Returning to the MPS construction, the tensors of the MPS could also be constructed directly by contractions of the gates as shown in Fig. 29. In this diagram the small black and white triangles signify projectors onto the appropriate occupations found, while the thick lines signify combined internal indices which form the internal bonds of the MPS. From this perspective it is easy to see that picking a block size B for diagonalizing the correlation matrix would correspond to an MPS with a bond dimension of $\chi = 2^{B-1}$. We find it simpler and more efficient to apply the gates layer by layer instead of this method. Layer by layer, it is natural to truncate the MPS with SVDs during the construction, and this can lead to an MPS with a smaller bond dimension than 2^{B-1} for the required accuracy. The SVD truncation takes one out of the manifold of Gaussian states, where the greater freedom for a fixed bond dimension allows one to find a state which is closer to the desired Gaussian state than one could within the Gaussian manifold. However, one should pick a block size so that 2^{B-1} is as close to the target bond dimension as possible.

We can adapt our circuits to complex quadratic Hamiltonians, where the gates are of the same form but the 2×2 submatrix rotating the singly occupied subspace is a general matrix in $SU(2)$ parameterized by two angles. Even more generally, we can extend this procedure to quadratic Hamiltonians with pairing terms to compress BCS states, where the gates required are not just number conserving but general parity conserving gates (so they involve mixing of unoccupied and doubly occupied subspaces of the 2 sites of interest). This matrix would in general be parameterized by 5 angles (one matrix in $SU(2)$ rotating the singly occupied subspace, one matrix in $SU(2)$ rotating the empty and doubly occupied subspaces, and a relative phase). This form of gates has been studied previously in the context of classically simulating quantum circuits using the matchgate formalism; see for example[22, 23].

2.4 Numerical Results

Here we show numerical results for the algorithms we presented. In order to study systems that are both gapless and gapped, we study a simple model, the Su-Schrieffer-Heeger model [24]. This is a model of 1D spinless fermions hopping on a lattice with staggered hopping amplitudes, t_1 and t_2 . The Hamiltonian is

$$\hat{H}_{SSH} = \sum_{i=1}^{\frac{N-1}{2}} (t_1 \hat{a}_{2i-1}^\dagger \hat{a}_{2i} + t_2 \hat{a}_{2i}^\dagger \hat{a}_{2i+1} + h.c.). \quad (2.11)$$

We will take $t_1 = -t \left(1 + \frac{\delta}{2}\right)$ and $t_2 = -t \left(1 - \frac{\delta}{2}\right)$. The model has an energy gap in the bulk between the ground state and first excited state of $\Delta = 2|\delta|t$ in the

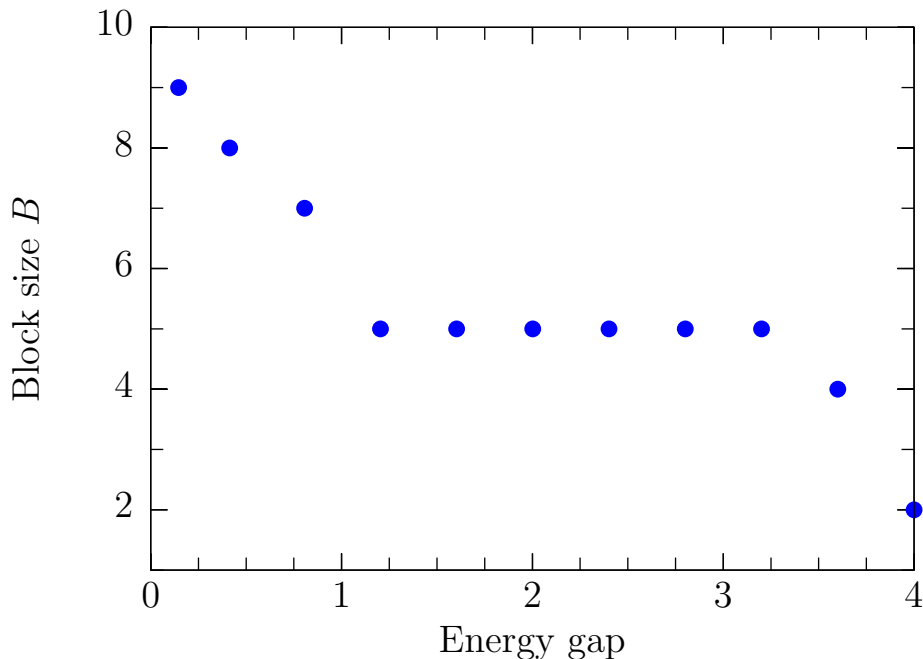


Figure 210: Block size required to obtain a relative error in the total energy of less than 10^{-6} as a function of the calculated energy gap (in units of t) for $N = 128$ sites.

thermodynamic limit ($N \rightarrow \infty$). With open boundary conditions, it can contain exponentially decaying zero energy modes localized on the ends of the chain.

2.4.1 Results for Compressing a Correlation Matrix as a GMPS

We start with a simple test of obtaining the GMPS compression of the ground state correlation matrix of the SSH model for $N = 128$ lattice sites for various energy gaps at half filling ($N_F = N/2$). We analyze the range of δ from 0 to 2. The ground state for $\delta = 0$ is (approximately) gapless while for $\delta = 2$ it is fully gapped (the chain uncouples). Fig. 210 shows the block size required to obtain a GMPS with a relative error in the total energy of less than 10^{-6} as a function of the calculated energy gap. The exact ground state energy and energy gap are calculated by diagonalizing the hopping Hamiltonian H_{SSH} . This corresponds to the accuracy of the MPS representation of the ground state if the GMPS written with many-body gates is contracted with no further truncation of the MPS, so a GMPS block size B corresponds to an MPS of bond dimension $\chi = 2^{B-1}$ (which is why the block size remains constant for intermediate energy gaps). The plot shows, as expected, that the block size required decreases as the energy gap is increased. Fig. 211 shows, for the case $\delta = 0$ (where the energy gap, due to the finite size, is $0.146088t$), the relative error in the energy as a function of the block size.

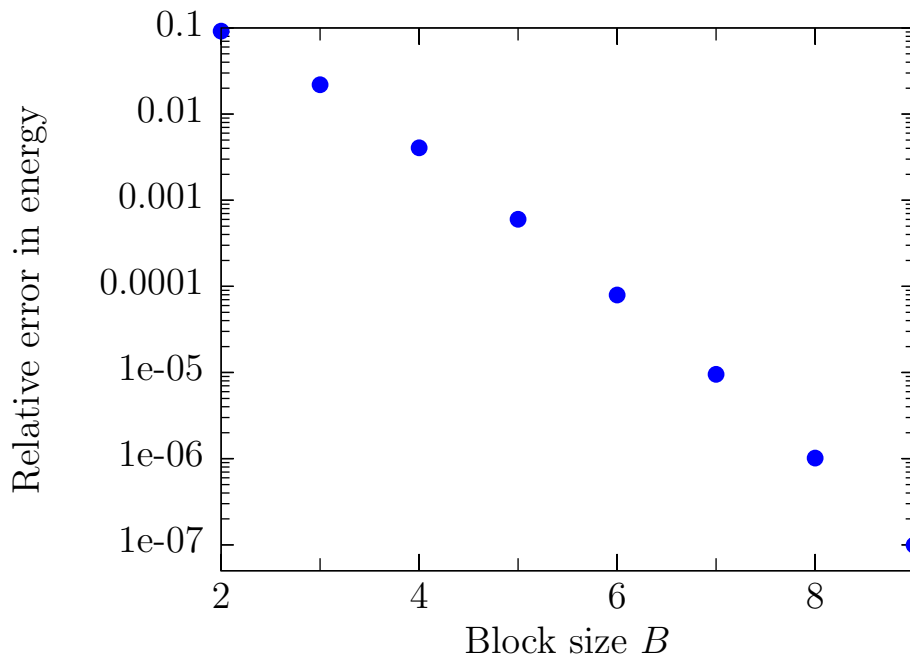


Figure 211: Relative error in the total energy as a function of the block size B for $N = 128$ sites and $\delta = 0$.

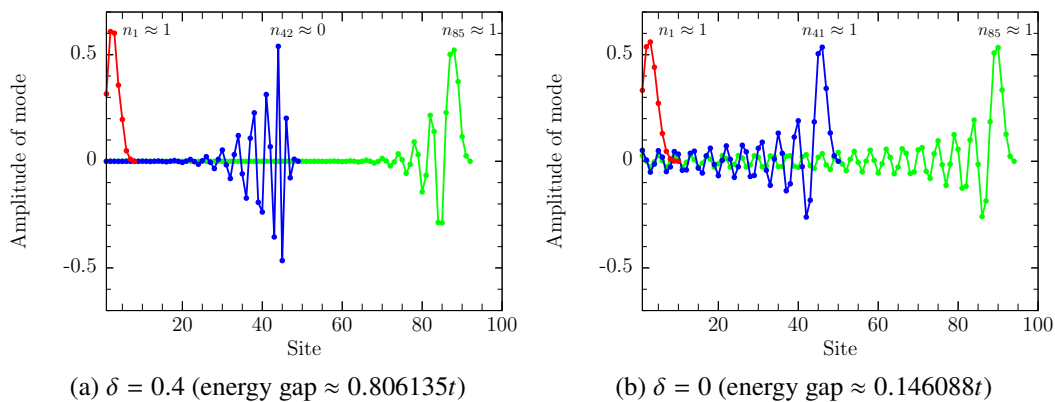


Figure 212: Examples of occupied and unoccupied modes found in the diagonalization process. Fig. 212(a) shows occupied/unoccupied modes for $\delta = 0.4$ (energy gap $\approx 0.806135t$). Fig. 212(b) shows occupied/unoccupied modes for $\delta = 0$ (energy gap $\approx 0.146088t$).

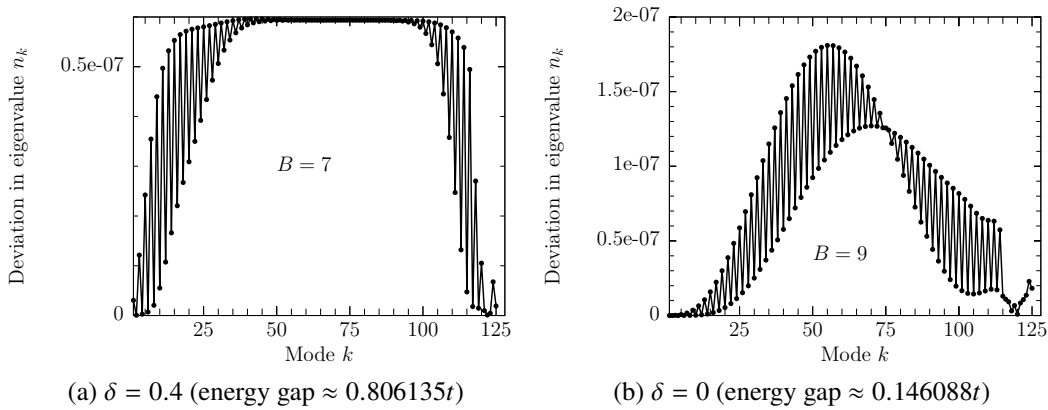


Figure 213: Examples of deviations in occupations at the end of the diagonalization procedure for $N = 128$ sites. Fig. 213(a) shows errors in the occupations for $\delta = 0.4$ (energy gap $\approx 0.806135t$). Fig. 213(b) shows errors in the occupations for $\delta = 0.0$ (energy gap $\approx 0.146088t$).

Fig. 212 shows examples of the modes obtained with the procedure, both filled and unfilled, for a small gap and a larger gap. The modes are seen to be localized for the case of the larger gap, and extend throughout the system for the smaller gap. The unfilled modes follow the same decay as the filled modes but oscillate more, since they are above the Fermi sea and are therefore higher in energy. Fig. 213 shows for the same two gaps the deviation in the eigenvalues n_k from 0 or 1 obtained during the diagonalization procedure. For the case of the larger gap, this error saturates to its maximum quickly for modes near the middle of the system, while for the smaller gap, the error increases more slowly due to the longer range correlations.

In Fig. 214 we analyze the block size scaling with system size N for the gapless case ($\delta = 0$). As we expect from arguments about entanglement made at the end of Section 2.3.1, the scaling is found to be $B \sim \log(N)$. This is the expected scaling for a critical 1D system. We can see that with this procedure we can analyze very large systems, up to $N = 2^{16} = 65536$ sites, even for gapless free fermions. To avoid storing correlation matrices this large, we begin with a very accurate compressed correlation matrix as a GMPS using the GDMRG algorithm presented in Appendix 2.B. With GDMRG, we begin with a state with a relative error in the energy of $< 10^{-10}$. For $N = 65536$ this requires a block size of $B = 22$. We then obtain the local correlation matrix for the block we are interested in using the gates from this accurate compression, and use it to obtain a less accurate compression with a smaller block size. This procedure should lead to, for a given block size, a more accurate overall state than one that would be obtained directly

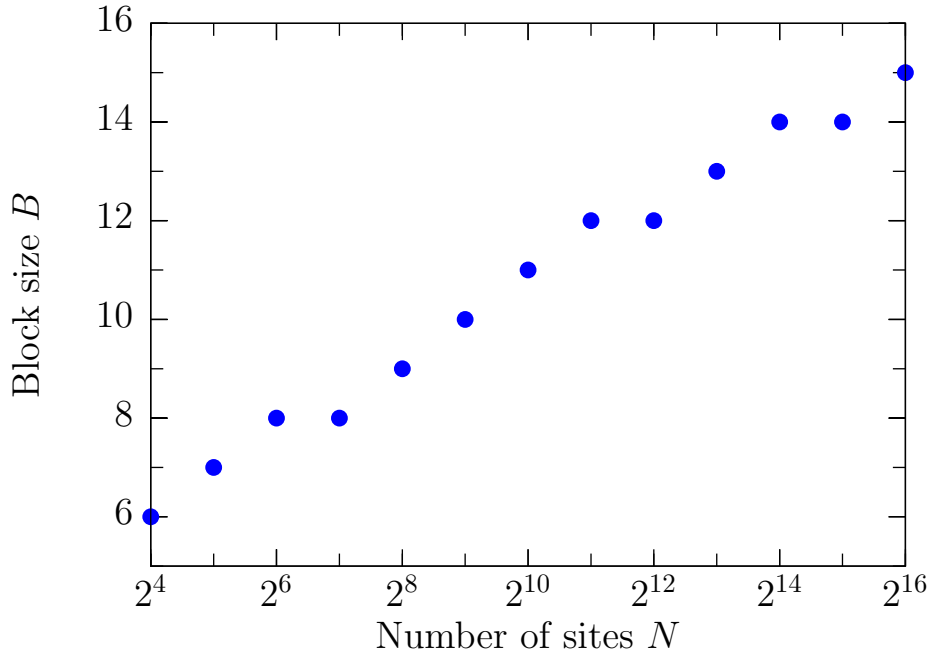


Figure 214: Block size B needed for a relative error in the energy of $< 10^{-6}$ as a function of number of sites N for spinless, gapless fermions with open boundary conditions at half filling. As expected from arguments about the entanglement of a critical system, we find $B \sim \log(N)$, tested up to $N = 2^{16} = 65536$ sites (note the log scale on the x axis). To study systems of this size and avoid the $O(N^3)$ diagonalization of the hopping Hamiltonian, we obtain the correlation matrix using the GDMRG algorithm as explained in Appendix 2.B.

from GDMRG, because GDMRG optimizes the energy which only depends on very local correlations.

2.4.2 GMERA Results

Here we present results for compressing a correlation matrix as a GMERA using the procedure presented in Section 2.3.2. We show the relative error in the energy for increasing number of sites for $B = 10$ in Fig. 215. We see that for this block size, the error stays below 10^{-6} for systems up to $N = 2^{14} = 16384$ and in fact appears to saturate at high number of sites (the change in the relative error in the energy approaches 0 for larger system sizes). This is in stark contrast to the GMPS, where a block size $B \sim \log(N)$ was required to obtain a fixed accuracy, as shown in Fig. 214. Instead, the GMERA obtains the same accuracy with constant block size B as shown in Fig. 215. The GMPS obtains the given accuracy with the same or smaller block size up to $N = 512$, after which it requires a larger block size than the GMERA to obtain the same level of accuracy. As we mentioned earlier, this is

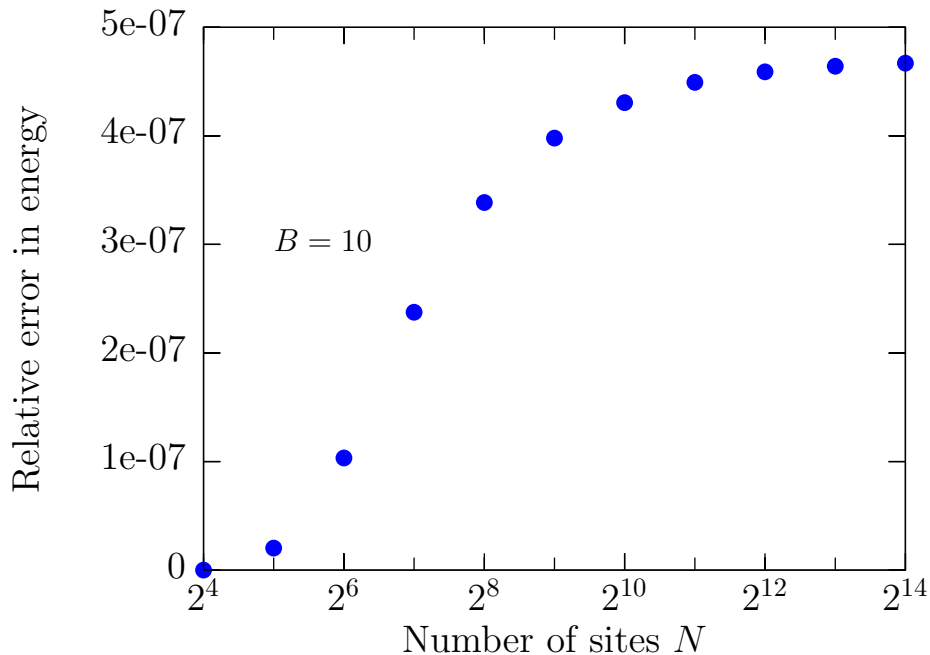


Figure 215: Relative error in the energy for the proposed GMERA construction for increasing number of sites for a block size $B = 10$. The system analyzed is the ground state of free fermions hopping on a lattice with open boundary conditions. All errors are below 10^{-6} . As expected for a MERA, the error is seen to saturate for large N , indicating a fixed block size is sufficient to obtain an accuracy $< 10^{-6}$ up to very large system sizes.

made possible partially because the GMERA structure involves nonlocal gates.

2.4.3 GMPS to Many-Body MPS Results

Plots of the time it takes to form the MPS of the ground state of a gapless free fermion system for up to $N = 1024$ sites using the method presented in Section 2.3.4 are shown in Fig. 216. As expected, the time it takes for a gapless system is polynomial in the system size N , while it is approximately linear in N for a gapped system. The SSH model is used with $\delta = 0.1$ or an energy gap $\Delta \approx 0.2t$. The time to form the gapless ground state is only a modest polynomial in N , $\sim N^{2.03}$, while as we expect from arguments about entanglement the time to form the gapped ground state is very nearly linear in N , $\sim N^{1.02}$, because the block size and bond dimension required to obtain the specified accuracy are constant for all N shown ($B = 8$ and $\chi = 55$). With this method, a gapless ground state of $N = 1025$ sites with a relative error in the energy of $< 10^{-6}$ can be formed on a laptop in only ~ 90 seconds.

An interesting point to emphasize is the quality of the compression. The GMPS

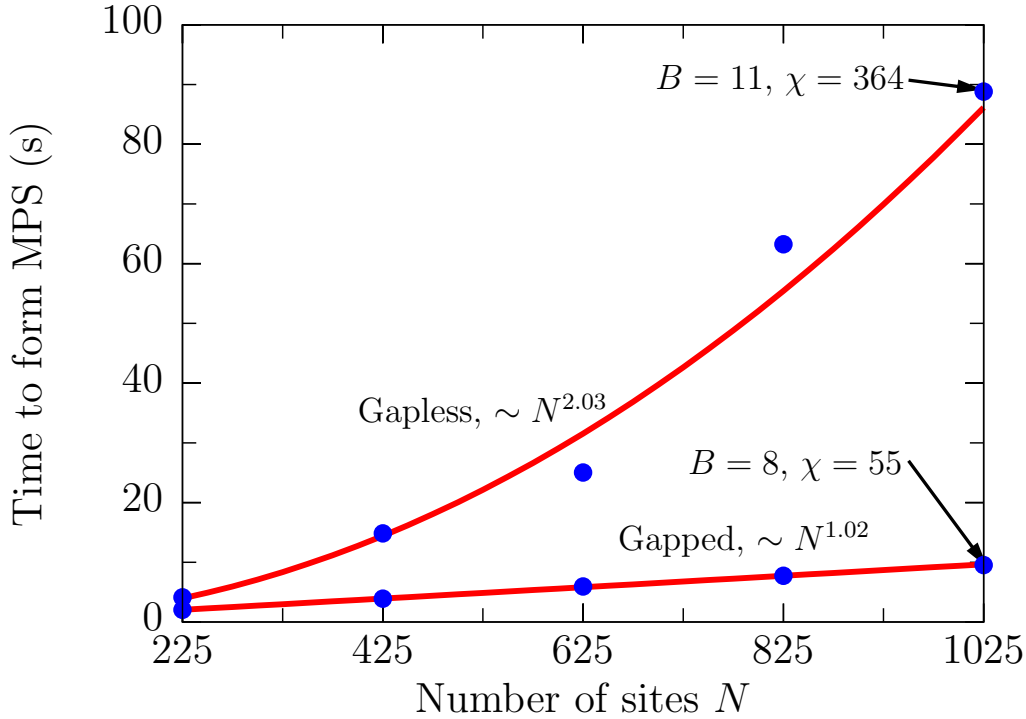


Figure 216: A plot of the time to form the MPS approximation of gapped and gapless free fermion ground states at half filling as a function of sites N using gates from a GMPS. The bond dimensions are chosen large enough such that the relative errors in the energy of the MPS are below 10^{-6} . The block size of the GMPS used to form the MPS are the minimum required to obtain a GMPS with a relative error in the energy of 10^{-6} . A cutoff in the singular values of the SVD of 10^{-11} was used when applying the gates to form the MPS using the method described in Section 2.3.4. For the gapped case, the SSH model with $\delta = 0.1$ is used, corresponding to an energy gap of $\Delta \approx 0.2t$ (exact as $N \rightarrow \infty$).

for the gapless ground state on $N = 1025$ sites requires a block size of $B = 11$ to obtain a relative error in the energy of $< 10^{-6}$. Naively, turning these gates into many-body gates and contracting the network (forming the MPS directly from the GMPS with no truncation) as explained earlier leads to a bond dimension of the MPS of $\chi = 2^{B-1} = 2^{10} = 1024$. However, applying the gates as described and using a cutoff of the singular values of 10^{-11} leads to the formation of an MPS approximation of the fermionic Gaussian ground state, still with a relative error in the energy of $< 10^{-6}$, with a bond dimension of only $\chi = 364$. This is a result of the fact that our GMPS only explores the manifold of fermionic Gaussian states limited to the specified block size. On the other hand, the MPS approximation of the Gaussian state we form from this GMPS is able to explore the entire manifold of MPSs up to the allowed bond dimension (and particle number if symmetric tensors

are used, as we do here), so through the SVD we are able to compress the state quite efficiently beyond what we initially might expect.

2.5 Conclusion

We have presented an efficient, numerically stable, and controllably accurate way to compress a correlation matrix into a set of 2×2 unitary gates. From these gates, we have also presented a method for easily and efficiently forming the MPS approximation of a fermionic Gaussian state. We explained the procedure in detail for the ground state of a generic number conserving Hamiltonian. We then presented results for the SSH model, a 1D chain of fermions with staggered hopping. We showed examples of the accuracy and block sizes needed for different gaps of the model. We hope this method can be used as a simple, efficient and reliable procedure for directly preparing many states of interest, either by creating starting states to aid DMRG calculations or preparing a particular ansatz as an MPS. We also presented one example of how the procedure can be modified to obtain different gate structures, in this case one that is related to the MERA. However, there are other possibilities to be explored, such as gate structures more directly suited for systems with 2 spatial dimensions, periodic boundary conditions, as well as how the method might be applied to study thermal fermionic Gaussian states. In addition, we presented how discrete wavelet transforms can be described very simply with the gate structure notation we introduced in this paper.

The method is easily generalized to cases beyond the one presented here. As we touched upon earlier, it can be generalized to the case of BCS states, the ground states of hopping Hamiltonians that include pairing terms. In this case, the correlation matrix in the Majorana basis can be written in terms of an anti-symmetric matrix which can be approximately block diagonalized by $\sim 5BN$ local 2×2 rotation gates, which are turned into nearest neighbor parity-conserving many-body gates. The case of spinless fermions was presented, but spinful fermions are a simple generalization. In addition, we expect similar methods as those presented can be used to study and compress bosonic Gaussian states. In this case, one could form the covariance matrix of the bosonic Gaussian state and locally diagonalize it to find the uncorrelated bosonic modes (see [25] for a previous study of bosonic Gaussian MERA). Additionally, more complicated many-body gates would be required because the local Hilbert space dimension is larger for bosons.

APPENDIX

2.A Calculation of the Entanglement Entropy of a Fermionic Gaussian State

In this section we give a simple, self-contained derivation for Eq. 2.6, the entanglement entropy for a block of a free fermion system. This expression has been shown elsewhere[12–15], though we show a simple, self-contained derivation here.

Assume the block of interest \mathcal{B} is the first B sites. Gaussian states have expectation values that obey Wick's theorem. This means that the expectation value of any operator contained within the block is specified if we know subblock \mathcal{B} of the correlation matrix, $\Lambda_{\mathcal{B}}$. This implies that the many-body density matrix of the block $\hat{\rho}_{\mathcal{B}}$ is also uniquely specified by $\Lambda_{\mathcal{B}}$. The entanglement entropy on block \mathcal{B} , defined as $S_{\mathcal{B}}[\hat{\rho}_{\mathcal{B}}] = -\text{Tr}[\hat{\rho}_{\mathcal{B}} \log(\hat{\rho}_{\mathcal{B}})]$, does not change under general unitary transformations within the block. Thus, we can perform the single particle unitary transformation of basis that makes $\Lambda_{\mathcal{B}}$ diagonal, with diagonal entries $n_b = \langle \hat{a}_b^\dagger \hat{a}_b \rangle$ for $b \in 1, \dots, B$. The n_b uniquely specify the reduced density matrix of the block, so the entanglement is a universal function of $\{n_b\}$:

$$S_B = S_B(n_1, \dots, n_B). \quad (2.12)$$

In fact, the details of the system outside the block are irrelevant. For example, different systems with different numbers of sites outside the block can have the same S_B as long as their $\{n_b\}$ are identical and the system is a Gaussian state. Thus to evaluate the function S_B , we can choose a simple system in which to evaluate it rather than using the actual system of interest.

Let's first consider a block with only one site ($B = 1$). We would like to know the universal function $S_1(n_1)$. We choose a two site system containing a single particle, with normalized wavefunction

$$|\psi\rangle = (\sqrt{n_1} \hat{a}_1^\dagger + \sqrt{1-n_1} \hat{a}_2^\dagger) |\Omega\rangle. \quad (2.13)$$

The correlation matrix is

$$\begin{pmatrix} n_1 & \sqrt{n_1(1-n_1)} \\ \sqrt{n_1(1-n_1)} & 1-n_1 \end{pmatrix}, \quad (2.14)$$

which has the required block correlation matrix $\Lambda_1 = (n_1)$. The Schmidt decomposition of $|\psi\rangle$ is

$$\begin{aligned} |\psi\rangle &= \sqrt{1-n_1} (|0\rangle)(\hat{a}_2^\dagger |0\rangle) + \sqrt{n_1} (\hat{a}_1^\dagger |0\rangle)(|0\rangle) \\ &= \sqrt{1-n_1} |0\rangle |1\rangle + \sqrt{n_1} |1\rangle |0\rangle, \end{aligned} \quad (2.15)$$

where $|1\rangle$ is the occupied state of the corresponding site. From this we see that the reduced density matrix for site 1, $\hat{\rho}_1 = \text{Tr}_2[|\psi\rangle\langle\psi|]$, is

$$\begin{aligned}\hat{\rho}_1 &= (1 - n_1) |0\rangle\langle 0| + n_1 |1\rangle\langle 1| \\ &= (1 - n_1)(\hat{I}_1 - \hat{n}_1) + n_1 \hat{n}_1,\end{aligned}\tag{2.16}$$

and

$$\begin{aligned}S_1(n_1) &= -\text{Tr}[\hat{\rho}_1 \log(\hat{\rho}_1)] \\ &= -(1 - n_1) \log(1 - n_1) - n_1 \log(n_1).\end{aligned}\tag{2.17}$$

For the system \mathcal{B} with block size $B > 1$, we can choose the system to be of size $2B$ and for each site in the block associate one site outside the block. The Gaussian state is the product state of the single particle states living on a pair, each identical in form to the $B = 1$ state, with B total particles. This system has no correlations or entanglement between these pairs. This means that the entanglement is the sum of the entanglement of each pair. Thus

$$S_B(\{n_b\}) = \sum_{b \in \mathcal{B}} S_1(n_b),\tag{2.18}$$

which is identical to Eq. 2.6. Note also that the overall reduced density matrix of the block is the product of the single site density matrices given in Eq. 2.16.

An alternative argument can be made to derive the same equation which avoids the introduction of a contrived environment. We could have taken as an ansatz that the reduced density matrix on \mathcal{B} , $\hat{\rho}_{\mathcal{B}}$, is the product of the single site reduced density matrix we derived in Eq. 2.16, in other words $\hat{\rho}_{\mathcal{B}} = \otimes_{b \in \mathcal{B}} \hat{\rho}_b$. We can show that this is in fact the unique reduced density of the state we are interested in if we can show that it reproduces the correct correlation matrix of our state and is a fermionic Gaussian state (that it obeys Wick's theorem). Both of these are easy to show explicitly. Once we verify that this is indeed the correct reduced density matrix of our state, we can calculate the entanglement entropy directly with $S_B = -\text{Tr}[\hat{\rho}_{\mathcal{B}} \log(\hat{\rho}_{\mathcal{B}})] = -\sum_{b \in \mathcal{B}} \text{Tr}[\hat{\rho}_b \log(\hat{\rho}_b)]$, which matches Eq. 2.6.

2.B GDMRG, an Algorithm to Obtain a Compressed Ground State Correlation Matrix as a GMPS

Here we describe fermionic Gaussian DMRG (GDMRG), a DMRG-like algorithm in the single particle context. The algorithm is an efficient method to directly obtain all the angles specifying the compressed correlation matrix as a GMPS without ever

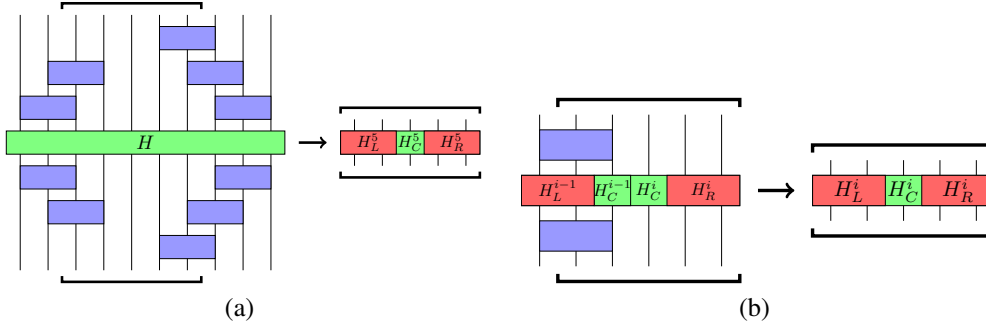


Figure 2.B1: Fig. 2.B1(a) shows an example of an effective Hamiltonian centered at site 5 for the GDMRG algorithm. The effective Hamiltonian is the submatrix of the indicated sites. The example is for $N = 10$ sites and a block size of $B = 3$. Here the center is only one site, but could be more to improve convergence just like in the standard DMRG algorithm. Fig. 2.B1(b) shows, for a sweep to the right, how to obtain the new left block from the previous effective Hamiltonian by rotating with the appropriate gates found and taking the submatrix of the indicated sites.

needing to express the matrix in uncompressed form. The ground state GMPS of a hopping Hamiltonian on N sites is calculated with a cost of only $O(B^3N)$, where B is the block size of the GMPS (which determines the accuracy of the compression and depends on the entanglement of the ground state).

Imagine that we start with a hopping Hamiltonian H on a lattice of N sites, and we would like to obtain the GMPS with a block size B that minimizes the energy of H . We begin with a random starting GMPS. Just like in the DMRG algorithm, we form an effective Hamiltonian centered at a site with a left and right block, which we show in Fig. 2.B1(a). Say that we start on the left side of the lattice and begin sweeping right. Our GMPS will start gauged to the left. For a single-site GDMRG, our center block is only one site, but we could use a larger center to decrease the number of sweeps required for convergence. In practice for free gapless fermions we find that a single center site works quite well. The first step is to diagonalize the $2B - 1$ site effective Hamiltonian, and obtain the effective correlation matrix Λ_{eff} . Using this Λ_{eff} , we diagonalize the first $B \times B$ block and, for a large enough B , find a fully occupied or unoccupied mode. Just as described in Fig. 24, we find the $B - 1$ nearest neighbor 2×2 gates that rotate Λ_{eff} into the basis containing this mode, partially diagonalizing it. These gates form the first block of the GMPS.

Next we would like to move the center to the right so that we can obtain the next block of the GMPS. Because the compression is a unitary transformation, we can start moving the center to the right by undoing the gates in the block of the GMPS

to the right of the center. This step is in contrast to ordinary DMRG where a sequence of right blocks are stored and are called from memory when needed. We then obtain the effective Hamiltonian for the next site using the block of the GMPS we obtained from the previous effective Hamiltonian of the sweep, as shown in Fig. 2.B1(b). We repeat this process until we reach the end of the lattice, completing our first sweep. We continue sweeping back and forth until the energy is sufficiently converged. We use this algorithm to obtain a very accurate correlation matrix for systems up to $N = 2^{16} = 65536$, from which we obtain the GMPS in Fig. 214. For $N = 65536$ sites to obtain a correlation matrix with a relative error in the energy of less than 10^{-10} , we require a block size of $B = 22$ and 14 sweeps (where a single sweep is from left to right or right to left).

VARIATIONAL ALGORITHMS FOR MATRIX PRODUCT STATES DIRECTLY IN THE THERMODYNAMIC LIMIT

¹V. Zauner-Stauber, L. Vanderstraeten, M. T. Fishman, F. Verstraete, and J. Haegeman, “Variational optimization algorithms for uniform matrix product states”, *Phys. Rev. B* **97**, 045145 (2018).

Here we present a new algorithm called the variational uniform matrix product state (VUMPS) algorithm, which is a DMRG-like algorithm that works directly in the thermodynamic limit. The algorithm uses the ansatz of a uniform matrix product state (uMPS), and explicitly optimizes that variational state. This is in contrast to the infinite DMRG (iDMRG) algorithm, which is a DMRG algorithm that reaches the thermodynamic limit by growing the system size at each step, and the infinite time evolving block decimation (iTEBD) algorithm, which works with a uMPS in the thermodynamic limit but optimizes the state with a power method instead of variationally. Careful benchmarks in Ref. [2] show that the VUMPS algorithm performs significantly better for finding ground states compared to both iDMRG and iTEBD, the state-of-the-art algorithms in the field, for a wide range of 1D and quasi-1D systems. Here we present the most basic form of the algorithm applied to one-dimensional (1D) quantum states with nearest neighbor interactions.

3.1 Introduction

The strategy of renormalization group (RG) techniques to successively reduce a large number of microscopic degrees of freedom to a smaller set of effective degrees of freedom has led to powerful numerical and analytical methods to probe and understand the effective macroscopic behavior of both classical and quantum many-body systems [26–29]. However, it was not until the advent of White’s celebrated density matrix renormalization group (DMRG) [5, 6] that variational RG methods reached unprecedented accuracy in numerically studying strongly correlated one-dimensional quantum lattice systems at low temperature. The underlying variational ansatz of matrix product states (MPS) [30–36] belongs to a class of ansatzes known as tensor network states [34, 37, 38]. These variational classes encode the many-body wavefunction in terms of virtual entanglement degrees of freedom living

on the boundary, and thus satisfy an area law scaling of entanglement entropy by construction. As such, they provide a natural parameterization for the physical corner of Hilbert space, where low energy states of quantum many-body systems ought to live in [39, 40]. MPS in particular are especially fit for studying ground states of strongly correlated one-dimensional quantum systems with local interactions [41–43].

The variational parameters in MPS are contained within local tensors associated with the individual sites of the lattice system. For homogeneous systems, the global wave function can then be captured using just a single (or a small number of) such tensors, independent of the system size. They consequently offer very natural access to the thermodynamic limit, providing a clear advantage over other numerical approaches such as Exact Diagonalization or Quantum Monte Carlo.

On finite lattices, (one-site) DMRG implements the variational principle (energy minimization) by exploiting that the quantum state is a multilinear function of the local tensors. By fixing all but one tensor, the global eigenvalue problem is transformed into an effective eigenvalue problem for the local tensor [5, 6, 35, 44–46]. Using a translation invariant parameterization gives rise to an energy expectation value with a highly nonlinear dependence on the tensor(s). Two different algorithms are widely used to obtain such an MPS in the thermodynamic limit. Infinite system DMRG (iDMRG) [5, 6, 47] proceeds by performing regular DMRG on a successively growing lattice, inserting and optimizing over new tensors in the center of the lattice in each step only, effectively mimicking an infinite lattice by using a finite, albeit very large lattice. After convergence, the most recently inserted tensors in the center are taken as a unit cell for an infinite MPS approximation of the ground state. An alternative approach is known as the infinite time evolving block decimation (iTEBD) [48, 49] algorithm. It works directly in the thermodynamic limit and is based on evolving an initial state in imaginary time by using a Trotter-Suzuki decomposition of the evolution operator.

We present a new variational algorithm, inspired by tangent space ideas [36, 50, 51] that combines the advantages of iDMRG and iTEBD and addresses some of their shortcomings. As such, it is directly formulated in the thermodynamic limit, but at the same time optimizes the state by solving effective eigenvalue problems, rather than employing imaginary time evolution. Benchmark results shown in Ref. [2] show a significant increase in efficiency for a wide range of test cases. The following section introduces MPS notations and definitions and presents our

variational algorithm, heuristically motivated from the perspective of finite size DMRG.

3.2 A Variational Algorithm for Matrix Product States in the Thermodynamic Limit

In this section, we introduce a variational algorithm for optimizing a MPS directly in the thermodynamic limit. Because the algorithm strongly resembles conventional DMRG, we explain it by describing a single iteration step from the viewpoint of DMRG and show that only a few additional ingredients are needed to arrive at our variational algorithm. Here we limit the presentation to the simplest case of a 1D quantum system with nearest neighbor interactions, and refer readers to Ref. [2] for extensions to more general Hamiltonians as well as more rigorous theoretical motivations.

We start by considering a setting familiar from conventional DMRG: a finite homogeneous one-dimensional quantum lattice system, where every site corresponds to a d level system. We label the sites by an integer n , and thus have a basis $\{|s\rangle_n, s = 1, \dots, d\}$ for the local Hilbert space on site n . The total Hilbert space is spanned by the product basis $|s\rangle = \bigotimes_n |s\rangle_n$. We assume the dynamics of the system to be governed by a translation invariant Hamiltonian H .

We further consider a variational parameterization of a ground state approximation of the system, for now in terms of a finite size (site dependent) MPS, but we will ultimately be interested in the thermodynamic limit. In DMRG, one finds the best variational ground state approximation by employing an alternating least squares minimization. Starting from some initial state, one successively optimizes each of the individual MPS tensors site by site by solving effective (Hamiltonian) eigenvalue problems, in a sweeping process through the lattice until convergence, where each iteration depends on already optimized tensors from previous iterations (see e.g. Refs [5, 6, 35, 44, 46]).

We are now, however, interested in the thermodynamic limit $n \in \mathbb{Z}$. In that case, the MPS ground state approximation will be given in terms of a translation invariant uniform MPS, described by a single MPS tensor (or a unit cell of N tensors), repeated on all sites. Two immediate difficulties arise. First, conventional DMRG updates the variational state site by site, thus breaking translation invariance. Second, the effective Hamiltonian for a single-site optimization has to be constructed from an infinite environment. Our goal here is to present an algorithm that overcomes these

two challenges.

After briefly introducing the variational class of uniform MPS and introducing necessary notation and conventions, we describe how the new algorithm modifies DMRG accordingly to exactly account for these two issues in order to arrive at a variational ground state algorithm directly formulated in the thermodynamic limit.

3.3 Uniform MPS

A uniform MPS (uMPS) of bond dimension D defined on an infinite translation invariant lattice is parameterized by a single collection of d matrices $A^s \in \mathbb{C}^{D \times D}$ for $s = 1, \dots, d$. The overall translation invariant variational state is then given by

$$|\Psi(A)\rangle = \sum_{\mathbf{s}} \left(\dots A^{s_{n-1}} A^{s_n} A^{s_{n+1}} \dots \right) |\mathbf{s}\rangle, \quad (3.1)$$

and can be represented diagrammatically as

$$|\Psi(A)\rangle = \dots \text{---} \boxed{A} \text{---} \boxed{A} \text{---} \boxed{A} \text{---} \boxed{A} \text{---} \boxed{A} \text{---} \dots$$

Exploiting the invariance of (3.1) under local gauge transformations $A^s \rightarrow X A^s X^{-1}$, with $X \in \mathbb{C}^{D \times D}$ invertible, the state can be cast into certain favorable representations, among them the *left and right canonical representation*

$$\sum_s A_L^{s \dagger} A_L^s = \mathbb{1} \quad \sum_s A_L^s R A_L^{s \dagger} = R \quad (3.2a)$$

$$\sum_s A_R^s A_R^{s \dagger} = \mathbb{1} \quad \sum_s A_R^{s \dagger} L A_R^s = L, \quad (3.2b)$$

or diagrammatically

Here, L and R correspond to the left and right reduced density matrices of a bipartition of the state respectively. We henceforth refer to A_L (A_R) as a left (right) isometric tensor, or just a left (right) isometry.

Defining the left and right transfer matrices

$$T_{L/R} = \sum_s \bar{A}_{L/R}^s \otimes A_{L/R}^s \quad (3.3)$$

and using the notation $(x|$ and $|x)$ for vectorizations of a $D \times D$ matrix x in the D^2 -dimensional “double layer” virtual space the transfer matrices act upon, the gauge conditions (3.2) are equivalent to

$$(\mathbb{1}|T_L = (\mathbb{1}| \quad T_L|R) = |R) \quad (3.4a)$$

$$T_R|\mathbb{1}) = |\mathbb{1}) \quad (L|T_R = (L|, \quad (3.4b)$$

i.e. $\mathbb{1}$ and R are the left and right dominant eigenvectors of T_L , while L and $\mathbb{1}$ are the left and right dominant eigenvectors of T_R .

As is standard practice in DMRG, we mix both of these representations and cast the state into the *mixed canonical representation*

$$|\Psi(A)\rangle = \sum_s (\dots A_L^{s_{n-1}} A_C^{s_n} A_R^{s_{n+1}} \dots) |s\rangle \quad (3.5a)$$

$$= \sum_s (\dots A_L^{s_{n-1}} A_L^{s_n} C A_R^{s_{n+1}} A_R^{s_{n+2}} \dots) |s\rangle, \quad (3.5b)$$

or diagrammatically,

$$\begin{aligned} |\Psi(A)\rangle &= \dots \text{---} \textcircled{A_L} \text{---} \textcircled{A_L} \text{---} \textcircled{A_C} \text{---} \textcircled{A_R} \text{---} \textcircled{A_R} \text{---} \dots \\ &= \dots \text{---} \textcircled{A_L} \text{---} \textcircled{A_L} \text{---} \textcircled{A_L} \text{---} \textcircled{C} \text{---} \textcircled{A_R} \text{---} \textcircled{A_R} \text{---} \dots \end{aligned}$$

Here we have defined the *center site tensor* A_C^s (known as the single-site wave function Ψ^s in DMRG)

$$\begin{aligned} A_C^s &= A_L^s C = C A_R^s \\ \text{---} \textcircled{A_C} \text{---} &= \text{---} \textcircled{A_L} \text{---} \textcircled{C} \text{---} = \text{---} \textcircled{C} \text{---} \textcircled{A_R} \text{---} \end{aligned} \quad (3.6)$$

in terms of the bond matrix C , which constitutes the (invertible) gauge transformation relating A_L and A_R via $A_L^s = C A_R^s C^{-1}$. The singular values of C then encode the entanglement spectrum of the state. Indeed, using $A_L^s C = C A_R^s$ we can verify that the left and right reduced density matrices in (3.2) are given by $L = C^\dagger C$ and $R = C C^\dagger$. Furthermore, $A_L^s C = C A_R^s$ ensures that (3.5a) and (3.5b) are translation invariant and that A_C and C can be shifted around arbitrarily. Normalization of the

state, as well as of the reduced density matrices L and R , corresponds to the single condition $\|C\|_2^2 = \text{Tr}(CC^\dagger) = 1$.

For ease of notation, we further introduce the following partial states

$$|\Psi_L^\alpha\rangle = \sum_s (\dots A_L^{s_{n-1}} A_L^{s_n})_\alpha |\dots s_{n-1} s_n\rangle \quad (3.7a)$$

$$= \dots \text{---} \boxed{A_L} \text{---} \boxed{A_L} \text{---} \boxed{A_L} \text{---} \alpha$$

$$|\Psi_R^\alpha\rangle = \sum_s (A_R^{s_n} A_R^{s_{n+1}} \dots)_\alpha |s_n s_{n+1} \dots\rangle \quad (3.7b)$$

$$= \alpha \text{---} \boxed{A_R} \text{---} \boxed{A_R} \text{---} \boxed{A_R} \text{---} \dots$$

with n arbitrary, and use them to define the reduced basis states

$$|\Psi_{AC}^{(\alpha,s,\beta)}\rangle = |\Psi_L^\alpha\rangle |s\rangle |\Psi_R^\beta\rangle \quad (3.8a)$$

$$|\Psi_C^{(\alpha,\beta)}\rangle = |\Psi_L^\alpha\rangle |\Psi_R^\beta\rangle. \quad (3.8b)$$

3.4 Effective Hamiltonian

The use of the mixed canonical representation (3.5a) in DMRG is of significant importance for the stability of the algorithm. With this gauge choice, the minimization of the (global) energy expectation value $\langle\Psi|H|\Psi\rangle / \langle\Psi|\Psi\rangle$ with respect to A_C is reduced to a standard (Hermitian) eigenvalue problem, instead of a generalized eigenvalue problem. The effective Hamiltonian for this eigenvalue problem is the system Hamiltonian H projected onto the degrees of freedom of A_C , and is known as the “reduced” or “superblock” Hamiltonian in DMRG.

We define the thermodynamic limit version of this reduced single-site Hamiltonian acting on A_C as

$$H_{AC}^{(\alpha',s',\beta')} = \langle\Psi_{AC}^{(\alpha',s',\beta')}\rangle |H| \Psi_{AC}^{(\alpha,s,\beta)}\rangle \quad (3.9)$$

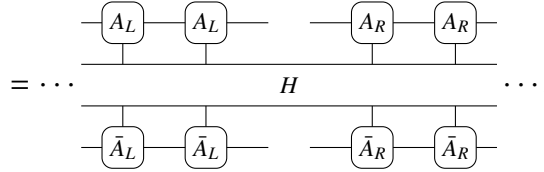
$$= \dots \text{---} \boxed{A_L} \text{---} \boxed{A_L} \text{---} \boxed{A_R} \text{---} \boxed{A_R} \text{---} \dots$$

$$\text{---} \boxed{\bar{A}_L} \text{---} \boxed{\bar{A}_L} \text{---} \boxed{\bar{A}_R} \text{---} \boxed{\bar{A}_R} \text{---} \dots$$

Additionally, we also define an effective Hamiltonian acting on the bond matrix C

as

$$H_C^{(\alpha',\beta')} = \langle \Psi_C^{(\alpha',\beta')} | H | \Psi_C^{(\alpha,\beta)} \rangle \quad (3.10)$$



which does not appear directly in the context of DMRG, but will be needed later for a consistent update of the state without breaking translation invariance. It can be interpreted as a “zero site” effective Hamiltonian, which would feature in an optimization of the global energy expectation value with respect to the Schmidt values.

In an efficient implementation, these effective eigenvalue problems are typically solved using an iterative eigensolver, so that we only need to implement the action of H_{AC} and H_C onto A_C and C .

While the energy expectation value is extensive and thus divergent in the thermodynamic limit, the effective Hamiltonians H_{AC} and H_C are well-defined and finite in the thermodynamic limit if one properly subtracts the current energy expectation value from the Hamiltonian H . We demonstrate this procedure for the case of nearest neighbor interactions $H = \sum_n h_{n,n+1}$, where the two-site Hamiltonian $h_{n,n+1}$ acts on neighboring sites $n, n+1$ only. We refer to Ref. [2] for the case of long range interactions and for general Hamiltonians given in terms of Matrix Product Operators (MPOs).

In the case of nearest neighbor interactions, the action of H_{AC} onto A_C splits up into four individual contributions, which follow from the decomposition $|\Psi\rangle = \sum_{\alpha,\beta,s} A_{C,(\alpha,\beta)}^s |\Psi_L^\alpha\rangle |s\rangle |\Psi_R^\beta\rangle$ (left block containing sites $n < 0$, center site $n = 0$, and right block containing sites $n > 0$). The action of H_{AC} onto A_C is given by

$$A_C'^s = \sum_{tk\ell} h_{k\ell}^{ts} A_L^t A_L^\dagger A_C^k A_C^\ell + h_{k\ell}^{st} A_C^k A_R^\ell A_R^t A_R^\dagger + H_L A_C^s + A_C^s H_R$$

$$(3.11)$$

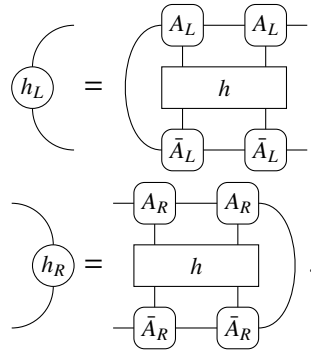
where the first two terms correspond to the Hamiltonian terms $h_{-1,0}$ and $h_{0,1}$ coupling the center site to the left and right block, respectively, and H_L and H_R sum up the contributions of all the Hamiltonian terms $h_{n,n+1}$ acting strictly to the left and to the right of the center site.

The environments H_L and H_R are usually constructed iteratively while sweeping through the (finite) lattice in conventional DMRG, or grown successively in every iteration in iDMRG. In the thermodynamic limit, these terms consist of a diverging number of individual local interaction contributions $h_{n,n+1}$, and care needs to be taken in their construction.

Indeed, the k^{th} contribution to $\langle H_L |$ comes from the Hamiltonian term $h_{-k-1,-k}$ and is given by $\langle h_L | [T_L]^{k-1}$. Likewise, $[T_R]^{k-1} | h_R \rangle$ is the k^{th} contribution to $| H_R \rangle$ stemming from $h_{k,k+1}$. Here, we have used the definitions

$$\begin{aligned} h_L &= \sum_{stkl} h_{k\ell}^{st} A_L^t \dagger A_L^s \dagger A_L^k A_L^\ell \\ h_R &= \sum_{stkl} h_{k\ell}^{st} A_R^k A_R^\ell A_R^t \dagger A_R^s \dagger, \end{aligned} \quad (3.12)$$

or diagrammatically



Summing up all such local contributions gives rise to infinite geometric sums of the transfer matrices $T_{L/R}$

$$\langle H_L | = \langle h_L | \sum_{k=0}^{\infty} [T_L]^k \quad | H_R \rangle = \sum_{k=0}^{\infty} [T_R]^k | h_R \rangle, \quad (3.13)$$

where $\langle H_L |$ can be presented diagrammatically as

$$\langle H_L | = \langle h_L | \left[\mathbb{1} + \begin{array}{c} \text{---} A_L \text{---} \\ | \\ \text{---} \bar{A}_L \text{---} \end{array} + \begin{array}{c} \text{---} A_L \text{---} A_L \text{---} \\ | \\ \text{---} \bar{A}_L \text{---} \bar{A}_L \text{---} \end{array} + \dots \right]$$

and likewise for $|H_R\rangle$.

The transfer matrix T_L has a dominant eigenvalue of magnitude one, with corresponding left and right eigenvectors $(\mathbb{1}|$ and $|R\rangle$. The projection $(h_L|[T_L]^k|R) = (h_L|R)$ is the energy density expectation value $e = \langle\Psi|h_{-k-1,-k}|\Psi\rangle$ and is independent of k . Subtracting the energy e from the Hamiltonian as $\tilde{h} = h - e\mathbb{1}$, we can write $(h_L| = (\tilde{h}_L| + e(\mathbb{1}|$. The second term is exactly proportional to the left eigenvector of eigenvalue 1, and therefore gives rise to a diverging contribution in the geometric sum, corresponding to the total energy of the left half infinite block. Since this contribution acts as the identity in the effective Hamiltonian H_{A_C} [Eq. (3.11)], we can however safely discard this diverging contribution without changing the eigenvectors of H_{A_C} . This corresponds to an overall energy shift of the left half infinite block such that $(H_L|R) = 0$. For the remaining part $(\tilde{h}_L|$ the geometric sum converges. With $(\tilde{h}_R) = (h_R) - e(\mathbb{1})$ the same comments apply to the construction of $|H_R\rangle$.

We can evaluate H_L and H_R recursively as

$$\begin{aligned} (H_L^{[n+1]}| &= (H_L^{[n]}|T_L + (\tilde{h}_L| \\ |H_R^{[n+1]}) &= T_R|H_R^{[n]}) + (\tilde{h}_R) \end{aligned} \quad (3.14)$$

with initialization $(H_L^{[0]}| = (\tilde{h}_L|$ and $|H_R^{[0]}) = (\tilde{h}_R)$. We can repeat these recursions until e.g. $\|H_{L/R}^{[n+1]} - H_{L/R}^{[n]}\|$ drops below some desired accuracy ϵ_S . This strategy is conceptually simple and closely resembles the successive construction of the environments in the context of (i)DMRG, but is not very efficient, as its performance is comparable to that of a power method.

A more efficient approach is to formally perform the geometric sums in (3.13) explicitly, and to iteratively solve the resulting two systems of equations

$$\begin{aligned} (H_L|[\mathbb{1} - T_L + |R)(\mathbb{1}|] &= (h_L| - (h_L|R)(\mathbb{1}| \\ [\mathbb{1} - T_R + |L)(L|]|H_R) &= (h_R) - (\mathbb{1})(L|h_R) \end{aligned} \quad (3.15)$$

for $(H_L|$ and $|H_R)$ to precision ϵ_S , as explained in Ref. [2].

So far, we have discussed the action of H_{A_C} . The action of H_C onto C follows simply from (3.11) by projecting onto A_L or A_R . Using the defining property of H_L or H_R ,

the result simplifies to

$$C' = \sum_{stkl} h_{kl}^{st} A_L^{s\dagger} A_L^k C A_R^\ell A_R^{t\dagger} + H_L C + C H_R. \quad (3.16)$$

The first two terms of (3.11) can be applied in $O(d^4 D^3)$ operations¹, and the last two terms in $O(dD^3)$ operations. For (3.16) the first term can be applied in $O(d^4 D^3)$ operations, and the last two terms in $O(D^3)$ operations. To generate the necessary terms for (3.11) and (3.16), we have to iteratively evaluate two infinite geometric sums, involving $O(D^3)$ operations (when iteratively solving (3.15) the solutions from the previous iteration can be used as starting vectors to speed up convergence).

3.5 Updating the state

In DMRG, we would update the state by replacing A_C with the lowest eigenvector \tilde{A}_C of H_{A_C} and then shift the center site to the right by computing an orthogonal factorization $\tilde{A}_C^s = \tilde{A}_L^s \tilde{C}_R$, or to the left by computing $\tilde{A}_C^s = \tilde{C}_L \tilde{A}_R^s$. As such, the state gets updated by only replacing the current site with \tilde{A}_L^s or \tilde{A}_R^s , while leaving all other sites untouched. However, applying this scheme in our setting would immediately destroy translation invariance after a single step.

We want to construct an alternative scheme that applies global updates in order to preserve translation invariance at any time. Global updates can most easily be applied with an explicit uniform parameterization in terms of a single tensor A . On the other hand, DMRG experience teaches us that the stability is greatly enhanced when applying updates at the level of A_C and C , which are isometrically related to the full state.

We therefore calculate the lowest eigenvector \tilde{A}_C of H_{A_C} like in DMRG, but additionally also the lowest eigenvector \tilde{C} of H_C . We then globally update the state by finding new \tilde{A}_L and \tilde{A}_R as the left and right isometric tensors that minimize $\sum_s \|\tilde{A}_L^s \tilde{C} - \tilde{A}_C^s\|^2$ and $\sum_s \|\tilde{C} \tilde{A}_R^s - \tilde{A}_C^s\|^2$ respectively. These minimization problems can be solved directly (not iteratively) and without inverting \tilde{C} (see below). As

¹In many cases, $h_{j,j+1}$ is sparse and the number d_h of nonzero elements is usually of the order $O(d^2)$. The first two terms can then be applied in $O(d_h D^3)$ operations.

shown in Ref. [2], at the variational optimum the values of these objective functions go to zero, and current A_C and C are the lowest eigenvectors of H_{A_C} and H_C respectively.

For the remainder of this section, we omit tildes and use the following matricizations of the 3-index tensors

$$\begin{aligned}\mathcal{A}_{L,(s\alpha,\beta)} &= A_{L,(\alpha,\beta)}^s \\ \mathcal{A}_{R,(\alpha,s\beta)} &= A_{R,(\alpha,\beta)}^s \\ \mathcal{A}_{C,(s\alpha,\beta)}^{[\ell]} &= \mathcal{A}_{C,(\alpha,s\beta)}^{[r]} = A_{C,(\alpha,\beta)}^s.\end{aligned}\tag{3.17}$$

We thus want to extract updated A_L and A_R from updated A_C and C by solving

$$\epsilon_L = \min_{\mathcal{A}_L^\dagger \mathcal{A}_L = \mathbb{1}} \|\mathcal{A}_C^{[\ell]} - \mathcal{A}_L C\|_2 \tag{3.18a}$$

$$\epsilon_R = \min_{\mathcal{A}_R \mathcal{A}_R^\dagger = \mathbb{1}} \|\mathcal{A}_C^{[r]} - C \mathcal{A}_R\|_2. \tag{3.18b}$$

In exact arithmetic, the solution of these minimization problems is known, namely \mathcal{A}_L will be the isometry in the polar decomposition of $\mathcal{A}_C^{[\ell]} C^\dagger$ (and similar for \mathcal{A}_R , see Thm. IX.7.2 in Ref. [52]). Computing the singular value decompositions (SVD)

$$\mathcal{A}_C^{[\ell]} C^\dagger = U^{[\ell]} \Sigma^{[\ell]} V^{[\ell]\dagger} \quad C^\dagger \mathcal{A}_C^{[r]} = U^{[r]} \Sigma^{[r]} V^{[r]\dagger}, \tag{3.19}$$

we thus obtain

$$\mathcal{A}_L = U^{[\ell]} V^{[\ell]\dagger} \quad \mathcal{A}_R = U^{[r]} V^{[r]\dagger}. \tag{3.20}$$

Notice that close to (or at) an exact solution $A_C^s = A_L^s C = C A_R^s$, the singular values contained in $\Sigma^{[\ell/r]}$ are the square of the singular values of C , and might well fall below machine precision. Consequently, in finite precision arithmetic, corresponding singular vectors will not be accurately computed.

An alternative that has proven to be robust and still close to optimal is given by directly using the following left and right polar decompositions

$$\mathcal{A}_C^{[\ell]} = U_{A_C}^{[\ell]} P_{A_C}^{[\ell]} \quad C = U_C^{[\ell]} P_C^{[\ell]} \tag{3.21a}$$

$$\mathcal{A}_C^{[r]} = P_{A_C}^{[r]} U_{A_C}^{[r]} \quad C = P_C^{[r]} U_C^{[r]} \tag{3.21b}$$

to obtain

$$\mathcal{A}_L = U_{A_C}^{[\ell]} U_C^{[\ell]\dagger} \quad \mathcal{A}_R = U_C^{[r]\dagger} U_{A_C}^{[r]}, \tag{3.22}$$

where matrices P are hermitian and positive. Alternative isometric decompositions might be considered in Eq. (3.21), though it is important that they are unique (e.g. QR with positive diagonal in R) in order to have $P_{A_C}^{[\ell/r]} \approx P_C^{[\ell/r]}$ close to convergence.

3.6 The Algorithm: VUMPS

We are now ready to formulate our *variational uniform MPS* (VUMPS) algorithm. As shown in Ref. [2], a variational minimum (vanishing energy gradient) in the manifold of uMPS is characterized by tensors A_L , C and A_R satisfying the conditions

$$\mathbf{H}_{A_C} \mathbf{A}_C = E_{A_C} \mathbf{A}_C \quad (3.23a)$$

$$\mathbf{H}_C \mathbf{C} = E_C \mathbf{C} \quad (3.23b)$$

$$A_C^s = A_L^s C = C A_R^s. \quad (3.23c)$$

Here, bold symbols denote vectorizations of the MPS tensors and matricizations of the effective Hamiltonians, and E_{A_C} and E_C are the lowest eigenvalues of the effective Hamiltonians.²

When iterating the steps outlined in the previous sections, convergence is obtained when these conditions are satisfied. In particular, starting with a properly orthogonalized initial trial state $|\Psi(A)\rangle$ of some bond dimension D , we begin by solving the two eigenvalue problems for the effective Hamiltonians H_{A_C} and H_C . Since we are still far from the fixed point, the resulting lowest energy states \tilde{A}_C and \tilde{C} will in general not satisfy the gauge condition (3.23c) together with current $A_{L/R}$.

Following the procedure of the previous section, we can, however, find optimal approximations \tilde{A}_L^s and \tilde{A}_R^s for (3.23c) to arrive at an updated uMPS. Conversely, \tilde{A}_C and \tilde{C} will not be the correct lowest energy eigenstates of the new effective Hamiltonians $H_{\tilde{A}_C}$ and $H_{\tilde{C}}$ generated from $\tilde{A}_{L/R}$. We then use the updated state and reiterate this process of alternately solving the effective eigenvalue problems, and finding optimal approximations for A_L and A_R to update the state.

We now elaborate on the various steps in the VUMPS algorithm. Firstly, extracting new $\tilde{A}_{L/R}$ from updated \tilde{A}_C and \tilde{C} can be done using the theoretically optimal (but numerically often inaccurate) Eq. (3.20) or the more robust Eq. (3.22), depending on the magnitude of the smallest singular value in \tilde{C} . As a good uMPS approximation will always involve small singular values, Eq. (3.22) is preferable most of the time, except maybe during the first few iterations.

²These values can be different and depend on the subtraction scheme for the divergent energy expectation value. If $h \rightarrow \tilde{h}$ is performed everywhere, we have $E_{A_C} = E_C = 0$. If, in the case of nearest neighbor interactions, we only substitute $h \rightarrow \tilde{h}$ in the construction of H_L and H_R , but not in the local terms, we will have $E_{A_C} = 2E_C = 2e$.

The maximum of the error quantities (3.18)

$$\epsilon_{\text{prec}} = \max(\epsilon_L, \epsilon_R) \quad (3.24)$$

provides an error measure for the fixed point condition in Eq. (3.23c), and is used as a global convergence criterion. It measures the precision of the current uMPS ground state approximation. Within every iteration, we use iterative methods (e.g. some variation of Lanczos) to find the eigenvectors \tilde{A}_C and \tilde{C} of the Hermitian operators H_{A_C} and H_C . As the goal is to drive the state towards the fixed point relations in Eqs. (3.23a) and (3.23b), it is not necessary to solve these eigenvalue problems to full machine precision. Rather, it is sufficient to use a tolerance ϵ_H chosen relative to ϵ_{prec} .³ A value of ϵ_H of the order of $\epsilon_{\text{prec}}/100$ has proven to work well in practice. It is also worthwhile to use tensors from the previous iteration as initial guess for the iterative solvers to speed up convergence.

As the main part of the algorithm works at fixed bond dimension (i.e. it is a single-site scheme in DMRG terminology), one might choose to increase the bond dimension D before starting a new iteration. A subspace expansion technique for increasing the bond dimension that works directly in the thermodynamic limit is explained in Ref. [2].

We refer readers to Ref. [2] for numerical results comparing this algorithm to iDMRG [5, 47] and iTEBD [49], which show impressive speedups of the new algorithm compared to those leading methods. However, we can already compare the theoretical properties of these algorithms. Neither iDMRG or iTEBD is truly solving the variational problem in the sense of directly trying to satisfy the fixed point conditions Eqs. (3.23). iDMRG closely resembles regular DMRG on a successively growing lattice, as it inserts and optimizes over new tensors in the center of the lattice in each step. Tensors from previous steps are not updated, as this would render the cost prohibitive. When this approach converges, the resulting fixed point tensors in the center can be assumed to specify the unit cell of an infinite MPS. VUMPS has the immediate advantage that i) it directly works in the thermodynamic limit at all iterations, and ii) it completely replaces the entire state after every iteration, thus moving faster through the variational manifold. In contrast, iDMRG keeps memory of earlier iterations, and cannot guarantee a monotonically decreasing energy that

³Further approximations to comparable accuracy can be made within the construction of the effective Hamiltonians, e.g. when determining H_L and H_R to precision ϵ_S . There, the approximations $\tilde{R} = CC^\dagger$ and $\tilde{L} = C^\dagger C$ for the true L and R needed for some of these operations are good enough, if ϵ_S is roughly of the same order of magnitude as ϵ_{prec} .

converges to an optimum associated with a translation invariant MPS in which the effects of the boundary have completely disappeared. The advantages of VUMPS come with a greater computational cost per iteration, as two eigenvalue problems (for A_C and for C) and – in the case of nearest neighbor interactions – two linear systems (for H_L and H_R) have to be solved. iDMRG only solves a single eigenvalue problem, and builds H_L and H_R step by step in every iteration. The latter approach is analogous to a power method for eigenvalue problems and, while very cheap, is expected to require many iteration steps to converge, especially for systems with large correlation lengths (e.g. close to criticality).

iTEBD [49] is based on evolving an initial state in imaginary time by using a Trotter-Suzuki decomposition of the evolution operator. Like VUMPS, iTEBD works in the thermodynamic limit at any intermediate step, typically with a unit cell that depends on how the Hamiltonian was split into local terms in order to apply the Trotter decomposition. Furthermore, as every application of the evolution operator increases the virtual dimension of the MPS, truncation steps are required to restore the original (or any suitable) value of the bond dimension. While VUMPS can take big steps through the variational space, time steps in iTEBD have to be chosen sufficiently small (especially in the final steps of the algorithm) to eliminate the Trotter error, which negatively affects the rate of convergence (Ref. [53], however, proposes a scheme to effectively obtain a larger time step). Furthermore, the Trotter splitting essentially limits the applicability of iTEBD to short-range interactions and dictates the size of the unit cell of the resulting MPS, e.g., in the most common case of nearest neighbor interactions a two-site unit cell is obtained. (The approach of Ref. [54] to obtain a translation invariant MPS is restricted to certain Hamiltonians, but see Ref. [55] for an alternative proposal that can in fact also deal with long range interactions).

We also note that for systems with time reversal symmetry, everything can be implemented in real arithmetic, and for systems with reflection symmetry, C and A_C^s can be chosen to be symmetric matrices and $A_R^s = A_L^{sT}$ (which implies that H_L and H_R are also related). In these cases, the computational cost is reduced. In the case of spontaneous symmetry breaking, MPS algorithms tend to converge to maximally symmetry broken states for which the entanglement is minimal. This is also the case for VUMPS. One can control which state the algorithm converges to by suitably biasing the initial state or by adding small perturbation terms to the Hamiltonian that explicitly break the symmetry, and which are switched off after a

few iterations.

Explicit conservation of translation symmetry was the very first requirement in the construction of VUMPS. In the case of spontaneous breaking of translation symmetry down to N -site translation symmetry (as e.g. in the case of a state with antiferromagnetic order), enforcing one-site translation symmetry would result in a (non-injective) equal weight superposition of all symmetry broken uMPS ground state approximations. In order to reach an optimal accuracy with a given bond dimension, such a superposition of N states is however undesirable, as the effective bond dimension is reduced to D/N . In the case where this situation cannot be amended by a simple unitary transformation that restores one-site translation symmetry (such as e.g. flipping every second spin in the case of an antiferromagnet), it is preferable to choose an MPS ansatz with a N -site unit cell, such that the state can spontaneously break translation symmetry. The generalization of the algorithm to multi-site unit cells is described in the next section.

3.7 Multi Site Unit Cell Implementations

We now generalize the VUMPS algorithm of the previous section for one-site translation invariant uMPS to the setting of translation invariance over N sites. Such a uMPS ansatz is then parameterized by N independent tensors $A(k)^s \in \mathbb{C}^{D \times d \times D}$, $k = 1, \dots, N$, which define the unit cell tensor

$$\mathbb{A}^{\mathbb{s}_n} = A(1)^{s_{nN+1}} \dots A(N)^{s_{nN+N}}, \quad (3.25)$$

where $\mathbb{s} = (s_1, \dots, s_N)$ is a combined index. We can then write the variational state as

$$|\Psi(\mathbb{A})\rangle = \sum_{\mathbf{s}} (\dots \mathbb{A}^{\mathbb{s}_{n-1}} \mathbb{A}^{\mathbb{s}_n} \mathbb{A}^{\mathbb{s}_{n+1}} \dots) |\mathbf{s}\rangle$$

and the left and right orthonormal forms are given by the relations

$$\begin{aligned} \sum_s A(k)_L^s \dagger A(k)_L^s &= \mathbb{1} \\ \sum_s A(k)_L^s R(k) A(k)_L^s \dagger &= R(k-1) \end{aligned} \quad (3.26a)$$

and

$$\begin{aligned} \sum_s A(k)_R^s A(k)_R^s \dagger &= \mathbb{1} \\ \sum_s A(k)_R^s \dagger L(k-1) A(k)_R^s &= L(k), \end{aligned} \quad (3.26b)$$

where it is understood that $N + 1 \equiv 1$ and $0 \equiv N$.

Defining the bond matrices $C(k)$ as the gauge transformation that relates left and right canonical form via $C(k-1)A(k)_R^s = A(k)_L^s C(k)$, we have $R(k) = C(k)C(k)^\dagger$ and $L(k) = C(k)^\dagger C(k)$. We can then also cast $|\Psi(\mathbb{A})\rangle$ in a mixed canonical form similar to (3.5a) with the center site tensor given by $A(k)_C^s = A(k)_L^s C(k) = C(k-1)A(k)_R^s$.

The variational minimum within this set of states is characterized by the following $3N$ fixed point relations

$$\mathbf{H}_{A(k)_C} \mathbf{A}(k)_C = E_{A(k)_C} \mathbf{A}(k)_C \quad (3.27a)$$

$$\mathbf{H}_{C(k)} \mathbf{C}(k) = E_{C(k)} \mathbf{C}(k) \quad (3.27b)$$

$$A(k)_C^s = A(k)_L^s C(k) = C(k-1)A(k)_R^s. \quad (3.27c)$$

Notice that due to (3.27c), the relations for different k are connected. There are several possible strategies for constructing algorithms which obtain states satisfying these conditions.

In the following, we present two approaches that have shown good performance and stable convergence, which we shall term the ‘‘sequential’’ and ‘‘parallel’’ methods. But let us first elaborate on computing effective Hamiltonians for multi-site unit cells, which works similarly in both methods. We again restrict to the case of nearest neighbor interactions, such that the effective Hamiltonians are constructed similar as in Sec. 3.4. To construct e.g. the left block Hamiltonian H_L , we first collect all local contributions from a single unit cell in h_L , before performing the geometric series of the transfer matrix, which now mediates a translation over an entire unit cell.

3.8 Sequential Algorithm

The sequential algorithm is inspired by finite size DMRG, in that we sweep through the unit cell, successively optimizing one tensor at a time while keeping tensors on other sites fixed. Notice that at site k , however, we need *two* updated bond matrices $\tilde{C}(k)_L = \tilde{C}(k-1)$ and $\tilde{C}(k)_R = \tilde{C}(k)$, in order to calculate updated $\tilde{A}(k)_{L/R}^s$ from $\tilde{A}(k)_C^s \approx \tilde{A}(k)_L^s \tilde{C}(k)_R \approx \tilde{C}(k)_L \tilde{A}(k)_R^s$. We thus have to amend steps of the single-site algorithm by constructing and solving for *two* effective Hamiltonians $H_{C(k-1)}$ and $H_{C(k)}$ instead of a single one. The newly optimized tensors then get replaced in *all* unit cells of the infinite lattice, and contributions to the effective Hamiltonians have to be recalculated accordingly, before moving on to the next site.

One could now try to argue, that e.g. in a left to right sweep it is enough at site k to calculate updated $\tilde{A}(k)_C$ and $\tilde{C}(k)_R = \tilde{C}(k)$ only, and to use $\tilde{C}(k-1)_R$ from the previous step at site $k-1$ as $\tilde{C}(k)_L$ for calculating $\tilde{A}(k)_R$. This approach fails, however, as the effective Hamiltonian used for calculating $\tilde{A}(k)_C$ already contains updated $\tilde{A}(k-1)_{L/R}$, while the effective Hamiltonian used for calculating $\tilde{C}(k-1)_R$ does not, and we cannot determine $\tilde{A}(k)_R$ from $\tilde{A}(k)_C$ and $\tilde{C}(k-1)_R$. Rather, $\tilde{C}(k)_L$ has to be recalculated using an *updated* effective Hamiltonian, which exactly leads to the sequential algorithm.

There is an additional subtlety that needs to be considered in order for all tensors to fulfill the gauge constraints (3.27c) to current precision. Bond matrices $\tilde{C}(k)$ are calculated as lowest energy eigenvectors of effective Hamiltonians $H_{C(k)}$ and are therefore only determined up to a phase. Consider $C(k)$ defined between sites k and $k+1$. At step k , it is updated as $\tilde{C}(k)_R$ and used to calculate $\tilde{A}(k)_L^s$. In the next step $k+1$, however, it is recalculated as $\tilde{C}(k+1)_L$ (with an updated effective Hamiltonian) and used to determine $\tilde{A}(k+1)_R^s$. At the fixed point we should then have $\tilde{C}(k)_R = \tilde{C}(k+1)_L = C(k)$, but this is only true if there is no phase ambiguity, which would also consequently lead to a phase mismatch between $\tilde{A}(k)_L$ and $\tilde{C}(k)$ after step $k+1$. This issue does not pose a problem for algorithm convergence (during calculations, matrices $C(k)$ always appear as products of the form $C(k)^\dagger C(k)$ or $C(k)C(k)^\dagger$ and mismatching phases thus cancel out), but can be easily circumvented by employing a phase convention when calculating updated $\tilde{C}(k)$.

3.9 Parallel Algorithm

In the parallel approach, we choose to update an entire unit cell at once, using effective Hamiltonians generated from the same current state. To that end, we first generate all terms necessary for all $H_{A(k)_C}$ and $H_{C(k)}$. For the case of nearest neighbor interactions, the contributions H_L and H_R to the left and right environment outside the unit cell can be shared, so that the corresponding geometric sum only needs to be computed once, and contributions inside the unit cell are obtained through successive applications of transfer matrices.

Next, we simultaneously and independently solve for the ground states $\tilde{A}(k)_C$ and $\tilde{C}(k)$ of all $2N$ effective Hamiltonians at once. Once these are obtained, we again simultaneously and independently determine all updated $\tilde{A}(k)_L$ and $\tilde{A}(k)_R$, concluding one iteration for updating the entire unit cell.

3.10 Comparison of the two approaches

Several comments on the two presented algorithms are in order. First, the parallel algorithm requires substantially less computational effort, since the construction of the different effective Hamiltonians $H_{A(k)_C}$ can recycle the calculation of the infinite geometric sum. Therefore, updating an entire unit cell only requires to evaluate two infinite geometric sums and $2N$ effective eigenvalue problems. In the sequential algorithm, updating the environment after every tensor update requires to reevaluate the geometric sum, thus leading to $2N$ infinite geometric sums and $3N$ effective eigenvalue problems for updating the complete unit cell. Additionally, the parallel approach offers the possibility of parallelizing the solution of all $2N$ eigenvalue problems in one iteration, while in the sequential approach only 3 eigenvalue problems can be solved in parallel for each site. However, while sweeping through the unit cell in the sequential approach, initial guesses for solving the infinite geometric sums can be generated easily from the previous iterations, and are usually much better than the initial guesses in the parallel algorithm. Equivalently, updated $\tilde{C}(k)$ obtained at site k is a very good initial guess for its recalculation with updated environment on site $k + 1$. Overall, the computational cost for the parallel update is still much cheaper, albeit less than expected.

On the other hand, state convergence in terms of iterations is generally substantially faster in the sequential approach. This seems reasonable, as the optimization on a current site takes into account all previous optimization steps, whereas in the parallel approach, the optimizations on different sites within one iteration are independent of each other. This effect gets amplified with increasing unit cell size N , and the performance of the parallel approach decreases, while the performance of the sequential approach seems more stable against increasing N .

In conclusion, while updating the entire unit cell is computationally cheaper in the parallel approach, the sequential algorithm usually requires a substantially smaller number of iterations due to faster convergence. While there are instances where one approach clearly outperforms the other by far, such cases are rare and strongly depend on initial conditions, and generally both approaches show comparable performance. For comparison benchmark results, see Ref. [2].

3.11 Conclusion and Outlook

We have introduced a novel algorithm for calculating MPS ground state approximations of strongly correlated one-dimensional quantum lattices models with nearest

neighbor or long range interactions, in the thermodynamic limit. Unlike conventional (i)DMRG, it variationally optimizes a uniform MPS, though just like conventional (i)DMRG, the optimization involves successive solutions of effective eigenvalue problems. The algorithm can easily be implemented by extending an existing single-site (i)DMRG implementation with routines for i) calculating effective Hamiltonian contributions from infinite environments and ii) solving an effective “zero site” eigenvalue problem in addition to the usual single-site problem. The new algorithm is free of any ill-conditioned inverses. Additionally, as it does not rely on imaginary time evolution, it is especially fit for studying systems with long range interactions.

In Ref. [2], the VUMPS algorithm is shown to perform remarkably better than leading algorithms, iDMRG and iTEBD, for finding ground states of 1D and quasi-1D quantum ground states. In Chapter 5, we also show that the VUMPS algorithm can be applied to finding fixed points of infinite-sized transfer matrices, and therefore can be applied to the problem of contracting infinite 2D tensor networks that show up naturally in tensor network calculations of 2D classical statistical mechanics problems as well as in iPEPS calculations of 2D quantum many-body problems.

REVISITING THE CORNER TRANSFER MATRIX RENORMALIZATION GROUP ALGORITHM FOR ASYMMETRIC LATTICES

¹M. T. Fishman, L. Vanderstraeten, V. Zauner-Stauber, J. Haegeman, and F. Verstraete, “Faster Methods for Contracting Infinite 2D Tensor Networks”, [arXiv:1711.05881](https://arxiv.org/abs/1711.05881).

Here we present a short review of the corner transfer matrix renormalization group (CTMRG) algorithm introduced by Nishino and Okunishi. We also propose a new variant of CTMRG for contracting asymmetric two-dimensional tensor networks that improves the numerical stability compared to the most commonly used method.

4.1 Introduction

Corner transfer matrices (CTMs) have an extensive history in many-body physics. The CTM method was originally proposed by Baxter [56–58] as an extension of the Kramers-Wannier approximation [59] to study two-dimensional (2D) classical statistical mechanics models. Baxter formulated a variational method using CTMs, and Baxter’s CTM method was subsequently used to both analytically and numerically study a variety of 2D statistical mechanics models [57, 58, 60–64].

Later on in a separate development, White proposed the famous density matrix renormalization group algorithm (DMRG) [5, 6] as a way to fix the numerical renormalization group (NRG) of Wilson [26]. It later became clear that the DMRG algorithm could be viewed as a variational method for optimizing a general class of states known as matrix product states (MPSs) [30, 32, 65]. With the correspondence between d -dimensional quantum lattice models and $(d + 1)$ -dimensional classical lattice models [66–69] in mind, Nishino formulated a version of DMRG for studying 2D classical statistical mechanics models by applying DMRG to find the fixed point of the transfer matrix [70]. Around the same time, Nishino and Okunishi made the explicit connection between White’s DMRG algorithm and Baxter’s CTM method and proposed the corner transfer matrix renormalization group (CTMRG) algorithm [71, 72] as an improved numerical CTM method.

Since its inception, Nishino and Okunishi's CTMRG method has been used in a wide variety of applications in the study of 2D classical systems, including extensions to studying 1D quantum systems, stochastic models and hyperbolic lattices [73–84]. In addition, CTMs and CTMRG turn out to be very useful for the study of 3D classical and 2D quantum many-body systems as well. Direct extensions of the CTM formalism to 3D have been proposed using corner tensors [56, 58, 82, 84, 85]. In addition, the 2D version of CTMRG is used extensively as a key part of other types of 3D classical and 2D quantum tensor network state algorithms. For example, CTMRG was used as part of the first tensor product state (TPS) and interaction-round-a-face (IRF)-type TPS calculations of 3D classical and 2D quantum many-body systems [86–92]. In those calculations, CTMRG is used as the method to approximate 2D tensor networks that naturally arise as part of the optimization of the TPS and IRF-type TPS.

Perhaps the most notable use of the CTMRG algorithm is in a modern iteration of TPS known as the infinite projected entangled pair state (iPEPS) [93–95] ansatz, which is used extensively in the study of 2D quantum many-body systems. In iPEPS calculations, a critical step of the algorithm is again the approximation of 2D tensor networks, and CTMRG has been and continues to be the most commonly used method for contracting those networks [82, 95–119]. In addition, the CTM ansatz plays an important role in more recent proposals for improved iPEPS optimization techniques [114, 120, 121]. We refer readers to Ref. [82, 84] for more general discussions of the role of CTMs in the study of classical and quantum many-body systems.

Baxter's CTM method and Nishino and Okunishi's CTMRG were originally formulated primarily for studying 2D classical many-body systems with local interactions known as interaction-round-a-face (IRF) models. Many of these models obey some set of lattice symmetries (for example for models on the square lattice, many are symmetric with respect to reflections about the horizontal and vertical directions). In this context, the CTM and CTMRG methods greatly simplify and the renormalization can generally be implemented with a Hermitian eigendecomposition of some product of CTMs. Baxter generalized his variational method to the asymmetric case, for example in Chapter 13 of Ref. [58]. In addition, Nishino and Okunishi discussed a generalization of their CTMRG method to the asymmetric case [71, 72]. Both of these methods relied in one way or another on full eigendecompositions of large asymmetric matrices, and we are not aware of them being used in their fully

asymmetric forms in practice.

Relying on the full diagonalization of large asymmetric matrices is not ideal. They are computationally expensive and of limited accuracy for large matrices and are difficult to constrain to being real for contracting real tensor networks. Finding numerically robust and accurate generalizations of CTMRG to the asymmetric case has proven to be less than straightforward. Most of the advancements in developing asymmetric CTMRG methods have been proposed in the context of iPEPS calculations [82, 95, 97–99, 108, 122]. There, it is important to have a CTMRG formulation that can contract general asymmetric tensor networks, which arise naturally when large unit cells are involved.

Many early proposals for use in iPEPS calculations involved to some extent a symmetric approximation of the network [95, 97–99], which may not be appropriate for “very asymmetric” networks. The most recent proposal described by Corboz et al. in Ref. [108] makes use of a procedure introduced by Huang [123–125] in the context of asymmetric transfer matrix DMRG (TM-DMRG or TMRG). This asymmetric CTMRG method has been applied very successfully in iPEPS calculations of infinite 2D quantum many-body systems [104, 107, 108, 114–118, 126], including the most modern iPEPS calculations of the 2D Hubbard model [119]. Additionally, it does not rely on any asymmetric eigendecompositions or symmetric approximations of the network.

In this work, we assess the asymmetric CTMRG method of Ref. [108] in detail and propose an improvement. We find that the method as proposed in Ref. [108] relies on a (potentially) ill-conditioned inverse which in practice, if care is not taken, can lead to numerical instabilities. Here, we propose a slight variation of that method which improves the conditioning of the inverse required in that method. Like the method of Ref. [108], the new method avoids symmetrizing the environment. The only matrix decomposition it relies on is the singular value decomposition (SVD), which is very reliable.

As a demonstration of our new approach, we use the example of the Ising model near criticality. To artificially break the natural reflection symmetries of the model, we apply a random non-unitary change of basis to the local spin degrees of freedom. We show examples of where the method of Corboz et al. from Ref. [108] may become numerically unstable (in practice limiting the precision), and show our new method is able to contract the partition function to the same precision as can be obtained by the fully symmetric version of CTMRG applied to the Ising model in

the original basis.

The chapter is organized as follows. In Section 4.2, we review the CTM formalism and describe general properties of the CTM ansatz. In Section 4.3.1, we give a general review of the asymmetric CTMRG algorithm and discuss different choices that can be made for the renormalization. In Section 4.3.2, we give a review of different methods for finding the CTMRG projector for the full asymmetric case, as well as give a detailed discussion of the method we will compare against from Ref. [108]. In Section 4.3.3, we describe our variation on the asymmetric CTMRG of Ref. [108]. In Section 4.3.4, we discuss possible alternatives to our new asymmetric CTMRG method. Finally, in Section 4.4, we present our results comparing our new method to the old method.

4.2 Corner transfer matrix (CTM) formalism

The corner transfer matrix (CTM) was introduced by Baxter as a variational ansatz for both solving for and approximating properties of 2D classical statistical mechanics models[56–58]. Transfer matrices have a long history in statistical mechanics, and were used as the basis for Onsager’s famous exact solution of the Ising model[58, 127]. Baxter generalized the ansatz of Kramers and Wannier [59] and recognized that the maximal eigenvectors of transfer matrices could be approximated by an ansatz comprised of products of (hopefully) small matrices [56–58], which we would recognize now as the matrix product state (MPS) [30, 32, 65], the basis for the remarkable DMRG algorithm of White [5, 6].

The row-to-row transfer matrix represents a row of lattice degrees of freedom, while the column-to-column transfer matrix represents a column of lattice degrees of freedom. Baxter recognized the usefulness of the so-called corner transfer matrices (CTMs), which represent entire quadrants of lattice degrees of freedom. Of particular interest in this work is the fact that CTMs turn out to be very well-suited for framing numerical renormalization algorithms [56–58, 71, 72].

A convenient language for describing the partition function of a statistical mechanics

model with local interactions is in terms of a tensor network ¹, for example:

$$Z_{MN} = \text{Tr} \left[\begin{array}{cccc} \vdots & \vdots & \vdots & \vdots \\ \cdots & \boxed{T} & \boxed{T} & \boxed{T} & \cdots \\ \cdots & \boxed{T} & \boxed{T} & \boxed{T} & \cdots \\ \cdots & \boxed{T} & \boxed{T} & \boxed{T} & \cdots \\ \cdots & \boxed{T} & \boxed{T} & \boxed{T} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{array} \right] \quad (4.1)$$

In Eq. (4.1), the fourth-order tensors T are either the Boltzmann weights or related to the Boltzmann weights by a local tensor renormalization. $\text{Tr}[\dots]$ denotes two traces, one over the open horizontal indices and another over the open vertical indices (for periodic boundary conditions). M, N denote the number of rows and columns, which we will take to infinity since we will work directly in the thermodynamic limit. See Ref. [128] for a simple construction of T for models with nearest neighbor interactions. Generalizations to models with other types of interactions, like interaction-round-a-face (IRF) models, are straightforward. Following Baxter [58], we define the ‘‘partition function per site’’ as

$$\Omega \equiv \lim_{M, N \rightarrow \infty} Z_{MN}^{1/(MN)}. \quad (4.2)$$

Tensor networks like the one shown in Eq. (4.1) also arise naturally in the context of TPS or iPEPS representations of 2D quantum body systems, where they are used to calculate properties of the state such as the norm [94].

The CTM ansatz of Baxter is a way to approximate Eq. (4.1) as well as other observables of the system with a local set of (hopefully) small tensors. The general ansatz used for the environment in the corner transfer matrix renormalization group (CTMRG) algorithm is as follows:

$$\begin{array}{cccc} \boxed{C_{LU}} & \boxed{A_U} & \boxed{A_U} & \boxed{C_{UR}} \\ | & | & | & | \\ \boxed{A_L} & \boxed{T} & \boxed{T} & \boxed{A_R} \\ | & | & | & | \\ \boxed{A_L} & \boxed{T} & \boxed{T} & \boxed{A_R} \\ | & | & | & | \\ \boxed{C_{DL}} & \boxed{A_D} & \boxed{A_D} & \boxed{C_{RD}} \end{array} . \quad (4.3)$$

¹For readers unfamiliar with tensor networks, we refer them to Ref. [105] for an introduction.

The matrices $\{C_{ab}\}$ (where $ab \in \{LU, UR, RD, DL\}$) in Eq. (4.3) are known as the ‘‘corner transfer matrices’’ (CTMs)[56–58, 71, 72], representing approximations of the infinite quadrants of the tensor network in Eq. (4.1). In our notation, the CTM C_{LU} represents an approximation for the upper-left quadrant of the tensor network, etc. The third-order tensors $\{A_a\}$ (where $a \in \{L, U, R, D\}$) in Eq. (4.3) represent approximations of the half infinite rows or columns of the tensor network. A_L and A_R are the so-called left and right ‘‘half row transfer matrices’’ (HRTMs), while A_U and A_D are the up and down ‘‘half column transfer matrices’’ (HCTMs). In the tensor network language, the CTMs, HRTMs, and HCTMs make up the environment tensors.

With this ansatz, the partition function per site can be approximated as follows:

$$\Omega \approx \frac{\begin{array}{c} \begin{array}{ccccc} \textcircled{C_{LU}} & \textcircled{A_U} & \textcircled{C_{UR}} & & \\ | & | & | & & \\ \textcircled{A_L} & \textcircled{T} & \textcircled{A_R} & & \\ | & | & | & & \\ \textcircled{C_{DL}} & \textcircled{A_D} & \textcircled{C_{RD}} & & \end{array} & \begin{array}{cc} \textcircled{C_{LU}} & \textcircled{C_{UR}} \\ | & | \\ \textcircled{C_{DL}} & \textcircled{C_{RD}} \end{array} \end{array}}{\begin{array}{c} \begin{array}{cc} \textcircled{C_{LU}} & \textcircled{C_{UR}} \\ | & | \\ \textcircled{A_L} & \textcircled{A_R} \\ | & | \\ \textcircled{C_{DL}} & \textcircled{C_{RD}} \end{array} & \begin{array}{ccccc} \textcircled{C_{LU}} & \textcircled{A_U} & \textcircled{C_{UR}} & & \\ | & | & | & & \\ \textcircled{C_{DL}} & \textcircled{A_D} & \textcircled{C_{RD}} & & \end{array} \end{array}}. \quad (4.4)$$

Baxter showed that finding the tensors $\{C_{ab}\}$ and $\{A_a\}$ that maximize Ω can be framed as a variational problem (see, for example, Chapter 13 of Ref. [58] for an in-depth discussion).

Additionally, local observables can be calculated in terms of the CTM ansatz as

follows:

$$\langle X \rangle \approx \frac{
 \begin{array}{c}
 \textcircled{C_{LU}} - \textcircled{A_U} - \textcircled{C_{UR}} \\
 | \quad | \quad | \\
 \textcircled{A_L} - \textcircled{T_X} - \textcircled{A_R} \\
 | \quad | \quad | \\
 \textcircled{C_{DL}} - \textcircled{A_D} - \textcircled{C_{RD}}
 \end{array}
 }{
 \begin{array}{c}
 \textcircled{C_{LU}} - \textcircled{A_U} - \textcircled{C_{UR}} \\
 | \quad | \quad | \\
 \textcircled{A_L} - \textcircled{T} - \textcircled{A_R} \\
 | \quad | \quad | \\
 \textcircled{C_{DL}} - \textcircled{A_D} - \textcircled{C_{RD}}
 \end{array}
 } . \tag{4.5}$$

Correlation functions are also straightforward to calculate using the CTM ansatz [71, 82]. See Ref. [115, 120, 121] for modern approaches to calculating arbitrary correlation functions in terms of the CTM ansatz.

If the tensor T is Hermitian about the horizontal and vertical direction ($T_{lurd} = \bar{T}_{ruld} = \bar{T}_{ldru}$), the CTM ansatz can be simplified to the following:

$$\begin{array}{c}
 \textcircled{C} - \textcircled{A} - \textcircled{A} - \textcircled{\bar{C}} \\
 | \quad | \quad | \\
 \textcircled{B} - \textcircled{T} - \textcircled{T} - \textcircled{B} \\
 | \quad | \quad | \\
 \textcircled{B} - \textcircled{T} - \textcircled{T} - \textcircled{B} \\
 | \quad | \quad | \\
 \textcircled{\bar{C}} - \textcircled{A} - \textcircled{A} - \textcircled{C}
 \end{array} . \tag{4.6}$$

where we have defined $A \equiv A_U = A_D$, $B \equiv A_L = A_R$, $C \equiv C_{LU} = C_{UR}^\dagger = C_{RD} = C_{DL}^\dagger$ (where we use the convention that the indices of the tensors in the diagram are ordered counterclockwise, and a complex conjugate that flips the index ordering). Additionally, $A^s = (A^s)^\dagger$ and $B^s = (B^s)^\dagger$ ². We will be focused on the full asymmetric ansatz (4.3), but the symmetric limit is good to have in mind for gaining intuition about the new method that we will discuss.

We will define the matrices $\{\rho_L, \rho_U, \rho_R, \rho_D\}$ from cyclic permutations of products of the four CTMs, i.e. $\rho_L \equiv C_{LU}C_{UR}C_{RD}C_{DL}$, $\rho_U \equiv C_{UR}C_{RD}C_{DL}C_{LU}$, etc. or

²Throughout this work, we use the notation that A^s is the matrix associated with fixing the index of A (an HRTM/HCTM) connected to the network tensor T to be s .

diagrammatically:

$$\rho_L \equiv \begin{array}{cc} C_{LU} & C_{UR} \\ | & | \\ C_{DL} & C_{RD} \end{array} \quad (4.7)$$

$$\rho_U \equiv \begin{array}{cc} C_{LU} & C_{UR} \\ | & | \\ C_{DL} & C_{RD} \end{array} \quad (4.8)$$

etc. We will discuss the significance of these matrices, which were discussed by Baxter [58], later on in the context of the CTMRG algorithm. Following Nishino and Okunishi (and possibly in slight abuse of terminology), we will refer to them as the “density matrices” [71, 72]. These are only the smallest density matrices, and we will discuss other density matrices later.

Some interesting properties of CTMs are worth pointing out. As Baxter discussed in Chapter 13 of Ref. [58], there exists a local basis change of the CTMs, HRTMs, and HCTMs (which following the tensor network literature we will call a “gauge” degree of freedom) in which the CTMs are simultaneously diagonal. This diagonal gauge can be constructed by diagonalizing the density matrices (which all have equal sets of eigenvalues) and transforming the CTMs (and HRTMs and HCTMs) appropriately by the eigenbases of the density matrices. In the case where the density matrices are Hermitian, for example when the network is Hermitian about reflections about the horizontal and vertical directions and the ansatz in Eq. (4.6) is used, then a Hermitian eigendecomposition can be used and the gauge transformations are unitary.

For example, if the ansatz in Eq. (4.6) is used, $\rho_L = \rho_R = (CC^\dagger)^2$ and $\rho_U = \rho_D = (C^\dagger C)^2$. The density matrices can therefore be diagonalized with Hermitian eigendecompositions:

$$\rho_{L/R} = U\Lambda U^\dagger \quad (4.9)$$

$$\rho_{U/D} = V\Lambda V^\dagger. \quad (4.10)$$

With the proper phases chosen for the eigenvectors, the CTMs can therefore be diagonalized with the unitary matrices U and V as follows:

$$\tilde{C} = U^\dagger C V = \Lambda^{1/4}. \quad (4.11)$$

Alternatively, U , V , and $\Lambda^{1/4}$ can be obtained directly from the SVD of C . The HCTMs must also be transformed as $\tilde{A}^s = V^\dagger A^s V$ and the HRTMs are transformed as $\tilde{B}^s = U^\dagger B^s U$.

Note that if the density matrices are not Hermitian, a non-Hermitian eigendecomposition must be used. In that case, there is a freedom in the normalization of the eigenvectors, which changes the entries of the diagonal CTMs [58].

In the next section, we review the CTMRG algorithm of Nishino and Okunishi [71, 72], which was introduced as an improvement to Baxter’s CTM method [56–58] and is the method most commonly used for numerically solving for the CTMs, HRTMs and HCTMs. Additionally, we discuss our new variant of the algorithm, and discuss possible further alternatives.

4.3 The corner transfer matrix renormalization group (CTMRG) algorithm

In this section, we review current approaches to the asymmetric CTMRG algorithm and discuss our new contribution. In Section 4.3.1, we introduce notation that is useful for discussing the CTMRG algorithm, and discuss in general terms how the algorithm works. In Section 4.3.2, we review current methods for a key step of the CTMRG algorithm, which is obtaining the projector, and discuss limitations of the current approaches. In Section 4.3.3, we discuss our new proposal for obtaining the projector, and in Section 4.3.4, give some context for the new proposal and discuss possible alternatives.

4.3.1 General overview of the CTMRG algorithm

Baxter introduced a variational method for solving numerically for the CTMs [56–58] which was used to study a variety of statistical mechanics models. Using insight from the DMRG algorithm of White [5, 6], Nishino and Okunishi improved the method to create their CTMRG algorithm.

The CTMRG algorithm fundamentally involves iteratively contracting row-to-row transfer matrices and column-to-column transfer matrices into the environment CTMs, HRTMs and HCTMs. This enlarges the size of the CTMs, HRTMs and HCTMs, and without some form of truncation, they would grow exponentially in size with the number of transfer matrices absorbed.

To make the algorithm easier to discuss, we will introduce some notation. We start by defining the following set of enlarged CTMs:

$$\begin{array}{ccccccc}
 \begin{array}{|c|} \hline C_{LU}^{(1,0)} \\ \hline \end{array} & \begin{array}{|c|} \hline C_{UR}^{(0,1)} \\ \hline \end{array} & & \begin{array}{|c|} \hline C_{LU} \\ \hline \end{array} \begin{array}{|c|} \hline A_U \\ \hline \end{array} & \begin{array}{|c|} \hline A_U \\ \hline \end{array} \begin{array}{|c|} \hline C_{UR} \\ \hline \end{array} \\
 & & \equiv & & \\
 \begin{array}{|c|} \hline C_{DL}^{(0,1)} \\ \hline \end{array} & \begin{array}{|c|} \hline C_{RD}^{(1,0)} \\ \hline \end{array} & & \begin{array}{|c|} \hline C_{DL} \\ \hline \end{array} \begin{array}{|c|} \hline A_D \\ \hline \end{array} & \begin{array}{|c|} \hline A_D \\ \hline \end{array} \begin{array}{|c|} \hline C_{RD} \\ \hline \end{array} .
 \end{array} \tag{4.12}$$

These types of CTMs appear when column-to-column transfer matrices are absorbed into the environment. Alternatively, if row-to-row transfer matrices are absorbed into the current environment, we would obtain the following enlarged CTMs:

$$(4.13)$$

The notation $C_{LU}^{(i,j)}$ can be understood as being a CTM for the upper left corner, with i columns of spins added and j rows of spins added. Note that in this notation, $\{C_{ab}^{(0,0)} \equiv C_{ab}\}$. For example, another set of enlarged CTMs $\{C_{ab}^{(1,1)}\}$ are:

$$(4.14)$$

In addition, we define the enlarged HRTMs and HCTMs $A_a^{(1)}$:

$$(4.15)$$

where again we define $A_a^{(0)} \equiv A_a$. We also define the enlarged half system transfer matrices $C_U^{(1,1)} = C_{LU}^{(1,1)} C_{UR}^{(1,1)}$, $C_R^{(1,1)} = C_{UR}^{(1,1)} C_{RD}^{(1,1)}$, $C_D^{(1,1)} = C_{RD}^{(1,1)} C_{DL}^{(1,1)}$, and $C_L^{(1,1)} =$

$C_{DL}^{(1,1)}C_{DL}^{(1,1)}$, for example:

$$(4.16)$$

Finally, we define the enlarged density matrices $\{\rho_a^{(1)}\}$ where $\rho_L^{(1)} \equiv C_U^{(1,1)}C_D^{(1,1)}$, $\rho_U^{(1)} \equiv C_R^{(1,1)}C_L^{(1,1)}$, etc., or diagrammatically:

$$(4.17)$$

In our notation, $\rho_a^{(0)} \equiv \rho_a$ defined in the previous section.

These are the CTMs, HCTMs and HRTMs most commonly used in CTMRG algorithms. Larger CTMs, HCTMs, HRTMs, and density matrices may give more accurate results, but the computational cost grows exponentially as sites are added to them. Also note that the enlarged half system transfer matrices $C_a^{(1,1)}$ and enlarged density matrices $\rho_a^{(1)}$ could be formed from larger or smaller CTMs, for example an alternative way to form $C_U^{(1,1)}$ could be from $C_{LU}^{(1,0)}C_{UR}^{(0,1)}$, $C_{LU}^{(1,2)}C_{UR}^{(2,1)}$, etc. In addition, there are even more ways of obtaining $\rho_a^{(1,1)}$, for example $\rho_L^{(1,1)} = C_{LU}^{(1,i)}C_{UR}^{(i,j)}C_{RD}^{(j,k)}C_{DL}^{(k,1)}$. In fact, the fixed point $\tilde{C}_U^{(1,1)}$ obtained from $C_{LU}^{(1,i)}C_{UR}^{(i,1)}$ for $i \rightarrow \infty$ is very related to the two site wavefunction used in DMRG, if the network is Hermitian. For the purpose of CTMRG, however, larger CTMs, HCTMs and HRTMs are not used in practice because of the larger computational costs. Also note that after the network is optimized, all ways of forming $C_a^{(1,1)}$ and $\rho_a^{(1)}$ should be approximately equal.

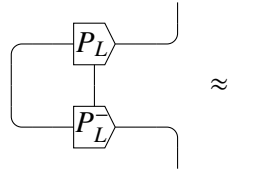
Note that the enlarged density matrices $\{\rho_a^{(1)}\}$ we define here are analogous to the density matrix used by White in the two site version of the DMRG algorithm [5, 6], which relies on truncating the density matrix according to its eigenvalues. This correspondence is most clear when the density matrix is Hermitian, and was one of

the key insights that led Nishino and Okunishi [70–72] to develop the corner transfer matrix renormalization group (CTMRG) algorithm.

The goal of CTMRG is to expand the CTMs, HCTMs and HRTMs and, using these enlarged tensors, obtain a projector to renormalize them back down to CTMs, HCTMs and HRTMs of the original size. There are four sets of projectors for the asymmetric CTMRG algorithm, one for each direction. We will call the projectors $P_L P_L^-$, $P_U P_U^-$, $P_R P_R^-$, and $P_D P_D^-$, for the projectors that renormalize the left, up, right and down boundaries, respectively. In the diagrammatic notation, the left projector is:


(4.18)

where P_L^- is the approximate left inverse of P_L , in other words they satisfy $[P_L^-]^s P_L^s \approx I$ or diagrammatically:


(4.19)

Roughly speaking, we want to find the projector $P_L P_L^-$ such that $Tr [P_L P_L^- \rho_L^{(1)}] \approx Tr [\rho_L^{(1)}]$. The topic of how to obtain these projectors will be addressed in the next sections.

In general, there are different procedures for the ordering of obtaining the projectors and using them to renormalize the boundaries. We summarize the most commonly used methods here:

1. **Unidirectional method.** A commonly used method in iPEPS is the unidirectional approach, where only a single direction is renormalized at a time. A single step of CTMRG consists for four separate directional moves, known as the left, up, right and down moves. For example, the left move involves obtaining the left projector $P_L P_L^-$ from the enlarged environment tensors, and then renormalizing the enlarged CTMs using the tensor P_L and P_L^- as follows:

$$\begin{aligned} C'_{LU} &= P_L^- C_{LU}^{(1,0)} \\ A'_L &= P_L^- A_L P_L \\ C'_{DL} &= C_{DL}^{(0,1)} P_L \end{aligned}$$

Then, the environment tensors are updated with these new tensors C'_{LU} , A'_L , and C'_{DL} . Next, the up move is performed by obtaining the up projector $P_U P_U^-$ and the using it to renormalize $C_{LU}^{(0,1)'}$, $A_U^{(1)}$, and $C_{(UR)}^{(1,0)}$, followed by the right move and down move. This process is repeated until convergence, which constitutes the unidirectional CTMRG algorithm. This was the first CTMRG method used in iPEPS calculation [95], and is the most commonly used in recent iPEPS studies.

2. **Bidirectional method.** The bidirectional approach is very similar to the unidirectional approach, except essentially the left and right moves are performed at once in a combined left-right move, and the up and down moves are performed at once in a combined up-down move. The up-down moves and left-right moves are alternated until convergence. The bidirectional approach was introduced for use in iPEPS calculations in Ref. [98].
3. **Quad-directional method.** The quad-directional approach was the one introduced in the original CTMRG proposal by Nishino and Okunishi [71, 72]. In this approach, with the current environment, all four directional projectors $P_L P_L^-$, $P_U P_U^-$, $P_R P_R^-$, and $P_D P_D^-$ are obtained at once. Then, all four enlarged CTMs $\{C_{ab}^{(1,1)}\}$ and HCTMs/HRTMs are renormalized at once:

$$\begin{aligned}
 A'_L &= P_L^- A_L^{(1)} P_L \\
 C'_{LU} &= P_L^- C_{LU}^{(1,1)} P_U \\
 A'_U &= P_U^- A_U^{(1)} P_U \\
 C'_{UR} &= P_U^- C_{UR}^{(1,1)} P_R \\
 &\dots
 \end{aligned}$$

The advantage of this approach is that it more explicitly preserves symmetries of the CTMs, if they are present.

The unidirectional approach is currently the most commonly used in iPEPS calculations, since it seems to be best suited for extending to contracting tensor networks with multi-site unit cells. See discussions in Ref. [99, 108] for generalizing the CTMRG algorithm to arbitrary multi-site unit cells using the unidirectional approach. Also see Ref. [115] for a discussion of calculating infinite sums of local operators using the unidirectional CTMRG method. We also refer readers to Ref. [82]

for an overview of different renormalization approaches to applying CTMRG to quantum systems ³.

We have not yet addressed how to obtain the projectors, which is the most important part of the algorithm. In the next section, we review the currently available methods for obtaining the projectors.

4.3.2 Obtaining the CTMRG projector: review of current methods

Here we will discuss currently available approaches to obtaining the projector in asymmetric CTMRG. For simplicity, in this section we will focus on obtaining the left projector for the left move of the unidirectional approach described in the previous section, but all discussions extend to the other directional approaches.

A sensible choice for obtaining the projector, as pointed out by both Baxter [58] and Nishino and Okunishi [71, 72], is to use the eigenvectors of $\rho_L^{(1)}$ associated with the largest eigenvalues. In other words, we perform the eigendecomposition $\rho_L^{(1)} \approx X \Lambda X^{-1}$ where we truncate according to the eigenvalues λ_i (where $\Lambda_{ij} = \lambda_i \delta_{ij}$). This works well for obtaining the projector for any direction in which the density matrix is Hermitian, in which case a Hermitian eigendecomposition is used. Then, X is an isometry composed of the orthonormal eigenvectors associated with the largest eigenvalues, so $X^\dagger X = I$ and the projector for the left boundary is XX^\dagger . This is the approach used in the original CTMRG proposal of Nishino and Okunishi [71, 72] and is extremely fast, robust and accurate.

If the density matrix is asymmetric, however, relying on a large asymmetric eigendecomposition is not ideal, and we are not aware of them being used in the literature for any fully asymmetric CTMRG calculations. There are a few drawbacks to using an eigendecomposition in the asymmetric case. As Baxter pointed out in Chapter 13 of Ref. [58], there is an arbitrary gauge degree of freedom associated with the normalization of the eigenvectors that may need to be fixed properly ⁴. It is also difficult to constrain the renormalization to be real when the network is real, which leads to a higher computational cost. Full eigendecompositions of large asymmetric matrices

³Note that we use slightly different terminology from Ref. [82], since that reference is focused on CTMRG when the network is Hermitian about a certain direction. In that reference, their one-directional method is what we refer to as the bidirectional method, and their two-directional method is what we refer to as the quad-directional method.

⁴In practice, it is a diagonal gauge degree of freedom that changes the eigenvalues of the corner matrices, so a choice may be to choose the gauge/normalization of the eigenvectors such that the eigenvalues of the corner matrices are all equal, which is related to a discussion on gauge fixing two biorthogonal MPSs in Ref. [124].

are not very accurate compared to other matrix decompositions more commonly used in tensor network calculations, such as the SVD or the QR decomposition.

In our own tests of a naive implementation of calculating the projectors using the eigenvectors of the asymmetric density matrix $\rho_L^{(1)}$, the method worked well for small bond dimensions far from criticality, but became numerically unstable closer to criticality and for large bond dimensions. Using non-Hermitian eigendecompositions may work fine for networks with more limited symmetry constraints than being Hermitian about horizontal and vertical reflections (for example, for complex symmetric networks, networks rotationally invariant by π [83], or networks that are Hermitian in one direction but not the other, such as for 1D quantum systems [82]), but in general in CTMRG it appears that asymmetric eigendecompositions have been avoided.

So, what are the alternatives? In practice, many methods have been proposed in the context of iPEPS [82, 95, 97–99, 108]. Early proposals generally involved symmetrizing the density matrix in one way or another, which may be appropriate for nearly symmetric networks. For example, Ref. [95] proposed a Hermitian eigendecomposition of the sum of squares of grown corners, and was used successfully as part of iPEPS calculations, for example Ref. [97, 106]. In Ref. [98, 99], it was proposed to take the SVD of $\rho_L^{(1)}$, truncate according to the singular values, and form an isometry from either the left or right singular vectors associated with the largest singular values. This approach was used successfully in a variety of iPEPS studies [98–101, 106, 109, 110]. Finally, we should note that some networks like those for stochastic models, where the fixed point in one direction is exactly a product state but in the other direction is nontrivial, appear to require specially designed CTMRG approaches [74].

Many of these discussions about the proper procedure for calculating the projector in CTMRG when the density matrix is non-Hermitian mirror the developments of non-Hermitian transfer-matrix DMRG (TM-DMRG or TMRG). TMRG was originated by Nishino [70], who was the first to apply the DMRG algorithm directly to the transfer matrix of a statistical mechanics model. In the original proposal, Nishino only addressed the Hermitian version of the algorithm. For the non-Hermitian case, there have been many proposals for the proper choice of the density matrix, and like in CTMRG the choice may depend on the model in question [129–134].

In fact, the latest proposal by Corboz et al. first used in iPEPS calculation in Ref. [104] and described in Ref. [108] for obtaining the CTMRG projector is adapted

from a proposal that was introduced by Huang [123–125] as a new approach to non-Hermitian TMRG. This has been used in a majority of the most modern iPEPS calculations [104, 107, 108, 114–119, 126].

We will summarize the method described in Ref. [108] here, using the notation we have introduced. In order to form the projector for the left move, $P_L P_L^-$, Corboz et al. proposed using $C_D^{(1,1)}, C_U^{(1,1)}$ and performing the following procedure, which we will refer to as “biorthogonalization”:

1. Take the following SVD of $C_D^{(1,1)} C_U^{(1,1)} (\equiv \rho_R^{(1)})$:

$$\begin{array}{c} \text{---} C_U^{(1,1)} \text{---} \\ \text{---} C_D^{(1,1)} \text{---} \end{array} \approx \begin{array}{c} \text{---} \bar{V}_L \text{---} \\ \text{---} \Sigma_L^2 \text{---} \\ \text{---} U_L \text{---} \end{array} \quad (4.20)$$

In other words, we take the SVD $C_D^{(1,1)} C_U^{(1,1)} = U_L \Sigma_L^2 V_L^\dagger$. Eq. (4.20) 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:

$$\boxed{P_L} = \text{---} C_U^{(1,1)} \text{---} V_L \text{---} \Sigma_L^+ \quad (4.21)$$

$$\boxed{P_L^-} = \text{---} C_D^{(1,1)} \text{---} \bar{U}_L \text{---} \Sigma_L^+ \quad (4.22)$$

Note the use of a pseudoinverse, which is necessary for numerical stability if the singular values are small and can limit the accuracy of the fixed point environment found by the algorithm.

Note that this presentation of the algorithm is changed slightly from how it is presented in Ref. [108]. First of all, the use of a pseudoinverse was not pointed out in the original proposal, but we find it can become necessary for the stability of the algorithm for sufficiently small singular values (if the singular values are not obtained to high enough precision). Note that in the limit of large bond dimension of the boundary tensors, the spectrum Σ_L^2 in Eq. (4.20) will generally contain small singular

values. Additionally, we present the algorithm without the QR decomposition, which does not change the final projector obtained by the algorithm. Plugging P_L, P_L^- as defined in Eq. (4.21)–(4.22) into Eq. (4.19) and using Eq. (4.20), we see that P_L, P_L^- are approximate inverses of each other as desired (up to numerical errors and errors introduced by the pseudoinverse).

Note that an alternative to this algorithm was also mentioned in Ref. [108], which is to biorthogonalize just the CTMs $C_{DL}^{(1,1)}$ and $C_{UL}^{(1,1)}$ instead of the half system transfer matrices. As mentioned in Ref. [108], this method is not as accurate, though the singular values Σ_L^2 in Eq. (4.20) do fall off slower in that case, and therefore the inverse is better conditioned. In spite of the better conditioned inverse, that method is likely not as accurate because it does not use enough information about the environment to form the projectors.

In the next section, we propose a very related scheme to these two approaches discussed in Ref. [108] which improves the conditioning of the inverse compared to using $C_D^{(1,1)}, C_U^{(1,1)}$ to form the projectors, but does not have the accuracy problems faced by using $C_{DL}^{(1,1)}$ and $C_{UL}^{(1,1)}$.

4.3.3 Obtaining the CTMRG projector: new proposal

Here we present our new proposal for obtaining the projector in the asymmetric CTMRG algorithm:

1. In our new approach, just like in the previous approach, we start with the half system transfer matrices $C_U^{(1,1)}$ and $C_D^{(1,1)}$, but we first perform the following matrix factorizations:

$$\begin{array}{ccc}
 \begin{array}{c} \text{---} \\ | \\ \boxed{C_U^{(1,1)}} \\ | \\ \text{---} \\ \text{---} \\ | \\ \boxed{C_D^{(1,1)}} \\ | \\ \text{---} \end{array} & \approx & \begin{array}{c} \text{---} \\ | \\ \boxed{U_U} \text{---} \boxed{S_U} \text{---} \boxed{\bar{V}_U} \\ | \\ \text{---} \\ | \\ \boxed{\bar{V}_D} \text{---} \boxed{S_D} \text{---} \boxed{U_D} \\ | \\ \text{---} \end{array} \\
 & & \approx \\
 & & \begin{array}{c} \text{---} \\ | \\ \boxed{F_{LU}} \text{---} \boxed{F_{UR}} \\ | \\ \text{---} \\ | \\ \boxed{F_{DL}} \text{---} \boxed{F_{RD}} \\ | \\ \text{---} \end{array} \tag{4.23}
 \end{array}$$

where we define $F_{LU}^s = U_U^s S_U^{1/2}$, $F_{UR}^s = S_U^{1/2} (V_U^s)^\dagger$, $F_{RD}^s = U_D^s S_D^{1/2}$, and $F_{DL}^s = S_D^{1/2} (V_D^s)^\dagger$. 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 projectors P_L, P_L^- . For this, we use the same biorthogonalization procedure as was used in Eq. (4.20)–(4.22). However, this time we biorthogonalize the tensors F_{LU}, F_{DL} :

$$\begin{array}{c} \text{---} F_{LU} \text{---} \\ | \\ \text{---} F_{DL} \text{---} \end{array} \approx \begin{array}{c} \tilde{\Sigma}_L^2 \\ | \\ \begin{array}{c} \bar{Q}_L \\ \bar{W}_L \end{array} \end{array} \quad (4.24)$$

where the second line is obtained by taking the SVD $F_{DL}^s F_{LU}^s = W_L \tilde{\Sigma}_L^2 Q_L^\dagger$ 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:

$$\text{---} P_L \text{---} = \text{---} F_{LU} \text{---} Q_L \text{---} \tilde{\Sigma}_L^+ \text{---} \quad (4.25)$$

$$\text{---} P_L^- \text{---} = \text{---} F_{DL} \text{---} \bar{W}_L \text{---} \tilde{\Sigma}_L^+ \text{---} \quad (4.26)$$

We reiterate that we use a pseudoinverse for $\tilde{\Sigma}_L$, which is important if the singular values are small.

We note that this new method is naturally suited for the bidirectional renormalization approach discussed previously in Section 4.3.1, because the tensor F_{UR} and F_{RD} can be biorthogonalized and directly used for the right move, whereas they would be discarded in the unidirectional approach. An interesting question is how the multi-site algorithms of Ref. [99, 108], which are formulated for the unidirectional approach, would be generalized to the bidirectional approach.

We find in practice that, although this method still involves inverting a matrix that is potentially ill-conditioned, the conditioning is better than the method of Ref. [108]. We show numerical results to this effect in Section 4.4. To help understand why we chose the procedure the way we did, it helps to discuss some alternative approaches, which we do in the next section.

4.3.4 Obtaining the CTMRG projector: alternatives to the new proposal

Our motivation for devising a new CTMRG method was to avoid inverting the singular values of the full environment. We will present numerical results in Section 4.4 showing that the method presented in Section 4.3.3 succeeds at improving

the conditioning of the inverse required for calculating the projectors. However, the approach presented in Section 4.3.2 is clearly not the only scheme that would have a better conditioned inverse than the algorithm shown in Section 4.3.2. To understand why we choose this scheme, it is useful to discuss some alternatives.

Looking back at the factorization we performed in Eq. (4.23), there are many alternative ways to perform the factorization. Different choices of factorizations can lead to different sets of bases to biorthogonalize in order to form the projectors. As we mentioned in Section 4.3.2, an alternative approach suggested in Ref. [108] was to biorthogonalize $C_{LU}^{(1,1)}$ and $C_{DL}^{(1,1)}$, which amounts to a trivial choice of the factorization. This also improves the conditioning of the inverse, but in practice, the method is less stable and does not always find the optimal fixed point environment for a given bond dimension. This is likely due to the fact that it does not make use of a large enough part of the environment to calculate the projector. We can think of our new method as a way to improve the accuracy and stability of that approach, where factorizing with the SVD helps incorporate information from the rest of the environment.

Another nontrivial factorization we tested is to use $F_{LU} = U_U$, $F_{DL} = V_D^\dagger$. This gives a much flatter spectrum of the singular values when performing the biorthogonalization. In practice, however, we found problems with numerical stability with this approach. What we found was that, at larger bond dimensions, small singular values appeared in the spectrum, but we could not safely discard them with a pseudoinverse.

Using the factorization such that $F_{LU} = U_U$, $F_{DL} = V_D^\dagger$ is in fact very related to the method proposed by Huang in Ref. [123] in the context of TMRG. In that context, he proposed forming a biorthogonal basis for non-Hermitian DMRG by taking the SVD of the asymmetric density matrix, truncating according to the singular values, and then biorthogonalizing the bases formed from the remaining left and right singular vectors [123]. In Ref. [123], Huang also pointed out a similar problem that we saw with a few small singular values that ruined the numerical stability of the algorithm. In that paper, Huang proposed fixing this problem with a subspace expansion technique. The closest translation of Huang's method to our current discussion is to take the SVD $\rho_L^{(1)} = C_D^{(1,1)} C_U^{(1,1)} \approx U \Sigma V^\dagger$ and biorthogonalize V^\dagger and U . We translated Huang's proposal into the CTMRG context, but didn't find the subspace expansion technique alleviated the numerical instability that we were observing. It would be an interesting area of further exploration to see if that

approach can be implemented successfully in CTMRG.

A slight adaptation of the full SVD approach of Huang is to biorthogonalize $U\Sigma^{1/2}$ and $\Sigma^{1/2}V^\dagger$ instead of biorthogonalizing V^\dagger and U . With this approach, a psuedoinverse can succussfully stabilize the algorithm. We find that this method performs nearly the same as the method of Corboz et al.

In the next section, we show numerical results for our new asymmetric CTMRG method, and compare its performance to the asymmetric CTMRG approach of Ref. [108] as well as to the fully symmetric CTMRG approach of Nishino and Okunishi.

4.4 Results

Here, we present benchmark results comparing our new asymmetric CTMRG method described in Section 4.3.3 to the asymmetric CTMRG method proposed by Corboz et al. in Ref. [108], which we summarized in Section 4.3.2.

In Section 4.4.1, the 2D classical ferromagnetic Ising model is used as a benchmark, which is useful because it is exactly solvable and the model can be tuned towards the critical point. It is also useful that it has a fully symmetric form of the tensor T , for example by using the construction in Ref. [95], so that we can compare the asymmetric CTMRG methods to the extremely reliable and accurate symmetric version of Nishino and Okunishi [71, 72]. The network is on the square lattice and has a single site unit cell, and all tensors used are real and dense. The environment is chosen to be a product state and the product state is optimized. Then, the environment is grown to the desired bond dimension, and the methods are iterated until convergence.

4.4.1 2D Classical Ising Model

In Figure 4.41, we present benchmark results for the isotropic 2D ferromagnetic classical Ising model. The tensor T comprising the partition function for this model has a representation where each index is of size $d = 2$ and where the tensor is real and symmetric about all rotations and reflections.

However, to test our new algorithm on an asymmetric network, we introduce the

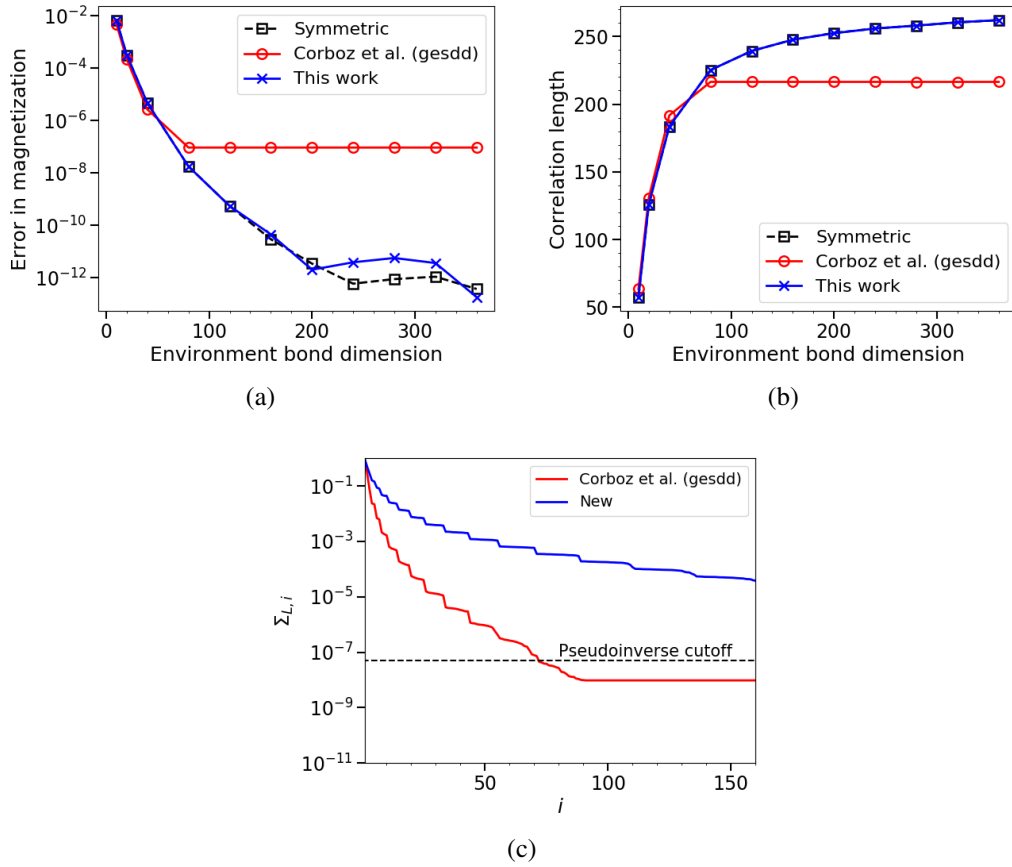


Figure 4.41: Plots comparing our new method for obtaining the CTMRG projector compared to the method proposed in Ref. [108]. We apply these methods to the isotropic ferromagnetic 2D classical Ising model at inverse temperature $\beta/\beta_c - 1 = 10^{-3}$ with a non-unitary change of basis. Here we plot a comparison of: (a) the error in the magnetization as a function of bond dimension, (b) the correlation length as a function of bond dimension, and (c) the spectrum that needs to be inverted in creating the CTMRG projectors. See the main text for details about the computations and analysis of the results.

following gauge transformation:

$$\begin{array}{c} \boxed{\tilde{T}} \end{array} = \begin{array}{c} \begin{array}{ccc} & \textcircled{Y} & \\ & | & \\ \textcircled{X} & \boxed{T} & \textcircled{X^{-1}} \\ & | & \\ & \textcircled{Y^{-1}} & \end{array} \end{array} \quad (4.27)$$

These gauge transformations, for random (complex) non-unitary matrices X and Y , artificially break the rotation and reflection symmetries of the tensor network

representation of the Ising model partition function. The results we present are for a single choice of X and Y , but the results are robust for a sampling of randomly generated X and Y .

In Fig. 4.41 we compare results for our new method with an implementation of the CTMRG approach introduced by Corboz et al. in Ref. [108] and summarized in Section 4.3.2. For this implementation of the method of Corboz et al., in Eq. (4.20) we use the gesdd SVD routine as implemented in LAPACK, which is a variant that is commonly used as a default in many numerical libraries.

For larger bond dimensions, we found that (when gesdd is used as the SVD routine) we needed to set the pseudoinverse cutoff to 5×10^{-8} to get the method of Corboz et al. to converge consistently. In practice this limits the accuracy in the magnetization to be $O(10^{-7})$, as seen in Fig. 4.41a. For our CTMRG method, we use a pseudoinverse of 1×10^{-7} , which we find only plays a role in stabilizing the algorithm early on in the calculation, soon after the bond dimension growth. In Fig. 4.41a, we plot the error in the magnetization for the Ising model at an inverse temperature of $\beta = (1 + 10^{-3})\beta_c$, where β_c is the exact critical temperature of the Ising model, and we compare to the exact result for the magnetization of the Ising model.

For reference, in Fig. 4.41a, we include results for the same network without any gauge transformations added, and where the fully symmetric version of CTMRG described in Ref. [71, 72] can be applied. This can be considered to be the most accurate CTMRG method for contracting the network for a given bond dimension. We see that our new approach for obtaining the CTMRG projectors in the asymmetric case obtains the same level of precision as the symmetric method (variations at higher bond dimensions are within the limits of noise fluctuations from step to step of CTMRG).

We also plot the correlation length calculated from the same networks that were used to calculate the magnetization in Fig. 4.41b. The correlation length is estimated by using the two dominant eigenvalues of the approximate column-to-column transfer matrix formed from the HCTMs A_U and A_D . Details about how these eigenvalues relate to the correlation length are explained in Ref. [105]. We see that the lower precision obtained by the method of Corboz et al. is even more noticeable for nonlocal observables like the correlation length. Calculating nonlocal observables to high precision can be particularly important for newly proposed gradient-based iPEPS optimization techniques, where it is found that the gradient must be obtained to very high accuracy to obtain proper convergence [121].

In Fig. 4.41c, we compare the spectra of the matrices that have to be inverted to obtain the CTMRG projectors (for this case a bond dimension for the environment of $\chi = 160$ is used). For the CTMRG method of Corboz et al., the spectrum plotted is the (ordered) diagonal entries of Σ_L , labelled $\Sigma_{L,i}$, the square root of the singular values obtained from Eq. (4.20) (normalized so that $\Sigma_{L,1} = 1$). For our new CTMRG method, we plot the (ordered) diagonal entries of $\tilde{\Sigma}_L$, $\tilde{\Sigma}_{L,i}$, obtained from Eq. (4.24) (normalized so that $\tilde{\Sigma}_{L,1} = 1$). The horizontal dotted line indicates the pseudoinverse of 5×10^{-8} , which is necessary for this example for the method of Corboz et al. to be numerically stable.

Fig. 4.41c gives us an indication for why the new method obtains a higher precision than the old method. We see that, in this case, the gesdd SVD routine only calculates singular values up to a precision of $O(10^{-16})$, so the pseudoinverse must be used to avoid large errors caused by inverting the small, inaccurate square roots of those singular values. Because of the pseudoinverse cutoff, only around 80-100 of the singular values are kept. This exactly corresponds with the limits of the accuracy of the magnetization and correlation length shown in Fig. 4.41a–4.41b. For the new method, after the truncation of the bond dimension back down to $\chi = 160$, the smallest value in the spectrum is $O(10^{-5})$, and the inverse is well-conditioned without the need of a pseudoinverse cutoff. Note that other SVD routines, such as the more computationally demanding gesvd implemented in LAPACK, can calculate singular values to higher precision than $O(10^{-16})$ for this example and therefore can allow the method of Corboz et al. to calculate more accurate environments. This is discussed more in Ref. [3]. We emphasize that because of the improved conditioning of the inverse for our newly proposed CTMRG method, it is not sensitive to which SVD routine is used.

4.5 Conclusion

Numerical methods based on corner transfer matrices (CTMs) have a long history, going back to their introduction in 1968 by Baxter [56–58]. Nishino and Okunishi [71, 72] proposed an improvement to the CTM method of Baxter using insights from the density matrix renormalization group (DMRG) [5, 6] algorithm of White. Since then, CTMRG has been used very successfully in a wide range of applications, from studying 2D classical statistical mechanics models to some of the most advanced calculations of strongly interacting 2D quantum many-body systems.

The CTMRG method proposed by Nishino and Okunishi was originally formulated

primarily for contracting tensor networks with many lattice symmetries, and in that context it is remarkably stable and accurate. Many methods have been proposed for extending the method to general asymmetric tensor networks, but each has their own limitations, as we have reviewed in this chapter. As we discussed in this work, either they are not appropriate for contracting very asymmetric tensor networks, or they rely on potentially ill-conditioned inverses, such as the method most commonly used in advanced iPEPS calculations proposed in Ref. [108]. In this manuscript, we propose a new CTMRG method that improves the conditioning of these inverses, providing a more numerically robust variant of CTMRG for contracting asymmetric 2D tensor networks.

FASTER METHODS FOR CONTRACTING INFINITE TWO-DIMENSIONAL TENSOR NETWORKS

¹M. T. Fishman, L. Vanderstraeten, V. Zauner-Stauber, J. Haegeman, and F. Verstraete, “Faster Methods for Contracting Infinite 2D Tensor Networks”, [arXiv:1711.05881](https://arxiv.org/abs/1711.05881).

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 (iPEPS) methods.

5.1 Introduction

Two-dimensional (2D) tensor networks are ubiquitous in many-body physics [4]. They occur naturally in the context 2D classical many-body systems as representations of partition functions [56–59, 70–72] 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 [48, 49, 66–69, 129, 130, 135]. Additionally, they occur in the context of tensor product state (TPS) [87, 88, 90, 92] or projected entangled pair state (PEPS) [93] 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 [5, 6, 47–49, 56–59, 70–72, 93, 123, 124, 128–131, 136–142].

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) [136] and

extensions such as second renormalization group (SRG) [137], higher order tensor renormalization group (HOTRG) [138], and tensor network renormalization (TNR) [139, 141, 142]. 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 matrix product state has to be determined in a variational way. Boundary methods include the density matrix renormalization group (DMRG) [5, 6, 70, 129–131], corner transfer matrix renormalization group (CTMRG) [56–58, 71, 72], time evolving block decimation (TEBD) [48, 49, 55, 128], TDVP [50, 51], etc. Boundary methods have certain advantages: they can be optimized iteratively (instead of optimized layer by layer like many 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 [93, 115, 120, 121].

The history of modern boundary methods goes back to Nishino’s application of DMRG to calculating fixed points of transfer matrices [70]. Soon after, Nishino and Okunishi created their CTMRG algorithm [71, 72] by combining the corner transfer matrix (CTM) of Baxter [56–58] and White’s DMRG algorithm [5, 6], CTMRG was initially introduced as a powerful numerical tool for contracting 2D classical partition functions. In addition, it is used extensively in TPS and 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 [88, 90–92]. An MPS-based boundary method was used for the original finite PEPS calculation [93] while iTEBD, an MPS-based power method, was used to perform the original infinite PEPS (iPEPS) [94, 143] calculations. Since then, iPEPS calculations 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 [82, 95, 97–99, 108, 113–115].

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) [2] 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.

5.2 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, and discuss the extension to multi site unit cells where appropriate. We are agnostic about where the tensor network comes from: it could be a 2D classical partition function, the norm of an iPEPS, etc.

For concreteness, we are interested in evaluating:

$$\Omega^{MN} = \text{Tr} \left[\begin{array}{cccc} \vdots & \vdots & \vdots & \vdots \\ \cdots & \boxed{T} & \boxed{T} & \boxed{T} & \boxed{T} & \cdots \\ \cdots & \boxed{T} & \boxed{T} & \boxed{T} & \boxed{T} & \cdots \\ \cdots & \boxed{T} & \boxed{T} & \boxed{T} & \boxed{T} & \cdots \\ \cdots & \boxed{T} & \boxed{T} & \boxed{T} & \boxed{T} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{array} \right] \quad (5.1)$$

where we work directly in the thermodynamic limit, i.e. the number of lattice sites in the horizontal and vertical directions, M, N , approaches ∞ . In Eq. (5.1), $\text{Tr}[\dots]$ 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 Ω is the ‘‘partition function per site,’’ related to the free energy per site. If the network is the evaluation of the norm of an iPEPS, each tensor T is the bra and ket PEPS tensor at each site contracted over the physical index¹, and Ω 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

¹The 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.

represented as impurity sites, such as:

$$\langle XY \rangle = \text{Tr} \left[\begin{array}{cccc} \vdots & \vdots & \vdots & \vdots \\ \cdots & \boxed{T} & \boxed{T} & \boxed{T} & \boxed{T} & \cdots \\ \cdots & \boxed{T} & \boxed{T} & \boxed{T} & \boxed{T_Y} & \cdots \\ \cdots & \boxed{T} & \boxed{T} & \boxed{T} & \boxed{T} & \cdots \\ \cdots & \boxed{T_X} & \boxed{T} & \boxed{T} & \boxed{T} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \end{array} \right] / \Omega^{MN} \quad (5.2)$$

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 iPEPS ground state optimizations[115, 120, 121]. 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:

$$\cdots \boxed{T} \boxed{T} \boxed{T} \boxed{T} \cdots \quad (5.3)$$

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:

$$\begin{array}{c} \vdots \\ \boxed{T} \\ \boxed{T} \\ \boxed{T} \\ \boxed{T} \\ \vdots \end{array} \quad (5.4)$$

For MPS boundary methods, the evaluation of diagrams like Eq. (5.1)–(5.2) is performed by finding the maximal up and down eigenvectors of the row-to-row transfer matrix portrayed in (5.3) and the maximal left and right eigenvectors of column-to-column transfer matrix portrayed in (5.4). Exact MPS representations of these eigenvectors are infinitely large, but for many 2D tensor networks representing physical many-body systems good MPS approximations exist (in some cases provably[39]). We refer to these uniform MPS (uMPS) fixed points as the up, down, left and right boundary uMPSs, and call their MPS tensors respectively A_U , A_D , A_L and A_R . The fixed point equation for the top uMPS is as follows:

$$\begin{array}{c}
 \cdots \textcircled{A_U} \textcircled{A_U} \textcircled{A_U} \textcircled{A_U} \cdots \\
 \vdots \\
 \cdots \boxed{T} \boxed{T} \boxed{T} \boxed{T} \cdots \\
 \vdots \\
 \Omega^M \cdots \textcircled{A_U} \textcircled{A_U} \textcircled{A_U} \textcircled{A_U} \cdots
 \end{array} \approx \quad (5.5)$$

where Ω is the partition function or norm per site defined in Eq. (5.1). At the fixed point, analogous equations to Eq. (5.5) should be satisfied by the other boundary uMPSs.

There are many different approaches to obtaining the four boundary MPS fixed points of the row-to-row transfer matrix (5.3) and column-to-column transfer matrix (5.4). In the next section, we review one very commonly used contraction method, the corner transfer renormalization group (CTMRG) algorithm of Nishino and Okunishi[56–58, 71, 72], and describe two new proposals, one based on the recently proposed variational uniform matrix product state (VUMPS) algorithm[2], 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.

5.3 Algorithm overview

One strategy for evaluating Eq. (5.1)–(5.2) involves finding a single boundary MPS eigenvector at a time. The infinite time evolving block decimation (iTEBD) of Orús and Vidal [48, 49, 128] 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 [2], which can be viewed as an improvement on the infinite density matrix renormalization group

(iDMRG) [5, 6, 47] 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. In analogy to how Nishino showed that DMRG could be applied to finding the fixed point MPS approximation of the transfer matrix of a partition function [70], in Section 5.3.4, we show how VUMPS can be applied to find the fixed point of infinite transfer matrices.

Another strategy for finding the boundary uMPSs is to attempt to find all four uMPSs at once. An example of this approach is the corner transfer matrix (CTM) method of Baxter [56–58], and its improvement by Nishino and Okunishi called the corner transfer matrix renormalization group (CTMRG) [71, 72]. These methods are variational methods for iteratively finding all four MPS fixed points at once. We give a brief review of CTMRG in Sections 5.3.1, 5.3.2 and 5.3.3. 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 5.3.5–5.3.6, 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.

5.3.1 Corner transfer matrix renormalization group (CTMRG) review

The general ansatz used for the environment in the corner transfer matrix renormalization group (CTMRG) algorithm is as follows:

$$\begin{array}{cccc}
 \boxed{C_{LU}} & \circ A_U & \circ A_U & \boxed{C_{UR}} \\
 \circ A_L & \boxed{T} & \boxed{T} & \circ A_R \\
 \circ A_L & \boxed{T} & \boxed{T} & \circ A_R \\
 \boxed{C_{DL}} & \circ A_D & \circ A_D & \boxed{C_{RD}}
 \end{array} . \tag{5.6}$$

The matrices $\{C_i\}$ in Eq. (5.6), known as the corner transfer matrices (CTMs), were originally introduced by Baxter for studying 2D classical statistical mechanics problems[56–58, 71]. The CTMs represent approximations of the infinite corners of the tensor network. The boundary MPS tensors $\{A_i\}$ in Eq. (5.6) 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. (5.6) can be constrained.

5.3.2 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 $T_{lurd} = \bar{T}_{ruld} = \bar{T}_{ldru} = \bar{T}_{drul}$). This is the case for many statistical mechanics models. In this case, we can constrain the environment tensors in the ansatz in Eq. (5.6) 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 $A^s = (A^s)^\dagger$. Eq. (5.6) becomes:

$$\begin{array}{cccc}
 \boxed{C} & \circledast A & \circledast A & \boxed{C} \\
 \circledast A & \boxed{T} & \boxed{T} & \circledast A \\
 \circledast A & \boxed{T} & \boxed{T} & \circledast A \\
 \boxed{C} & \circledast A & \circledast A & \boxed{C}
 \end{array}
 \tag{5.7}$$

This is the CTMRG case that was covered in the initial proposal of Nishino and Okunishi [71, 72] (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 ²:

The diagram shows a tensor network on the left consisting of four tensors: C (top-left), A (top-right), A (bottom-left), and T (bottom-right). C and A are connected horizontally, A and T are connected horizontally, and C and A are connected vertically, as are A and T . This network is approximately equal to a tensor network on the right. The right network has a vertical line on the left, a horizontal line at the bottom, and a vertical line on the right. It contains two tensors: U (bottom) and \bar{U} (top), connected vertically. To the right of U is a diagonal tensor D , connected horizontally to U and vertically to \bar{U} . The right vertical line is connected to \bar{U} .

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. (5.8), 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 diagrammatically:

The diagram shows two tensors, U (top) and \bar{U} (bottom), connected vertically. The left side of U and \bar{U} are connected by a horizontal line, and the right side of U and \bar{U} are connected by a horizontal line. This is equal to a vertical line representing the identity I .

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 diagrammatically is:

The diagram shows two tensors, \bar{U} (top) and U (bottom), connected vertically. The left side of \bar{U} and U are connected by a horizontal line, and the right side of \bar{U} and U are connected by a horizontal line. This represents the projector $U^s(U^{s'})^\dagger$.

The projector Eq. (5.10) 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'

²The original proposal actually involved a symmetric eigendecomposition of a product of four of the grown corners in Eq. (5.8) which has the interpretation of a density matrix, but the eigenbasis is the same as that of a single corner.

and A' are obtained as follows:

$$\begin{array}{c} \boxed{C'} \end{array} = \begin{array}{c} \begin{array}{c} \boxed{C} \quad \boxed{A} \\ \boxed{A} \quad \boxed{T} \quad \boxed{U} \\ \quad \quad \boxed{\bar{U}} \end{array} \end{array} \quad (5.11)$$

$$\begin{array}{c} \boxed{A'} \end{array} = \begin{array}{c} \quad \quad \boxed{U} \\ \boxed{A} \quad \boxed{T} \\ \quad \quad \boxed{\bar{U}} \end{array} . \quad (5.12)$$

Of course, from Eq. (5.8), we can trivially see that $C' = D$, but the more general form of Eq. (5.11) 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 corner matrices 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. (5.8) 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.

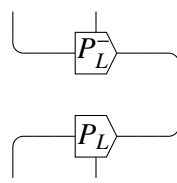
If the tensor network is not Hermitian about either a horizontal or vertical reflection, a simple Hermitian eigendecomposition like that shown in Eq. (5.8) will not suffice, and a more involved scheme must be invoked. In general, the projectors used will not be isometric, and there will be different projectors for renormalizing each direction (left, up, right, and down). We refer readers to Ref. [82] for a discussion of exploiting different types of reflection symmetries and alternative approaches to finding projectors.

Next, we will discuss strategies for generalizing CTMRG to asymmetric tensor networks.

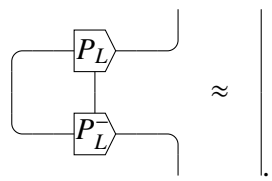
5.3.3 Asymmetric CTMRG review

When the tensor network does not contain reflections 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 for 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 iPEPS calculation [99, 108].

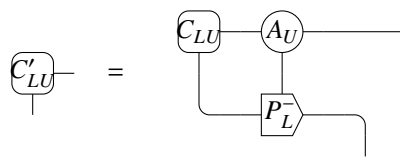
In the directional approach, the “left move” involves contracting just the column-to-column transfer matrix into the left environment, and renormalizing with a projector which we will call $P_L^s [P_L^-]^{s'}$ or diagrammatically:

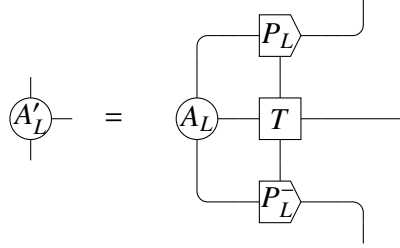

(5.13)

where P_L^- is the approximate left inverse of P_L , in other words they satisfy $[P_L^-]^s P_L^s \approx I$ or diagrammatically:


(5.14)

Using the projector in Eq. (5.13), the left move is shown below:


(5.15)


(5.16)

$$C_{DL}' = \begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} C_{DL}' \text{---} = \begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} C_{DL} \begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} A_D \begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} P_L \begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} \text{---} \quad (5.17)$$

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. (5.13)) 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 [71, 82, 95, 97–99, 108]. In this work, for asymmetric CTMRG, we will use the method that is most commonly used in iPEPS calculations, the one proposed in Ref. [108]. In addition, we will use a newly proposed method for obtaining the projector, introduced in Chapter 4. For brevity, we do not review those methods here, and we refer readers to those works for an explanation of how the projectors are obtained.

5.3.4 VUMPS for contracting infinite 2D tensor networks

Here, we present the application of the recently proposed VUMPS algorithm [2] 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. (5.5).

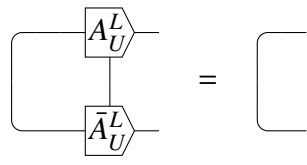
We now present VUMPS for obtaining the top fixed point uMPS of the network. We would like to find the uMPS satisfying Eq. (5.5). In VUMPS, we use the mixed canonical form of the uMPS, so Eq. (5.5) becomes:

$$\begin{array}{c} \dots \text{---} A_U^L \text{---} A_U^L \text{---} A_U^C \text{---} A_U^R \text{---} A_U^R \text{---} \dots \infty \\ | \quad | \quad | \quad | \quad | \\ \text{---} T \text{---} T \text{---} T \text{---} T \text{---} T \text{---} \\ | \quad | \quad | \quad | \quad | \\ \dots \text{---} A_U^L \text{---} A_U^L \text{---} A_U^C \text{---} A_U^R \text{---} A_U^R \text{---} \dots \end{array} \quad (5.18)$$

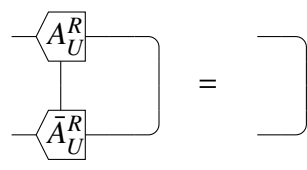
In the mixed canonical gauge, the tensors satisfy the relations:

$$\begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} A_U^L \begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} C_U \text{---} \approx \begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} A_U^C \text{---} \approx \begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} C_U \begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} A_U^R \text{---} \quad (5.19)$$

where the singular values of the matrix C_U are the Schmidt values of the uMPS. Note the inequalities in Eq. (5.19), 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:



$$(5.20)$$

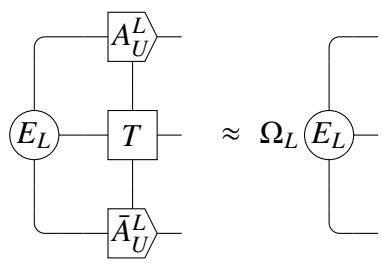


$$(5.21)$$

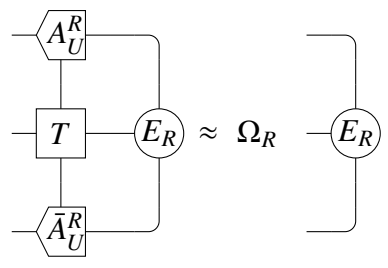
at all times. Any uMPS can be turned into this form, for example with the algorithm introduced in Ref. [128] or with the algorithm discussed in Ref. [4] and expanded on in Appendix 5.A.

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

1. Solve for the environments:



$$(5.22)$$



$$(5.23)$$

where $\Omega_L \approx \Omega_R$ up to errors in Eq. (5.19).

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

$$\approx \Omega_{C_U} \text{---} C_U \text{---}$$
(5.24)

$$\approx \Omega_{A_U^C} \text{---} A_U^C \text{---}$$
(5.25)

where $\Omega_{A_U^C}/\Omega_C \approx \Omega_{L/R}$ near or at the fixed point.

3. Solve for the new uMPS tensors A_U^L and A_U^R from A_U^C and C_U using techniques described in the original proposal[2].

The VUMPS algorithm proceeds by repeating steps 1–3 until convergence (measured, for example, by the change in the singular values of C from step to step or the gradient, as described in Ref. [2]). 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[2], 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 point 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, the gauged uMPS 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 are related to the product of CTMs $C_{LU}C_{LR}$. This correspondence is discussed in more detail in Ref. [4]. 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 [70–72].

For contracting 2D statistical mechanics partition functions and calculating the norm of an iPEPS, transfer matrix VUMPS is in fact simpler than the original proposal, because we do not have to be concerned about summing Hamiltonian terms which can lead to divergences if the fixed point is not calculated properly [2, 144], and methods such as Arnoldi can be directly employed to find the fixed points. One may

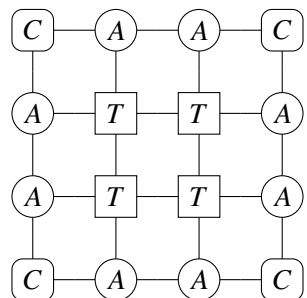
have to be more careful contracting networks that involve sums of local operators, such as when calculating structure factors or gradients of iPEPS. See Ref. [115, 120, 121] for approaches to contracting such networks, where the environments calculated from the norm of an iPEPS are used to aid in the 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 uMPSs 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 uMPS. 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 in the same phase (for example by using starting states for the optimization which are in the same phase).

For non-Hermitian networks, the method we propose here is analagous to iTEBD, where each of the four boundary uMPSs is solved for in seperate 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. [2]). 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 [123–125, 129–134, 145]. 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.

5.3.5 Symmetric FPCM

We start with the simplest version of our new fixed point corner method (FPCM), when the network is comprised of a real tensor T that is symmetric 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 5.3.2 and repeat here:



(5.26)

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

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

$$\text{---} \boxed{C'} \text{---} \bigcirc A \text{---} \propto \text{---} \boxed{U} \text{---} \boxed{C'} \text{---} . \quad (5.27)$$

This is performed with a new uMPS gauging method described in Appendix 5.A.

2. Obtain the new MPS boundary tensor A 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):

$$\begin{array}{c} \text{---} \bigcirc A' \text{---} \\ \text{---} \boxed{U} \text{---} \\ \text{---} \boxed{T} \text{---} \\ \text{---} \boxed{\bar{U}} \text{---} \end{array} \propto \text{---} \bigcirc A' \text{---} . \quad (5.28)$$

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. (5.28) 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.

We would also like to point out that obtaining the isometry as proposed in Eq. (5.27) can be viewed as a translationally invariant version of the so-called ‘‘simplified one-directional 1D method’’ discussed in Ref. [82]. At the fixed point of the algorithm, we can interpret the uMPS formed by U solved for in Eq. (5.27) as the translationally invariant thermodynamic limit of the analogous isometry U formed from eigenvectors of the grown corner transfer matrix at the symmetric CTMRG fixed point.

After convergence, a higher accuracy for observables can be obtained by using the corner transfer matrix C calculated with the following fixed point equation:

$$(5.29)$$

A similar fixed point equation for the corner transfer matrix was discussed previously in Ref. [120, 121].

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

5.3.6 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. (5.15)–(5.17), 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. 5.3.3):

$$(5.30)$$

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

1. “Biorthogonalize” the top and bottom uMPSs 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:

$$\begin{array}{c} \text{---} C'_{LU} \text{---} A_U \text{---} \\ | \\ \text{---} P_L \text{---} C'_{LU} \text{---} \end{array} \propto \quad (5.31)$$

$$\begin{array}{c} \text{---} C'_{DL} \text{---} A_D \text{---} \\ | \\ \text{---} P_L^- \text{---} C'_{DL} \text{---} \end{array} \propto \quad (5.32)$$

where P_L, P_L^- satisfy Eq. (5.14). There are multiple possible methods for finding tensors P_L, P_L^- and C'_{LU}, C'_{DL} that satisfy Eq. (5.31)–(5.32), and the choices are not unique. The method we use is described in detail in Appendix 5.B.

2. Obtain the left and right environments 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):

$$\begin{array}{c} \text{---} P_L \text{---} \\ | \\ A'_L \text{---} T \text{---} \\ | \\ \text{---} P_L^- \text{---} \end{array} \propto A'_L \quad (5.33)$$

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 the 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 at each step, the CTMs can be obtained in an alternative way from the

corner transfer fixed point equations:

(5.34)

which are generalizations of Eq. (5.29), the symmetric corner transfer fixed point equation. We find that obtaining the corners with these equations leads to a more accurate environment than those found in Eq. (5.31)–(5.31), which is particularly important for calculating accurate observables, or if the step of FPCM is to be followed by a step of CTMRG.

The algorithm looks very similar to the VUMPS algorithm when the network is Hermitian about a certain 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 corner matrices 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. [4].

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 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. However, we will see in the next section that exploiting fixed points leads to a large speedup in total convergence time, because substantially fewer steps are needed for convergence. The speedup 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 row or column of tensors is 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 [146]). In addition, the projectors that are used for renormalization in the new corner method are obtained from the current guess for the entire (translationally invariant) boundary, not just a set of local tensors.

5.4 Results

Here, we present benchmark results for the methods described in the previous section: CTMRG, VUMPS, and the new fixed point corner method (FPCM). We benchmark the 2D classical ferromagnetic Ising model in Section 5.4.1, the 2D classical XY model in Section 5.4.2, the 2D quantum spin-1/2 Heisenberg model in Section 5.4.3, and the chiral Resonating Valence Bond (RVB) iPEPS in Section 5.4.4.

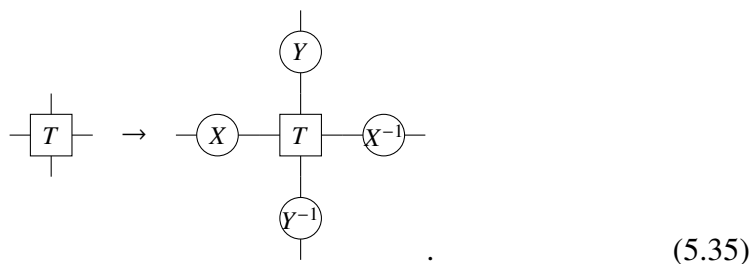
For all of the examples shown, the networks are on the square lattice and have a single-site unit cell, and all tensors used are dense. Calculations were performed with a single BLAS thread. For a fair comparison between different methods, the starting boundary states are chosen to be small (usually 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. [2] is used for VUMPS). Most of the calculations were performed using the Extreme Science and Engineering Discovery Environment (XSEDE) [147] with Intel Math Kernel Library (MKL), except calculations in Fig. 5.43, which were performed on a laptop with OpenBLAS. Fixed points are calculated using the Arnoldi method as implemented in ARPACK.

5.4.1 2D classical Ising model

In Figures 5.41 and 5.42, 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. (5.6), 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. (5.18), 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 uMPS equal to the top fixed point uMPS. For CTMRG, we find the projector to renormalize the boundary using a symmetric eigendecomposition, which is fast and numerically very stable.

From Figures 5.41 and 5.42, 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 over CTMRG increases. This can be understood by the fact that the boundary tensors for the new methods are obtained from 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” [148–151].

In Figure 5.43, we present results for the 2D ferromagnetic classical Ising model with a gauge transformation introduced on the links, as follows:



$$\begin{array}{c}
 \begin{array}{c} | \\ \boxed{T} \\ | \end{array} \rightarrow \begin{array}{c} \textcircled{Y} \\ | \\ \textcircled{X} - \boxed{T} - \textcircled{X^{-1}} \\ | \\ \textcircled{Y^{-1}} \end{array} . \quad (5.35)
 \end{array}$$

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 an iPEPS 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.

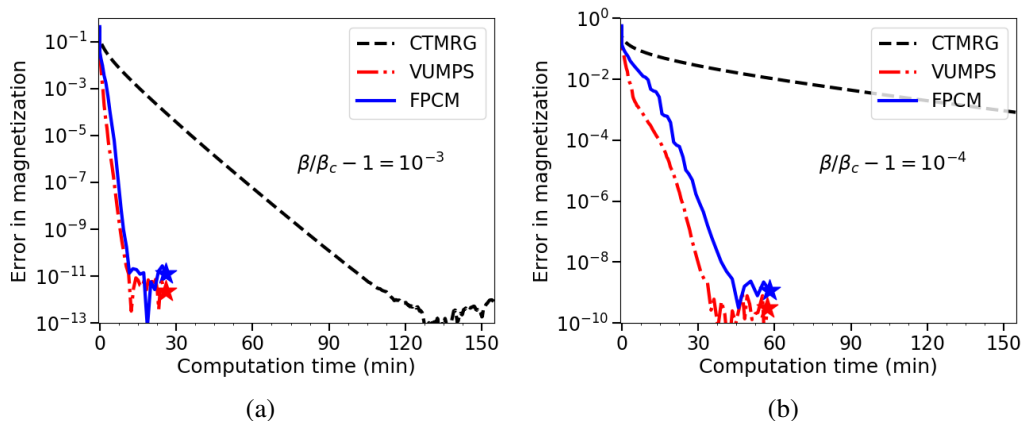


Figure 5.41: Plots of the error in the magnetization for the isotropic 2D classical Ising model at two temperatures near criticality, where (b) is closer to criticality than (a). The network has a bond dimension of $d = 2$, and a boundary MPS bond dimension of $\chi = 600$ is used. A fully symmetric 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.

We compare four different methods. In Fig. 5.43a, we show results using the asymmetric CTMRG method proposed by Corboz et al. in Ref. [108]. For this example, the LAPACK routine `gesdd`, a common default in many numerical libraries, is used for the SVD. Because of a limit we find in the precision for this SVD routine, a pseudoinverse cutoff of 5×10^{-8} must be used for the method to be numerically stable, as discussed in more detail in Chapter 4. Note that a higher precision can be obtained using the more computationally expensive LAPACK routine `gesvd` for the SVD, as we show in Ref. [3]. In Fig. 5.43b, we show results for performing a step of the FPCM introduced in Section 5.3.6 every five steps of the CTMRG method of Corboz et al., which we see substantially speeds up the convergence time, but has the same limit in precision when the `gesdd` LAPACK routine is used in the biorthogonalization step of CTMRG.

In Fig. 5.43c, we show results for a newly proposed asymmetric CTMRG method introduced in Chapter 4. This method is a variation of the method of Corboz et al. to improve the conditioning of the inverse to allow for a higher precision. In Fig. 5.43c, we show that performing FPCM every few steps of CTMRG (in practice we use five) substantially speeds up the convergence time. We refer readers to Ref. [108] and

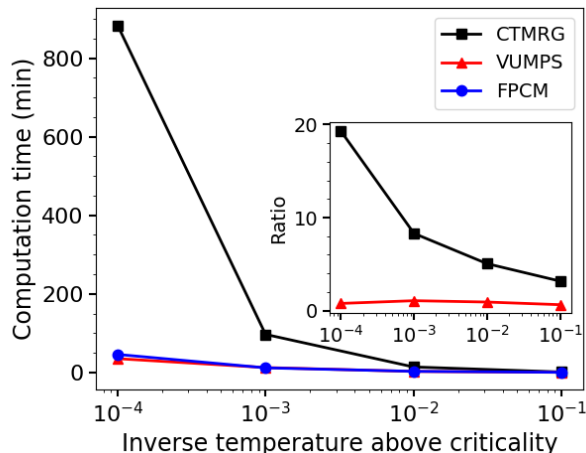


Figure 5.42: Convergence time as a function of inverse temperature above criticality, $\beta/\beta_c - 1$, for the 2D classical Ising model. For all data points, a boundary MPS bond dimension of $\chi = 600$ is used. All data is converged to an error in the magnetization of $< 2 \times 10^{-9}$. The inset shows the ratio of the convergence time of CTMRG and VUMPS with respect to the FPCM convergence time.

Chapter 4 for more details on the asymmetric CTMRG methods that we use here.

Note that we found that the asymmetric FPCM method as presented in Section 5.3.6 alone has a tendency to “get stuck,” i.e., not find the proper 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.

5.4.2 2D classical XY model

In Figure 5.44a, 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 a transfer matrix DMRG (TMRG) [152], which is a method that is closely related to CTMRG [70–72] and the transfer matrix version of VUMPS presented in Section 5.3.4. Additionally, related models have been studied previously with CTMRG [75, 76]. The construction we use for the partition function is described in Ref. [153, 154], where HOTRG was used to contract the partition function, and we refer readers to those references for details. We refer readers to those previous 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

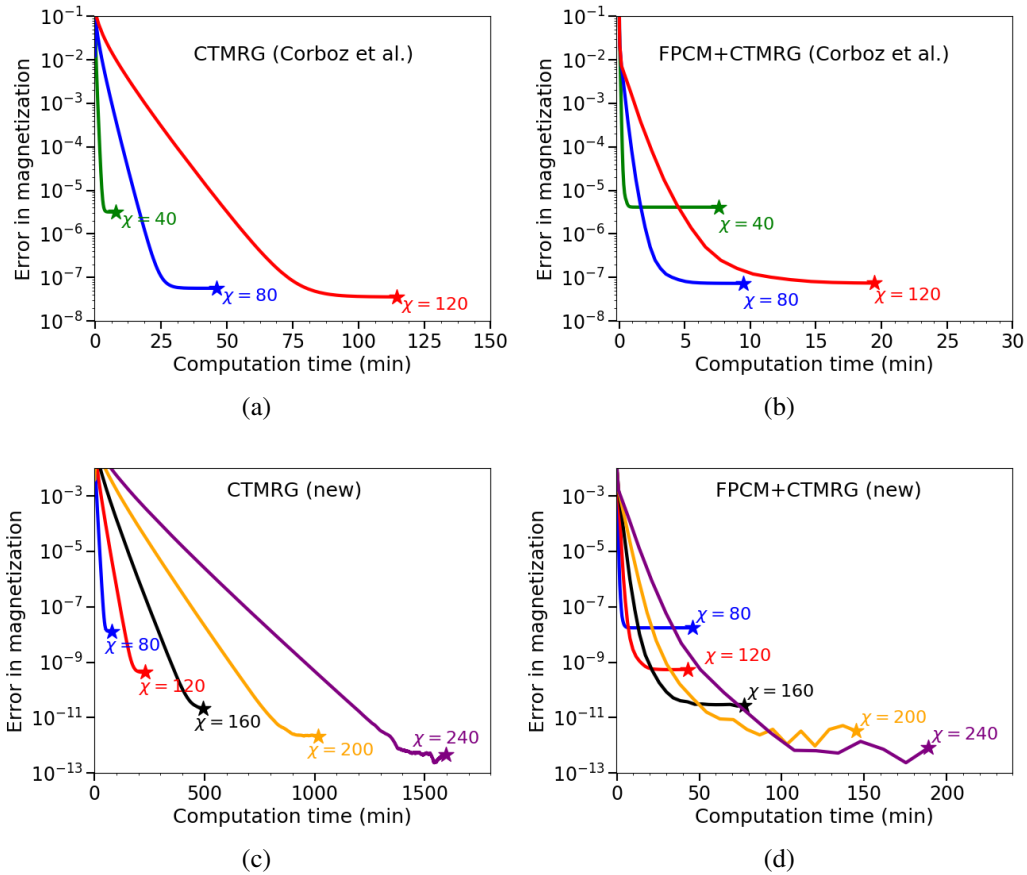


Figure 5.43: Plots of error in magnetization for the isotropic ferromagnetic 2D classical Ising model at $\beta/\beta_c - 1 = 10^{-3}$ with random non-unitary gauge transformations introduced on the horizontal and vertical links, as shown in Eq. (5.35). This artificially breaks the lattice symmetry in order to test each method on an asymmetric network. (a) shows results for the asymmetric CTMRG algorithm by Corboz et al. in Ref. [108], (b) shows results for the FPCM introduced in this work combined with the CTMRG algorithm used in (a), (c) shows results for a newly introduced CTMRG algorithm introduced in Chapter 4, and (d) shows results for the FPCM introduced in this work combined with the CTMRG algorithm used in (c). For more details about the methods used, we refer readers to the main text.

an approximation for the MPO with a link bond dimension of $d = 25$. We use zero applied magnetic field, and at this temperature the model is known 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 note about the x and y axes. The environments we use for all methods is restricted to being real. For CTMRG

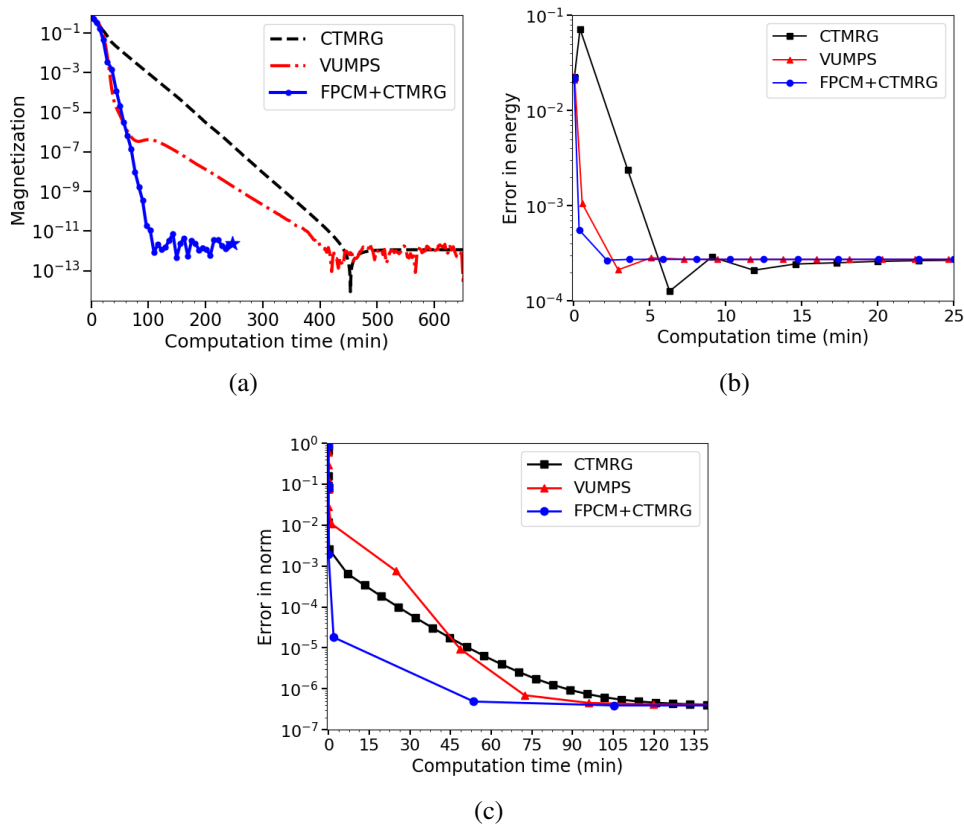


Figure 5.44: (a) Plot of magnetization for the 2D classical XY model, for network bond dimension $d = 25$ and boundary MPS bond dimension $\chi = 50$. (b) Plot of error in energy (compared to Monte Carlo results) for the 2D quantum Heisenberg model. The network bond dimension is $d = 25$ (or PEPS bond dimension $\sqrt{d} = 5$), 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 $\sqrt{d} = 3$), and the boundary MPS bond dimension is $\chi = 800$.

and the FPCM, in the ansatz in Eq. (5.6), 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, in Eq. (5.18), we don’t impose any symmetries, but when we calculate observables, we set the bottom fixed point uMPS equal to the transpose of the top fixed point uMPS (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 5.4.1, 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 used for VUMPS cannot 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.

5.4.3 2D quantum Heisenberg model

In Figure 5.44b, we present results for contracting an iPEPS approximation to the ground state of the 2D quantum Heisenberg model. The iPEPS tensor was optimized using the conjugate gradient method described in Ref. [121]. We plot the error in the energy relative to the energy obtained from Monte Carlo simulations [155].

The iPEPS 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 iPEPS is also complex and symmetric about all rotations and reflections. The environment we use for all methods are necessarily complex. For CTMRG and the FPCM, in the ansatz in Eq. (5.6), 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. (5.18), we don't impose any symmetries ³, but when we calculate observables, we set the bottom fixed point uMPS equal to the top fixed point uMPS (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. [108], 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 5.3.6, where the symmetry of the network is exploited wherever possible. Additionally, we use a modification of the uMPS biorthogonalization procedure in Appendix 5.B, where we first gauge the uMPSs isometrically before we biorthogonalize them. As previously mentioned in Sections 5.4.1–5.4.2, 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 convergence time.

³One 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.

5.4.4 Chiral resonating valence bond iPEPS

In Figure 5.44c, we present results for contracting a chiral resonating valence bond (RVB) iPEPS. The chiral RVB PEPS state was introduced as a chiral extension of the traditional nearest neighbor RVB PEPS [156, 157]. As in the 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. (5.6), 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. (5.18) we do not impose any symmetries⁴. When we calculate observables, we set the bottom fixed point uMPS obtained from VUMPS equal to the complex conjugate of the top fixed point uMPS. 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 uMPS.

Again, we see an improvement in performance of the FPCM and VUMPS over CTMRG, but the new fixed point corner method 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.

5.5 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 iPEPS. 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 iPEPS calculations. We found

⁴One 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.

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 Lanczos 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 iPEPS optimization techniques, where the contraction of the network is the most computationally expensive step. When combined with recently introduced variational methods for optimizing iPEPS[115, 121], we expect that significant improvements can still be made to existing iPEPS algorithms.

APPENDIX

5.A New algorithm for isometrically gauging a uMPS

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:

$$\text{---} \boxed{C} \text{---} \boxed{A} \text{---} \propto \text{---} \boxed{U} \text{---} \boxed{C} \text{---} \quad (5.36)$$

$$\begin{array}{c} \text{---} \boxed{U} \text{---} \\ | \\ \text{---} \boxed{\bar{U}} \text{---} \end{array} = \left. \begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} \right. \quad (5.37)$$

A method for finding the orthogonal gauge for a uMPS was first proposed in the context of the iTEBD algorithm[128] 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 uMPS 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:

$$\begin{array}{c} \boxed{C^2} \text{---} \boxed{A} \text{---} \\ | \\ \text{---} \boxed{\bar{A}} \text{---} \end{array} \propto \begin{array}{c} \boxed{C^2} \text{---} \\ | \\ \text{---} \end{array} \quad (5.38)$$

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:

$$\text{---} \boxed{C} \text{---} \boxed{A} \text{---} = \text{---} \boxed{U} \text{---} \boxed{P} \text{---} \quad (5.39)$$

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 maximal eigenvector of the fixed point equation:

$$\text{Diagram: } \square_C \text{---} \circlearrowleft_A \text{---} \text{pentagon}_{\bar{U}} \text{---} \square_C \approx \square_C \text{---} \quad (5.40)$$

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.,

$$\text{Diagram: } \square_{C'} \text{---} \circlearrowleft_A = \square_{U'} \text{---} \square_{P'} \quad (5.41)$$

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

5.B New algorithm for “biorthogonalizing” two uMPS

We now describe how to “biorthogonalize” two uMPSs (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 uMPSs 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:

$$\text{Diagram: } \square_{C_{LU}} \text{---} \circlearrowleft_{A_U} \approx \text{pentagon}_{P_L} \text{---} \square_{C_{LU}} \quad (5.42)$$

$$\text{Diagram: } \square_{C_{DL}} \text{---} \circlearrowleft_{A_D} \approx \text{pentagon}_{P_L^-} \text{---} \square_{C_{DL}} \quad (5.43)$$

where P_L and P_L^- are tensors satisfying

$$\text{Diagram: } \text{pentagon}_{P_L} \text{---} \text{pentagon}_{P_L^-} \approx \text{vertical line} \quad (5.44)$$

If $A_D^s = (A_U^s)^\dagger$ (possibly after fixing a gauge degree of freedom), one can use the approach introduced in Appendix 5.A, or use the iTEBD algorithm from Ref. [128]. 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. (5.42)–(5.43). 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 :

$$(5.45)$$

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:

$$(5.46)$$

$$(5.47)$$

Here, $C_{LU}^+ = V_L \Sigma_L^+$ and $C_{DL}^+ = \Sigma_L^+ U_L^\dagger$.

This procedure can be viewed as one possible fixed point formulation of biorthogonalization procedure introduced by Huang [123–125] in the context of non-Hermitian transfer matrix DMRG (TMRG) and originally applied to CTMRG by Corboz et al. [108].

In practice, we find that P_L, P_L^- from Eq. (5.42)–(5.43) may not satisfy Eq. (5.44) 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. (5.42)–(5.43) above:

$$(5.48)$$

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

This is only one possible choice for biorthogonalizing two uMPS, and more details about alternative methods are discussed in Ref. [124].

Note that we found that, once the tensors P_L, P_L^- were obtained, an alternative way to obtain $C_{LU/DL}$ is with the following fixed point equations:

$$\text{Diagram (5.49)} \quad \propto \quad \text{Diagram (5.49)}$$

$$\text{Diagram (5.50)} \quad \propto \quad \text{Diagram (5.50)}$$

which may give better matrices $C_{LU/DL}$ for the purpose of FPCM.

Notice that in the limit when $A_D^s = (A_U^s)^\dagger$, this biorthogonalization procedure reduces to an algorithm similar to the iTEBD algorithm from Ref. [128], if that algorithm is used to just gauge a uMPS isometrically from the left.

Additionally, we will describe an alternative biorthogonalization method that we have tested with the asymmetric FPCM. One could first gauge the uMPS A_U and A_D isometrically from the left (for example using the method described in Appendix 5.A) 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. (5.46)–(5.47). 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 uMPSs (for example in the Heisenberg model example in Section 5.4.3).

BIBLIOGRAPHY

- ¹M. T. Fishman and S. R. White, “Compression of correlation matrices and an efficient method for forming matrix product states of fermionic gaussian states”, [Phys. Rev. B **92**, 075132 \(2015\)](#).
- ²V. Zauner-Stauber, L. Vanderstraeten, M. T. Fishman, F. Verstraete, and J. Haegeman, “Variational optimization algorithms for uniform matrix product states”, [Phys. Rev. B **97**, 045145 \(2018\)](#).
- ³M. T. Fishman, L. Vanderstraeten, V. Zauner-Stauber, J. Haegeman, and F. Verstraete, “Faster Methods for Contracting Infinite 2D Tensor Networks”, [arXiv:1711.05881](#).
- ⁴J. Haegeman and F. Verstraete, “Diagonalizing transfer matrices and matrix product operators: a medley of exact and computational methods”, [Annual Review of Condensed Matter Physics **8**, 355–406 \(2017\)](#).
- ⁵S. R. White, “Density matrix formulation for quantum renormalization groups”, [Phys. Rev. Lett. **69**, 2863–2866 \(1992\)](#).
- ⁶S. R. White, “Density-matrix algorithms for quantum renormalization groups”, [Phys. Rev. B **48**, 10345–10356 \(1993\)](#).
- ⁷M. C. Gutzwiller, “Effect of correlation on the ferromagnetism of transition metals”, [Phys. Rev. Lett. **10**, 159–162 \(1963\)](#).
- ⁸S. Zhang, J. Carlson, and J. E. Gubernatis, “Constrained path quantum monte carlo method for fermion ground states”, [Phys. Rev. Lett. **74**, 3652–3655 \(1995\)](#).
- ⁹S. Zhang, J. Carlson, and J. E. Gubernatis, “Constrained path monte carlo method for fermion ground states”, [Phys. Rev. B **55**, 7464–7477 \(1997\)](#).
- ¹⁰G. Evenbly and G. Vidal, “Entanglement renormalization in noninteracting fermionic systems”, [Phys. Rev. B **81**, 235102 \(2010\)](#).
- ¹¹C. V. Kraus, N. Schuch, F. Verstraete, and J. I. Cirac, “Fermionic projected entangled pair states”, [Phys. Rev. A **81**, 052338 \(2010\)](#).
- ¹²J. I. Latorre and A. Riera, “A short review on entanglement in quantum spin systems”, [Journal of Physics A: Mathematical and Theoretical **42**, 504002 \(2009\)](#).
- ¹³I. Peschel and V. Eisler, “Reduced density matrices and entanglement entropy in free lattice models”, [Journal of Physics A: Mathematical and Theoretical **42**, 504003 \(2009\)](#).
- ¹⁴G. Vidal, J. I. Latorre, E. Rico, and A. Kitaev, “Entanglement in quantum critical phenomena”, [Phys. Rev. Lett. **90**, 227902 \(2003\)](#).
- ¹⁵J. I. Latorre, E. Rico, and G. Vidal, “Ground state entanglement in quantum spin chains”, [Quantum Info. Comput. **4**, 48–92 \(2004\)](#).

- ¹⁶G. Vidal, “Entanglement renormalization”, *Phys. Rev. Lett.* **99**, 220405 (2007).
- ¹⁷I. Daubechies, “Orthonormal bases of compactly supported wavelets”, *Communications on Pure and Applied Mathematics* **41**, 909–996 (1988).
- ¹⁸I. Daubechies et al., *Ten lectures on wavelets*, Vol. 61 (SIAM, 1992).
- ¹⁹X.-L. Qi, “Exact holographic mapping and emergent space-time geometry”, [arXiv:1309.6282](https://arxiv.org/abs/1309.6282).
- ²⁰G. Vidal, “Efficient simulation of one-dimensional quantum many-body systems”, *Phys. Rev. Lett.* **93**, 040502 (2004).
- ²¹S. R. White and A. E. Feiguin, “Real-time evolution using the density matrix renormalization group”, *Phys. Rev. Lett.* **93**, 076401 (2004).
- ²²L. G. Valiant, “Quantum circuits that can be simulated classically in polynomial time”, *SIAM Journal on Computing* **31**, 1229–1254 (2002).
- ²³R. Jozsa and A. Miyake, “Matchgates and classical simulation of quantum circuits”, *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* **464**, 3089–3106 (2008).
- ²⁴W. P. Su, J. R. Schrieffer, and A. J. Heeger, “Solitons in polyacetylene”, *Phys. Rev. Lett.* **42**, 1698–1701 (1979).
- ²⁵G. Evenbly and G. Vidal, “Entanglement renormalization in free bosonic systems: real-space versus momentum-space renormalization group transforms”, *New Journal of Physics* **12**, 025007 (2010).
- ²⁶K. G. Wilson, “The renormalization group: Critical phenomena and the Kondo problem”, *Rev. Mod. Phys.* **47**, 773 (1975).
- ²⁷M. E. Fisher, “The renormalization group in the theory of critical behavior”, *Rev. Mod. Phys.* **46**, 597 (1974).
- ²⁸M. E. Fisher, “Renormalization group theory: Its basis and formulation in statistical physics”, *Rev. Mod. Phys.* **70**, 653 (1998).
- ²⁹J. Zinn-Justin, *Quantum Field Theory and Critical Phenomena* (Oxford University Press, 1996).
- ³⁰M. Fannes, B. Nachtergaele, and R. F. Werner, “Finitely correlated states on quantum spin chains”, *Communications in Mathematical Physics* **144**, 443–490 (1992).
- ³¹A. Klümper, A. Schadschneider, and J. Zittartz, “Matrix Product Ground States for One-Dimensional Spin-1 Quantum Antiferromagnets”, *Europhys. Lett.* **24**, 293 (1993).
- ³²S. Östlund and S. Rommer, “Thermodynamic limit of density matrix renormalization”, *Phys. Rev. Lett.* **75**, 3537–3540 (1995).

- ³³D. Perez-Garcia, F. Verstraete, M. M. Wolf, and J. I. Cirac, “Matrix Product State Representations”, *Quant. Inf. Comput.* **75**, 401 (2007).
- ³⁴F. Verstraete, V. Murg, and J. I. Cirac, “Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems”, *Adv. Phys.* **57**, 143 (2008).
- ³⁵U. Schollwöck, “The density-matrix renormalization group in the age of matrix product states”, *Ann. Phys.* **326**, 96 (2011).
- ³⁶J. Haegeman, T. J. Osborne, and F. Verstraete, “Post-matrix product state methods: To tangent space and beyond”, *Phys. Rev. B* **88**, 075133 (2013).
- ³⁷R. Orús, “A practical introduction to tensor networks: Matrix product states and projected entangled pair states”, *Ann. Phys.* **349**, 117 (2014).
- ³⁸J. Bridgeman and C. T. Chubb, “Hand-waving and Interpretive Dance: An Introductory Course on Tensor Networks”, (2016), [arXiv:1603.03039](https://arxiv.org/abs/1603.03039).
- ³⁹M. Hastings, “An area law for one-dimensional quantum systems”, *J. Stat. Mech. Theor. Exp.* **2007**, P08024 (2007).
- ⁴⁰J. Eisert, M. Cramer, and M. B. Plenio, “Colloquium: Area laws for the entanglement entropy”, *Rev. Mod. Phys.* **82**, 277 (2010).
- ⁴¹F. Verstraete and J. I. Cirac, “Matrix product states represent ground states faithfully”, *Phys. Rev. B* **73**, 094423 (2006).
- ⁴²M. B. Hastings, “Entropy and entanglement in quantum ground states”, *Phys. Rev. B* **76**, 035114 (2007).
- ⁴³J. Haegeman, S. Michalakis, B. Nachtergaele, T. J. Osborne, N. Schuch, and F. Verstraete, “Elementary Excitations in Gapped Quantum Spin Systems”, *Phys. Rev. Lett.* **111**, 080401 (2013).
- ⁴⁴F. Verstraete, D. Porras, and J. I. Cirac, “Density Matrix Renormalization Group and Periodic Boundary Conditions: A Quantum Information Perspective”, *Phys. Rev. Lett.* **93**, 227205 (2004).
- ⁴⁵S. R. White, “Density matrix renormalization group algorithms with a single center site”, *Phys. Rev. B* **72**, 180403 (2005).
- ⁴⁶I. P. McCulloch, “From density-matrix renormalization group to matrix product states”, *J. Stat. Mech.*, P10014 (2007).
- ⁴⁷I. McCulloch, “Infinite size density matrix renormalization group, revisited”, [arXiv:0804.2509](https://arxiv.org/abs/0804.2509).
- ⁴⁸G. Vidal, “Efficient classical simulation of slightly entangled quantum computations”, *Phys. Rev. Lett.* **91**, 147902 (2003).
- ⁴⁹G. Vidal, “Classical simulation of infinite-size quantum lattice systems in one spatial dimension”, *Phys. Rev. Lett.* **98**, 070201 (2007).

- ⁵⁰J. Haegeman, J. I. Cirac, T. J. Osborne, I. Pi žorn, H. Verschelde, and F. Verstraete, “Time-dependent variational principle for quantum lattices”, *Phys. Rev. Lett.* **107**, 070601 (2011).
- ⁵¹J. Haegeman, C. Lubich, I. Oseledets, B. Vandereycken, and F. Verstraete, “Unifying time evolution and optimization with matrix product states”, *Phys. Rev. B* **94**, 165116 (2016).
- ⁵²R. Bhatia, *Matrix Analysis* (Springer-Verlag, New York, 1997).
- ⁵³H. N. Phien, I. P. McCulloch, and G. Vidal, “Fast convergence of imaginary time evolution tensor network algorithms by recycling the environment”, *Phys. Rev. B* **91**, 115137 (2015).
- ⁵⁴B. Pirvu, V. Murg, J. I. Cirac, and F. Verstraete, “Matrix product operator representations”, *New J. Phys.* **12**, 025012 (2010).
- ⁵⁵M. P. Zaletel, R. S. K. Mong, C. Karrasch, J. E. Moore, and F. Pollmann, “Time-evolving a matrix product state with long-ranged interactions”, *Phys. Rev. B* **91**, 165112 (2015).
- ⁵⁶R. J. Baxter, “Dimers on a rectangular lattice”, *Journal of Mathematical Physics* **9**, 650–654 (1968).
- ⁵⁷R. J. Baxter, “Variational approximations for square lattice models in statistical mechanics”, *Journal of Statistical Physics* **19**, 461–478 (1978).
- ⁵⁸R. J. Baxter, *Exactly solved models in statistical mechanics* (Academic Press, London, 1982) Chap. 13.
- ⁵⁹H. A. Kramers and G. H. Wannier, “Statistics of the two-dimensional ferromagnet. part ii”, *Phys. Rev.* **60**, 263–276 (1941).
- ⁶⁰S. B. Kelland, “Estimates of the critical exponent β for the potts model using a variational approximation”, *Canadian Journal of Physics* **54**, 1621–1626 (1976).
- ⁶¹S. K. Tsang, “Square lattice variational approximations applied to the ising model”, *Journal of Statistical Physics* **20**, 95–114 (1979).
- ⁶²R. J. Baxter and I. G. Enting, “Series expansions from corner transfer matrices: the square lattice ising model”, *Journal of Statistical Physics* **21**, 103–123 (1979).
- ⁶³R. J. Baxter, “Hard hexagons: exact solution”, *Journal of Physics A: Mathematical and General* **13**, L61 (1980).
- ⁶⁴R. Baxter, “Corner transfer matrices”, *Physica A: Statistical Mechanics and its Applications* **106**, 18–27 (1981).
- ⁶⁵A. Klümper, A. Schadschneider, and J. Zittartz, “Matrix product ground states for one-dimensional spin-1 quantum antiferromagnets”, *EPL (Europhysics Letters)* **24**, 293 (1993).
- ⁶⁶H. F. Trotter, “On the product of semi-groups of operators”, *Proceedings of the American Mathematical Society* **10**, 545–551 (1959).

- ⁶⁷M. Suzuki, “Generalized trotter’s formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems”, [Communications in Mathematical Physics](#) **51**, 183–190 (1976).
- ⁶⁸M. Suzuki, “Transfer-matrix method and monte carlo simulation in quantum spin systems”, [Phys. Rev. B](#) **31**, 2957–2965 (1985).
- ⁶⁹M. Suzuki, “Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations”, [Physics Letters A](#) **146**, 319–323 (1990).
- ⁷⁰T. Nishino, “Density matrix renormalization group method for 2d classical models”, [Journal of the Physical Society of Japan](#) **64**, 3598–3601 (1995).
- ⁷¹T. Nishino and K. Okunishi, “Corner transfer matrix renormalization group method”, [Journal of the Physical Society of Japan](#) **65**, 891–894 (1996).
- ⁷²T. Nishino and K. Okunishi, “Corner transfer matrix algorithm for classical renormalization group”, [Journal of the Physical Society of Japan](#) **66**, 3040–3047 (1997).
- ⁷³T. Nishino and K. Okunishi, “Numerical latent heat observation of the $q = 5$ potts model”, [Journal of the Physical Society of Japan](#) **67**, 1492–1493 (1998).
- ⁷⁴A. Kemper, A. Gendiar, T. Nishino, A. Schadschneider, and J. Zittartz, “Stochastic light-cone ctmrg: a new dmrg approach to stochastic models”, [Journal of Physics A: Mathematical and General](#) **36**, 29 (2003).
- ⁷⁵D. P. Foster and C. Pinettes, “Corner-transfer-matrix renormalization-group method for two-dimensional self-avoiding walks and other $o(n)$ models”, [Phys. Rev. E](#) **67**, 045105 (2003).
- ⁷⁶D. P. Foster and C. Pinettes, “A corner transfer matrix renormalization group investigation of the vertex-interacting self-avoiding walk model”, [Journal of Physics A: Mathematical and General](#) **36**, 10279 (2003).
- ⁷⁷K. Ueda, R. Otani, Y. Nishio, A. Gendiar, and T. Nishino, “Snapshot observation for 2d classical lattice models by corner transfer matrix renormalization group”, [Journal of the Physical Society of Japan](#) **74**, 111–114 (2005).
- ⁷⁸K. Ueda, R. Krmar, A. Gendiar, and T. Nishino, “Corner transfer matrix renormalization group method applied to the ising model on the hyperbolic plane”, [Journal of the Physical Society of Japan](#) **76**, 084004 (2007).
- ⁷⁹R. Krmar, A. Gendiar, K. Ueda, and T. Nishino, “Ising model on a hyperbolic lattice studied by the corner transfer matrix renormalization group method”, [Journal of Physics A: Mathematical and Theoretical](#) **41**, 125001 (2008).
- ⁸⁰E. Bartel and A. Schadschneider, “Quantum corner — transfer matrix dmrg”, [International Journal of Modern Physics C](#) **19**, 1145–1161 (2008).

- ⁸¹Y.-b. Chan, “Series expansions from the corner transfer matrix renormalization group method: the hard-squares model”, *Journal of Physics A: Mathematical and Theoretical* **45**, 085001 (2012).
- ⁸²R. Orús, “Exploring corner transfer matrices and corner tensors for the classical simulation of quantum lattice systems”, *Phys. Rev. B* **85**, 205117 (2012).
- ⁸³Y. B. Chan and A. Reznitz, “Accurate lower bounds on 2-d constraint capacities from corner transfer matrices”, *IEEE Transactions on Information Theory* **60**, 3845–3858 (2014).
- ⁸⁴C.-Y. Huang, T.-C. Wei, and R. Orús, “Holographic encoding of universality in corner spectra”, *Phys. Rev. B* **95**, 195170 (2017).
- ⁸⁵T. Nishino and K. Okunishi, “A density matrix algorithm for 3d classical models”, *Journal of the Physical Society of Japan* **67**, 3066–3072 (1998).
- ⁸⁶K. Okunishi and T. Nishino, “Kramers-wannier approximation for the 3d ising model”, *Progress of Theoretical Physics* **103**, 541–548 (2000).
- ⁸⁷T. Nishino, K. Okunishi, Y. Hieida, N. Maeshima, and Y. Akutsu, “Self-consistent tensor product variational approximation for 3d classical models”, *Nuclear Physics B* **575**, 504–512 (2000).
- ⁸⁸T. Nishino, Y. Hieida, K. Okunishi, N. Maeshima, Y. Akutsu, and A. Gendiar, “Two-dimensional tensor product variational formulation”, *Progress of Theoretical Physics* **105**, 409–417 (2001).
- ⁸⁹A. Gendiar and T. Nishino, “Latent heat calculation of the three-dimensional $q = 3, 4$, and 5 potts models by the tensor product variational approach”, *Phys. Rev. E* **65**, 046702 (2002).
- ⁹⁰A. Gendiar, N. Maeshima, and T. Nishino, “Stable optimization of a tensor product variational state”, *Progress of Theoretical Physics* **110**, 691–699 (2003).
- ⁹¹N. Maeshima, Y. Hieida, Y. Akutsu, T. Nishino, and K. Okunishi, “Vertical density matrix algorithm: a higher-dimensional numerical renormalization scheme based on the tensor product state ansatz”, *Phys. Rev. E* **64**, 016705 (2001).
- ⁹²Y. Nishio, N. Maeshima, A. Gendiar, and T. Nishino, “Tensor Product Variational Formulation for Quantum Systems”, [arXiv:cond-mat/0401115](https://arxiv.org/abs/cond-mat/0401115).
- ⁹³F. Verstraete and J. J. I. Cirac, “Renormalization algorithms for Quantum-Many Body Systems in two and higher dimensions”, [arXiv:cond-mat/0407066](https://arxiv.org/abs/cond-mat/0407066).
- ⁹⁴J. Jordan, R. Orús, G. Vidal, F. Verstraete, and J. I. Cirac, “Classical simulation of infinite-size quantum lattice systems in two spatial dimensions”, *Phys. Rev. Lett.* **101**, 250602 (2008).
- ⁹⁵R. Orús and G. Vidal, “Simulation of two-dimensional quantum systems on an infinite lattice revisited: corner transfer matrix for tensor contraction”, *Phys. Rev. B* **80**, 094403 (2009).

- ⁹⁶B. Bauer, G. Vidal, and M. Troyer, “Assessing the accuracy of projected entangled-pair states on infinite lattices”, [Journal of Statistical Mechanics: Theory and Experiment](#) **2009**, P09006 (2009).
- ⁹⁷P. Corboz, R. Orús, B. Bauer, and G. Vidal, “Simulation of strongly correlated fermions in two spatial dimensions with fermionic projected entangled-pair states”, [Phys. Rev. B](#) **81**, 165104 (2010).
- ⁹⁸P. Corboz, J. Jordan, and G. Vidal, “Simulation of fermionic lattice models in two dimensions with projected entangled-pair states: next-nearest neighbor hamiltonians”, [Phys. Rev. B](#) **82**, 245119 (2010).
- ⁹⁹P. Corboz, S. R. White, G. Vidal, and M. Troyer, “Stripes in the two-dimensional t - j model with infinite projected entangled-pair states”, [Phys. Rev. B](#) **84**, 041108 (2011).
- ¹⁰⁰P. Corboz, A. M. Läuchli, K. Penc, M. Troyer, and F. Mila, “Simultaneous dimerization and $su(4)$ symmetry breaking of 4-color fermions on the square lattice”, [Phys. Rev. Lett.](#) **107**, 215301 (2011).
- ¹⁰¹B. Bauer, P. Corboz, A. M. Läuchli, L. Messio, K. Penc, M. Troyer, and F. Mila, “Three-sublattice order in the $su(3)$ heisenberg model on the square and triangular lattice”, [Phys. Rev. B](#) **85**, 125116 (2012).
- ¹⁰²P. Czarnik, L. Cincio, and J. Dziarmaga, “Projected entangled pair states at finite temperature: imaginary time evolution with ancillas”, [Phys. Rev. B](#) **86**, 245101 (2012).
- ¹⁰³P. Czarnik and J. Dziarmaga, “Fermionic projected entangled pair states at finite temperature”, [Phys. Rev. B](#) **90**, 035144 (2014).
- ¹⁰⁴P. Corboz and F. Mila, “Tensor network study of the shastry-sutherland model in zero magnetic field”, [Phys. Rev. B](#) **87**, 115144 (2013).
- ¹⁰⁵R. Orús, “A practical introduction to tensor networks: matrix product states and projected entangled pair states”, [Annals of Physics](#) **349**, 117–158 (2014).
- ¹⁰⁶J. Osorio Iregui, P. Corboz, and M. Troyer, “Probing the stability of the spin-liquid phases in the kitaev-heisenberg model using tensor network algorithms”, [Phys. Rev. B](#) **90**, 195102 (2014).
- ¹⁰⁷D. Poilblanc, P. Corboz, N. Schuch, and J. I. Cirac, “Resonating-valence-bond superconductors with fermionic projected entangled pair states”, [Phys. Rev. B](#) **89**, 241106 (2014).
- ¹⁰⁸P. Corboz, T. M. Rice, and M. Troyer, “Competing states in the t - j model: uniform d -wave state versus stripe state”, [Phys. Rev. Lett.](#) **113**, 046402 (2014).
- ¹⁰⁹P. Corboz and F. Mila, “Crystals of bound states in the magnetization plateaus of the shastry-sutherland model”, [Phys. Rev. Lett.](#) **112**, 147203 (2014).
- ¹¹⁰L. Wang, P. Corboz, and M. Troyer, “Fermionic quantum critical point of spinless fermions on a honeycomb lattice”, [New Journal of Physics](#) **16**, 103008 (2014).

- ¹¹¹P. Czarnik and J. Dziarmaga, “Projected entangled pair states at finite temperature: iterative self-consistent bond renormalization for exact imaginary time evolution”, *Phys. Rev. B* **92**, 035120 (2015).
- ¹¹²P. Czarnik and J. Dziarmaga, “Variational approach to projected entangled pair states at finite temperature”, *Phys. Rev. B* **92**, 035152 (2015).
- ¹¹³H. N. Phien, J. A. Bengua, H. D. Tuan, P. Corboz, and R. Orús, “Infinite projected entangled pair states algorithm improved: fast full update and gauge fixing”, *Phys. Rev. B* **92**, 035142 (2015).
- ¹¹⁴P. Corboz, “Improved energy extrapolation with infinite projected entangled-pair states applied to the two-dimensional hubbard model”, *Phys. Rev. B* **93**, 045116 (2016).
- ¹¹⁵P. Corboz, “Variational optimization with infinite projected entangled-pair states”, *Phys. Rev. B* **94**, 035133 (2016).
- ¹¹⁶J. Osorio Iregui, M. Troyer, and P. Corboz, “Infinite matrix product states versus infinite projected entangled-pair states on the cylinder: a comparative study”, *Phys. Rev. B* **96**, 115113 (2017).
- ¹¹⁷I. Niesen and P. Corboz, “A tensor network study of the complete ground state phase diagram of the spin-1 bilinear-biquadratic Heisenberg model on the square lattice”, *SciPost Phys.* **3**, 030 (2017).
- ¹¹⁸I. Niesen and P. Corboz, “Emergent haldane phase in the $s = 1$ bilinear-biquadratic heisenberg model on the square lattice”, *Phys. Rev. B* **95**, 180404 (2017).
- ¹¹⁹B.-X. Zheng, C.-M. Chung, P. Corboz, G. Ehlers, M.-P. Qin, R. M. Noack, H. Shi, S. R. White, S. Zhang, and G. K.-L. Chan, “Stripe order in the underdoped region of the two-dimensional hubbard model”, *Science* **358**, 1155–1160 (2017).
- ¹²⁰L. Vanderstraeten, M. Mariën, F. Verstraete, and J. Haegeman, “Excitations and the tangent space of projected entangled-pair states”, *Phys. Rev. B* **92**, 201111 (2015).
- ¹²¹L. Vanderstraeten, J. Haegeman, P. Corboz, and F. Verstraete, “Gradient methods for variational optimization of projected entangled-pair states”, *Phys. Rev. B* **94**, 155123 (2016).
- ¹²²P. Corboz, M. Lajkó, A. M. Läuchli, K. Penc, and F. Mila, “Spin-orbital quantum liquid on the honeycomb lattice”, *Phys. Rev. X* **2**, 041013 (2012).
- ¹²³Y.-K. Huang, “Biorthonormal transfer-matrix renormalization-group method for non-hermitian matrices”, *Phys. Rev. E* **83**, 036702 (2011).
- ¹²⁴Y.-K. Huang, “Biorthonormal matrix-product-state analysis for the non-hermitian transfer-matrix renormalization group in the thermodynamic limit”, *Journal of Statistical Mechanics: Theory and Experiment* **2011**, P07003 (2011).

- ¹²⁵Y.-K. Huang, P. Chen, and Y.-J. Kao, “Accurate computation of low-temperature thermodynamics for quantum spin chains”, *Phys. Rev. B* **86**, 235102 (2012).
- ¹²⁶P. Nataf, M. Lajkó, P. Corboz, A. M. Läuchli, K. Penc, and F. Mila, “Plaquette order in the $su(6)$ heisenberg model on the honeycomb lattice”, *Phys. Rev. B* **93**, 201113 (2016).
- ¹²⁷L. Onsager, “Crystal statistics. i. a two-dimensional model with an order-disorder transition”, *Phys. Rev.* **65**, 117–149 (1944).
- ¹²⁸R. Orús and G. Vidal, “Infinite time-evolving block decimation algorithm beyond unitary evolution”, *Phys. Rev. B* **78**, 155117 (2008).
- ¹²⁹R. J. Bursill, T. Xiang, and G. A. Gehring, “The density matrix renormalization group for a quantum spin chain at non-zero temperature”, *Journal of Physics: Condensed Matter* **8**, L583 (1996).
- ¹³⁰X. Wang and T. Xiang, “Transfer-matrix density-matrix renormalization-group theory for thermodynamics of one-dimensional quantum systems”, *Phys. Rev. B* **56**, 5061–5064 (1997).
- ¹³¹N. Shibata, “Thermodynamics of the anisotropic heisenberg chain calculated by the density matrix renormalization group method”, *Journal of the Physical Society of Japan* **66**, 2221–2223 (1997).
- ¹³²A. Kemper, A. Schadschneider, and J. Zittartz, “Transfer-matrix density-matrix renormalization group for stochastic models: the domany-kinzel cellular automaton”, *Journal of Physics A: Mathematical and General* **34**, L279 (2001).
- ¹³³T. Enss and U. Schollwöck, “On the choice of the density matrix in the stochastic tmrg”, *Journal of Physics A: Mathematical and General* **34**, 7769 (2001).
- ¹³⁴U. Schollwöck, “The density-matrix renormalization group”, *Rev. Mod. Phys.* **77**, 259–315 (2005).
- ¹³⁵M. Suzuki, “General theory of fractal path integrals with applications to many-body theories and statistical physics”, *Journal of Mathematical Physics* **32**, 400–407 (1991).
- ¹³⁶M. Levin and C. P. Nave, “Tensor renormalization group approach to two-dimensional classical lattice models”, *Phys. Rev. Lett.* **99**, 120601 (2007).
- ¹³⁷Z. Y. Xie, H. C. Jiang, Q. N. Chen, Z. Y. Weng, and T. Xiang, “Second renormalization of tensor-network states”, *Phys. Rev. Lett.* **103**, 160601 (2009).
- ¹³⁸Z. Y. Xie, J. Chen, M. P. Qin, J. W. Zhu, L. P. Yang, and T. Xiang, “Coarse-graining renormalization by higher-order singular value decomposition”, *Phys. Rev. B* **86**, 045139 (2012).
- ¹³⁹G. Evenbly and G. Vidal, “Tensor network renormalization”, *Phys. Rev. Lett.* **115**, 180405 (2015).

- ¹⁴⁰S.-J. Ran, “Ab initio optimization principle for the ground states of translationally invariant strongly correlated quantum lattice models”, *Phys. Rev. E* **93**, 053310 (2016).
- ¹⁴¹S. Yang, Z.-C. Gu, and X.-G. Wen, “Loop optimization for tensor network renormalization”, *Phys. Rev. Lett.* **118**, 110504 (2017).
- ¹⁴²M. Bal, M. Mariën, J. Haegeman, and F. Verstraete, “Renormalization group flows of hamiltonians using tensor networks”, *Phys. Rev. Lett.* **118**, 250602 (2017).
- ¹⁴³J. Jordan, R. Orús, and G. Vidal, “Numerical study of the hard-core bose-hubbard model on an infinite square lattice”, *Phys. Rev. B* **79**, 174515 (2009).
- ¹⁴⁴L. Michel and I. P. McCulloch, “Schur Forms of Matrix Product Operators in the Infinite Limit”, (2010), [arXiv:1008.4667](https://arxiv.org/abs/1008.4667).
- ¹⁴⁵G. K.-L. Chan and T. V. Voorhis, “Density-matrix renormalization-group algorithms with nonorthogonal orbitals and non-hermitian operators, and applications to polyenes”, *The Journal of Chemical Physics* **122**, 204101 (2005).
- ¹⁴⁶V. Zauner, D. Draxler, L. Vanderstraeten, M. Degroote, J. Haegeman, M. Rams, V. Stojevic, N. Schuch, and F. Verstraete, “Transfer matrices and excitations with matrix product states”, *New J. Phys.* **17**, 053002 (2015).
- ¹⁴⁷J. Towns, T. Cockerill, M. Dahan, I. Foster, K. Gaither, A. Grimshaw, V. Hazelwood, S. Lathrop, D. Lifka, G. D. Peterson, R. Roskies, J. R. Scott, and N. Wilkins-Diehr, “Xsede: accelerating scientific discovery”, *Computing in Science & Engineering* **16**, 62–74 (2014).
- ¹⁴⁸L. Tagliacozzo, T. R. de Oliveira, S. Iblisdir, and J. I. Latorre, “Scaling of entanglement support for matrix product states”, *Phys. Rev. B* **78**, 024410 (2008).
- ¹⁴⁹F. Pollmann, S. Mukerjee, A. M. Turner, and J. E. Moore, “Theory of Finite-Entanglement Scaling at One-Dimensional Quantum Critical Points”, *Phys. Rev. Lett.* **102**, 255701 (2009).
- ¹⁵⁰B. Pirvu, G. Vidal, F. Verstraete, and L. Tagliacozzo, “Matrix product states for critical spin chains: finite-size versus finite-entanglement scaling”, *Physical review b* **86**, 075117 (2012).
- ¹⁵¹V. Stojevic, J. Haegeman, I. P. McCulloch, L. Tagliacozzo, and F. Verstraete, “Conformal data from finite entanglement scaling”, *Phys. Rev. B* **91**, 035120 (2015).
- ¹⁵²S. G. Chung, “Essential finite-size effect in the two-dimensional xy model”, *Phys. Rev. B* **60**, 11761–11764 (1999).
- ¹⁵³Y. Liu, Y. Meurice, M. P. Qin, J. Unmuth-Yockey, T. Xiang, Z. Y. Xie, J. F. Yu, and H. Zou, “Exact blocking formulas for spin and gauge models”, *Phys. Rev. D* **88**, 056005 (2013).

- ¹⁵⁴J. F. Yu, Z. Y. Xie, Y. Meurice, Y. Liu, A. Denblyker, H. Zou, M. P. Qin, J. Chen, and T. Xiang, “Tensor renormalization group study of classical xy model on the square lattice”, *Phys. Rev. E* **89**, 013308 (2014).
- ¹⁵⁵A. W. Sandvik, “Finite-size scaling of the ground-state parameters of the two-dimensional heisenberg model”, *Phys. Rev. B* **56**, 11678–11690 (1997).
- ¹⁵⁶D. Poilblanc, J. I. Cirac, and N. Schuch, “Chiral topological spin liquids with projected entangled pair states”, *Phys. Rev. B* **91**, 224431 (2015).
- ¹⁵⁷D. Poilblanc, N. Schuch, and I. Affleck, “ $SU(2)_1$ chiral edge modes of a critical spin liquid”, *Phys. Rev. B* **93**, 174414 (2016).