QUANTUM MANY-BODY PHYSICS OF ULTRACOLD MOLECULES IN OPTICAL LATTICES: MODELS AND SIMULATION METHODS

by

Michael L. Wall

A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Doctor of Philosophy (Physics).

Golden, Colorado

Date ________________________

Signed: ________________________
Michael L. Wall

Signed: ________________________
Dr. Lincoln D. Carr
Thesis Advisor

Golden, Colorado

Date ________________________

Signed: ________________________
Dr. Tom Furtak
Professor and Head
Department of Physics

ABSTRACT

Ultracold atoms have revolutionized the field of quantum many-body physics due to excellent understanding of their microscopic dynamics and a high degree of control over these dynamics with external fields. The next revolution in ultracold physics promises to come with ultracold molecules, whose production lies at the cutting edge of research. In this thesis we are concerned with how ultracold molecules trapped in optical lattices may be used as resources for novel many-body physics.

There are six main parts to this thesis. The first part gives a general introduction to the topics covered in the thesis, and provides some technical details on the derivation of many-body lattice models from few-body physics. In the second part, we derive a low-energy Hamiltonian, the molecular Hubbard Hamiltonian, describing ultracold heteronuclear bialkali dimer molecules loaded into an optical lattice and elucidate its many-body properties. These molecules have large permanent electric dipole moments which give rise to long range and anisotropic dipole-dipole interactions in an electric field and allow access to the rich internal structure of rotational and hyperfine states in an AC microwave field. Rather than focusing on simulating models relevant to condensed matter physics, we focus on the many-body physics available to near-term experimental setups with the minimal tuning of external fields.

The third part of this thesis studies how fermions pair to make bosons in a discrete context, via a two-channel model for a Feshbach resonance in the presence of an optical lattice and possibly strong interchannel coupling. The two-body problem is solved numerically using a scaling theory to extract the result for an infinite number of Bloch bands. The bound states of a partitioned Hamiltonian, which we call dressed molecules, are identified as the relevant short-range degrees of freedom at low density, and are chosen so as to reproduce the two-body scattering length identically. From

this two-body solution we derive a low-energy many-body Hamiltonian which takes the form of a multichannel resonance model between unpaired fermions in the lowest Bloch band and dressed molecules. This approach is valid for arbitrary two-body scattering length and resonance width, and is systematically correctible to higher relative scattering energy.

In the fourth part of this thesis we discuss matrix product states (MPSs), a class of entanglement-restricted states which are useful for variational calculations in one spatial dimension. This part begins with an overview of the theory of MPSs, with special emphasis on intuitive notions of their use as variational ansätze. Algorithms for finding eigenstates of 1D Hamiltonians, for time-evolution under a general time-dependent 1D Hamiltonian, and for equilibrium properties at finite temperature are presented. Furthermore, it is shown how MPS algorithms may be made generic by the identification of a class of operators known as a matrix product operators and a set of rules for constructing such operators.

The fifth part of this thesis deals with open source implementations of variational MPS algorithms and educational materials designed to facilitate the use and understanding of these methods. The open source projects include a stand-alone open source implementation of the time-evolving block decimation algorithm, open source TEBD, and an implementation of time-evolving block decimation for the widely used algorithms and libraries for physics simulations (ALPS) package, as part of the ALPS international collaboration. In the final part of the thesis, we conclude, give suggestions for future work, and provide appendices.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

Molecular state with total electronic spin and total projection of electronic orbital angular momentum along the internuclear axis equal to zero.	$^1\Sigma$
Dynamical polarizability tensor	$\tilde{\alpha}$
DC electric field	$\mathbf{E}_{\mathrm{DC}}$
AC electric field	$\mathbf{E}_{\mathrm{AC}}$
Optical lattice field	$\mathbf{E}_{\mathrm{opt}}$
Unnormalized spherical harmonic of order k and projection q	$C_q^{(k)}$
Permanent dipole moment (Part II). Local dimension (Part IV)	d
Projection of dipole operator along spherical direction q	$\hat{d}_q$
Rotational constant	B_N (B in Chapter 3)
Rotational angular momentum operator	$\hat{\mathbf{N}}$ ($\hat{\mathbf{J}}$ in Chapter 3)
Hyperfine spin operator of the i^{th} nucleus	$\hat{\mathbf{I}}_i$
Lattice spacing	a
Characteristic lattice momentum π/a	k_l
Recoil energy	E_R
Number of lattice sites	L
3D Bloch function	$\psi_{\sigma\mathbf{n}\mathbf{q}}$
1D Bloch function	$\psi_{\sigma n q}$
3D Wannier function	$w_{\sigma\mathbf{n}}\left(\mathbf{r}-\mathbf{r}_i\right)$
s-wave scattering length	a_s

Effective range	r_B
Interchannel coupling constant	g (Part III)
Interchannel detuning	ν (Part III)
Total quasimomentum	$\mathbf{K}$
von Neumann entropy of entanglement in region A	$S_{\mathrm{vN}}(A)$
Kronecker product of two indices a and b	(ab)
Bond dimension of a matrix product object	χ

LIST OF ABBREVIATIONS

1D/2D/3D	One dimension/two dimensions/three dimensions
AKLT	Affleck, Lieb, Kennedy, and Tasaki
ALPS	Algorithms and libraries for physics simulations
BCS	Bardeen-Cooper-Schrieffer (theory of superconductivity)
BEC	Bose-Einstein condensate
BHH	Bose-Hubbard Hamiltonian
BZ	Brillouin zone
CDW	Charge-density wave
CFME	Commutator-free Magnus expansion
CPS	Classical product state
CTPRG	Carr theoretical physics research group
DFT	Density functional theory
DMFT	Dynamical mean-field theory
DMRG	Density-matrix renormalization group
ED	Exact diagonalization
FFT	Fast Fourier transform algorithm
FLOP	Floating-point operation
FRH	Fermi resonance Hamiltonian
FSA	Finite state automaton
GAXPY	Generalized A times X plus Y

GMRES	Generalized minimal residual algorithm
HMHH	Hyperfine molecular Hubbard Hamiltonian
iMPS	Infinite size variational matrix product state algorithm
MCQDT	Multichannel quantum defect theory
MERA	Multiscale entanglement renormalization ansatz
METTS	Minimally entangled typical thermal states
MHH	Molecular Hubbard Hamiltonian
MI	Mott insulator
MPI	Message passing interface
MPO(s)	Matrix product operator(s)
MPS(s)	Matrix product state(s)
OA	Operator alphabet
OBC	Open boundary conditions
OpenMP	Open multiprocessing
OSTEBD	Open source time-evolving block decimation
PBC	Periodic boundary conditions
PEPS	Projected entangled-pair states
QMC	Quantum Monte Carlo
RG	Renormalization group
SF	Superfluid
SPDM	Single particle density matrix
STIRAP	Stimulated Raman adiabatic passage
SVD	Singular value decomposition

TEBD	Time-evolving block decimation
ZGESVD	Complex double precision singular value decomposition

ACKNOWLEDGMENTS

First I would like to thank my advisor, Lincoln Carr. Lincoln's enthusiasm for research and his dedication to his students were what drew me to the Colorado school of Mines in the first place. During the course of my Ph.D. he has given me every opportunity I could have wished for. Additionally, Lincoln has been very supportive of my family life over the progress of my degree, and for that I am immensely grateful.

I would also like to thank my committee, John Bohn, Mahadevan Ganesh, James McNeil, and Zhigang Wu, for their encouragement and for reading a long thesis.

Thanks are due to the members of the Algorithms and Libraries for Physics Simulations (ALPS) collaboration for maintaining great coding and pedagogical resources and allowing me to be a part of the collaboration. Special thanks go to Matthias Troyer, who was very helpful and patient as I grew accustomed to being part of a large code collaboration and learned Python in the process.

I am grateful to Dr. Tim Kaiser for managing the high performance computing facilities here at Mines and for his advice on high performance computing applications.

Thanks to the many people who contributed to what would become open source TEBD, including Charles Clark, Barry Schneider, Jamie Williams, Ippei Danshita, and Daniel Schirmer. I have also benefitted from feedback on the open source projects and their associated educational materials from many people, including Giuseppe Carleo, Meng Cheng, Stephen Clark, Juan Manuel Florez, Lukas Gamper, Joseph Glick, Martin Heimsoth, Alexander Itin, Mateusz Łacki, Hong Lu, Grégoire Misguich, Dominik Muth, Jen Patterson, Han Pu, Bryce Robbins, and Joe Whalen.

Hans Peter Büchler was very kind to discuss his work on strongly interacting fermions in optical lattices and share his computer code with me. His generosity, as well as the initial inspiration from his work, were crucial to bringing the Fermi

resonance Hamiltonian to fruition.

I would like to thank J. Ignacio Cirac and his group for their hospitality during a two week stay in Garching in October 2009. Special thanks go to Mari Carmen Bañuls for her assistance in scheduling and for arranging a pleasant evening out in Munich. Also special thanks to Juan José García Ripoll for sharing some Matlab-based MPS code with me during my visit.

I am grateful to Erich Hoover for maintaining a very nice LaTeX thesis template, and for modifying the template to suit some specific needs of the present thesis.

I have been fortunate to attend many great workshops and conferences during my tenure as a graduate student. I would like to thank the Institute for Pure and Applied Mathematics at the University of California, Los Angeles and the organizers for the opportunity to attend the program "Numerical Approaches to Quantum Many-Body Systems"; the organizers of the 2010 Boulder summer school for condensed matter and materials physics for allowing me to participate in a wonderful and very insightful summer school; the organizers of the physics of quantum electronics conference for their invitation to speak in 2012; the organizers of the first quantum technologies conference for a nice conference and a pleasant week in Toruń; and Ana Maria Rey for an invitation to speak about the Fermi resonance Hamiltonian at her group meeting.

I have benefitted from discussions with many researchers about about molecules and matrix product states, including Jesus Aldegunde, Jeremy Hutson, Roman Krems, Salvatore Manmana, Ian McCulloch, Ulrich Schollwöck, Miles Stoudenmire, Frank Verstraete, and Steve White. I would especially like to acknowledge Silke Ospelkaus and the groups of Jun Ye and Debbie Jin at the University of Colorado, Boulder for showing me around their experimental labs.

I would like to thank all of the members of the Carr theoretical physics research group over the years for their friendship and insightful discussions. Special thanks are due to Laith Haddad for his humor, kindness, and empathy as we finished our

degrees simultaneously; to Scott Strong for the good times and for watching the cat while I was away; and to Erman Bekaroglu for his careful readings of my manuscripts and discussions about molecules. I would also like to thank Ryan Mishmash for his friendship, for sharing his knowledge of condensed matter physics, and for always giving advice with his characteristic acerbic wit.

I would not be where I am today without the support and love of my parents, Gail and Rick, and my brother, John. Finally, my deepest thanks to my wife, Teresa, and our two little girls, Maren and Juni, for their love, patience, caring support, and perspective.

To the memory of my grandfathers, Richard V. Walker and Richard H. Wall Sr.

Test everything; retain what is good. 1 Thessalonians 5:21

PART I

INTRODUCTION

CHAPTER 1

GENERAL INTRODUCTION

According to the postulates of quantum mechanics, a quantum system is completely specified by the state $|\psi(t)\rangle$ whose evolution is provided by the Schrödinger equation

$$i\hbar\frac{\partial}{\partial t}|\psi(t)\rangle = \hat{H}|\psi(t)\rangle\,. \tag{1.1}$$

Here, $\hat{H}$ is the Hamiltonian describing the interactions of all of the microscopic degrees of freedom in the system under study. Unfortunately, the dimension of the Hilbert space in which the state $|\psi(t)\rangle$ lives grows exponentially with the number of constituents in a many-body system, rendering Eq. (1.1) essentially useless for extracting physically relevant information from systems with more than a few particles. Practical concerns aside, there is a more fundamental reason why Eq. (1.1) does not enable us to answer all relevant questions in many-body physics. This reason is put succinctly by P. W. Anderson in his now famous article "More is different" [1] when he says that "The ability to reduce everything to simple fundamental laws does not imply the ability to start from those laws and reconstruct the universe." That is to say, many-body systems can display very different, *emergent*, behavior from their microscopic constituents. In particular, the ground state of a many-body system need not have the same symmetry as its governing Hamiltonian due to the phenomenon of *spontaneous symmetry breaking*.

A powerful method for studying weakly-interacting[1] many-body systems is provided by the principle of *adiabatic continuity* [2] which allows the eigenstates of the

---

[1] The precise sense in which we mean the system is weakly interacting is that it is not strongly correlated, the latter of which will be defined below. In particular, by weakly interacting we do not mean that the interactions lie within the radius of convergence of a perturbation series.

interacting system to be connected to the eigenstates of the non-interacting system. The key assumption of this notion is that levels do not change symmetry as interactions are introduced, and so levels corresponding to the same symmetry have no crossings with increasing interactions. Hence, the ordering of quantum numbers in the interacting and non-interacting systems will be roughly the same. The most prominent many-body theory resulting from adiabatic continuity is Landau's Fermi liquid theory, which applies to weakly interacting fermions in three dimensions. Here, the elementary excitations of the interacting system are quasiparticles which stand in one-to-one correspondence with those of a free Fermi gas, albeit with renormalized physical parameters [3]. Adiabatic continuity clearly fails if the system changes its symmetry, as it does near a point of non-analyticity known as a *quantum phase transition* (QPT) [4]. We define the class of systems which cannot be adiabatically connected to the non-interacting counterparts of their microscopic degrees of freedom as being *strongly correlated*.

The degrees of freedom which are relevant to the low-energy theory of a strongly correlated system may be difficult to identify, as we have no reference non-interacting state to which they may be related. An example of this is the fractional quantum Hall effect, in which the relevant degrees of freedom are quasiparticles which carry rational fractions of the elementary charge [5] and also obey fractional exchange statistics [6, 7]. The identification of the relevant macroscopic degrees of freedom in a many-body system is provided, at least in principle, by the *renormalization group* (RG) procedure [8]. An RG analysis involves studying the behavior of a system under a scaling transformation in which some set of degrees of freedom are integrated out to yield an effective description of the system in fewer variables. This procedure is called *coarse graining*. The main classical focus of RG was in studying critical systems. Critical systems have no length scales due to the diverging of the correlation length ξ describing exponential decay of equal-time order parameter correlations in the ground state.[2] Thus, provided that we have coarse grained to a scale large compared to the microscopic scales, further coarse graining should not produce any significant effect on our description of the system. That is, critical points represent fixed points of the RG iteration. More generally, the fixed points of the RG iteration correspond to the possible macroscopic states of the system. Here the remarkable feature of *universality* naturally arises, in which microscopic details of the system are not relevant to its macroscopic behavior.

However powerful the RG idea, one still must choose an appropriate coarse graining procedure. A particularly powerful procedure for one-dimensional (1D) systems is *White's rule* [9], which posits that the states which should be kept when coarse graining a system in real space are those which have the largest weight in the reduced density matrix obtained by tracing out all degrees of freedom not being coarse grained. The real-space RG procedure utilizing White's rule is formulated algorithmically as the *density-matrix renormalization group* (DMRG), which has been the method of choice for strongly correlated 1D systems for nearly 20 years. Theoretical analysis of DMRG has revealed [10, 11] that it may be formulated as a variational method within a class of quantum states known as *matrix product states* (MPSs). This realization together with insights from quantum information theory, particularly from entanglement theory, led to a generalization of the DMRG procedure to time evolution for short-range interacting systems [12–14] and to the proper entanglement structure to describe states with periodic boundary conditions efficiently via DMRG [15]. Also, research along these lines has led to proposals of real-space RG schemes based on entanglement decimation in higher dimensions such as projected entangled-pair states (PEPS)[16, 17] and the multiscale entanglement renormalization algorithm (MERA) [18]; this is still an area of intense research activity.

---

[2] In the absence of exponential decay, the correlation length may be taken to be a length scale on which correlations qualitatively shift to a long-distance behavior.

Using MPSs as a variational ansatz builds upon a long history of variational methods for strongly correlated systems. Many of the most successful methods in many-body theory, for example the Bardeen-Cooper-Schrieffer (BCS) theory of superconductivity [19], the theory of the quantum Hall effect [5], and density functional theory (DFT) [20, 21], are all variational in nature. The availability of open source packages for DFT, such as SIESTA [22], ABINIT [23], and OCTOPUS [24] have allowed for great progress in materials physics. MPSs and their generalizations provide the best hope of stimulating such progress for strongly correlated systems. Hence, one of the goals of this thesis is to provide flexible MPS algorithms which are applicable to a wide array of systems, and open source implementations of MPS algorithms which are simultaneously high performance and easily modified to meet user's needs.

In addition to the theoretical hurdles to studying strongly correlated systems, it is difficult in a condensed matter setting to predict which systems will be strongly correlated. Even when a strongly correlated system has been discovered, often its parameters are difficult to control, and its microscopic dynamics may be too fast to be reliably studied. The crossover of atomic physics into the field of strongly correlated condensed matter began circa 1995 when advances in laser cooling [25] and evaporative cooling [26, 27] led to the creation of Bose-Einstein condensates (BEC) of the alkali metal species Rb [28], Na [29], and Li [30, 31]. As opposed to liquid Helium, the only other known elemental quantum liquid at the time, the interactions in these highly dilute gases were weak, enabling more detailed analysis both theoretically and experimentally. Following the success of bosonic atoms, fermionic atoms were also brought to quantum degeneracy [32–34]. While Pauli exclusion prevents fermions in the same internal state from interacting through s-wave interactions and hence drastically slows evaporative cooling at low temperatures [35], Fermi gases may be cooled either through sympathetic cooling of the gas with a BEC or through evaporative cooling when multiple species of fermions are present.

Ultracold[3] atomic gases have many advantages over traditional condensed matter systems for studying many-body physics. For one, the timescales which are relevant to an atomic many-body system are on the order of milliseconds to minutes, several orders of magnitude longer than typical timescales of condensed matter experiments. This has enabled for the study of slow many-body dynamics of a quantum phase transition [36] and the collapse and revival of matter wave coherence [37]. Additionally, ultracold atoms are extremely well isolated from their environment, and sources of decoherence such as spontaneous emission events can be controlled such that lifetimes are on the order of seconds to more than a minute. Finally, using tools such as optically or magnetically tunable Feshbach resonances [38] allows for precise tunability of the interactions in ultracold gases. We will discuss Feshbach resonances in more detail in Sec. 1.1.

Another essential tool for strongly correlated many-body physics with ultracold atoms is provided by *optical lattices*. An optical lattice [39] is a standing wave array of light formed by counter-propagating laser beams in three dimensions. The light couples to the dynamical polarizability of the object, and the resulting AC Stark shift induces a periodic trapping potential. By altering the geometry or phase of the beams forming the lattice, one can induce a wide variety of geometries [40], including confinement to a quasi-1D geometry [41, 42], as well as time-dependence [43]. Such a lattice mimics the effects of the lattices common in solid state systems, but without the difficulties provided by disorder and phonons which are inevitable in solid state systems at finite temperature. A major avenue of research stemming from this capability is in tuning the parameters of an atomic gas trapped in an optical lattice such that the governing Hamiltonian reproduces a model relevant to condensed matter physics. Such specialized experimental systems represent *quantum simulators* [44], which are essentially single-program quantum computers first envisioned by Feyn-

---

[3] By ultracold, we mean temperatures less than 1 microKelvin.

man [45]. The first quantum simulator both to be proposed [46] and built [36] was the Bose-Hubbard model [47]

$$\hat{H} = -t\sum_{\langle i,j\rangle}\left[\hat{b}_i^{\dagger}\hat{b}_j + \mathrm{h.c.}\right] + \frac{U}{2}\sum_i \hat{n}_i\left(\hat{n}_i - 1\right)\, . \tag{1.2}$$

Here i and j label sites in a lattice, $\langle i,j\rangle$ denotes all pairs which are nearest-neighbors in the lattice, $\hat{b}_i$ destroys a bosonic particle at site i, and $\hat{n}_i = \hat{b}_i^{\dagger}\hat{b}_i$ counts the number of bosonic particles on site i. The first term represents quantum mechanical tunneling of bosons between neighboring sites with an associated tunneling energy t, and the latter term is an energetic penalty for two or more bosons to occupy the same site due to interactions. While it is remarkable that ultracold atoms are able to realize a model of great relevance to condensed matter physics, Eq. (1.2) is also important in that it represents a minimal, natural Hamiltonian for ultracold atoms in an optical lattice. That is, if one were to load an ultracold bosonic alkali gas into an optical lattice without fine tuning of fields, Eq. (1.2) would govern its properties. A topic which forms much of the bulk of this thesis is determining the corresponding natural Hamiltonian for molecules, which have more a complex internal structure than atoms. We will discuss the microscopic derivation of Hamiltonians such as Eq. (1.2) in further detail in Chapter 2.

The remainder of the introduction is organized as follows. In Sec. 1.1 we review Feshbach resonances, an indispensable experimental tool for tuning the interactions of ultracold atomic gases and the production of ultracold molecules. In Sec. 1.2 we discuss experimental production of ultracold molecules. In Sec. 1.3 we provide a basic review of the structure and few-body properties of the molecules relevant to this thesis. In Sec. 1.4 we present a digest of modern numerical techniques for the quantum many-body problem, with a focus on methods applicable to strongly correlated systems. In Sec. 1.5 we overview work by other groups which is related to the results of this thesis. Finally, in Sec. 1.6, we outline the layout of the body of the thesis.

1.1 Feshbach Resonances

Feshbach resonances [38, 48] are an essential component both of the tunability of interactions in ultracold atomic gases and of the production of ultracold molecules. The basic physics of a Feshbach resonance can be explained through a two-channel model. In this model, we partition our Hilbert space into open and (energetically) closed channels, with the asymptotic limit of the open channel potential corresponding to two free atoms. The closed channel potential is assumed to support a bound molecular state near the threshold of the open channel potential. In the presence of a phenomenological interchannel coupling g, the bound state of the closed channel is no longer a true bound state but becomes a resonance due to its mixing with the open channel.[4] Now, a Feshbach resonance occurs when one of the bound states in the closed channel becomes near degenerate with the scattering state in the open channel. Even a weak coupling g is sufficient to cause strong mixing of the two channels when these energies nearly coincide, and this causes a drastic change in the scattering properties. Expressed in terms of the detuning ν between the open and closed channels, the two-channel scatting amplitude is [49]

$$f\left(\mathbf{k}\right) = -\frac{2\mu}{4\pi\hbar^2}\frac{g^2}{\epsilon_{\mathbf{k}} - \nu + \frac{\mu g^2}{2\pi\hbar^2}ik}\,, \tag{1.3}$$

where μ is the reduced mass, $\mathbf{k}$ the incident momentum, and $\epsilon_{\mathbf{k}}$ the incident energy. We may write this scattering amplitude as

$$f\left(\mathbf{k}\right) = -\frac{1}{1/a_s + ik + r_b k^2}\,, \tag{1.4}$$

---

[4] This is what differentiates a Feshbach resonance from a shape resonance. In the latter no bound state exists in the absence of the coupling.

by identifying the s-wave scattering length $a_s = -2\mu g^2/4\pi\hbar^2\nu$ and the effective range $r_B = \pi\hbar^4/\mu^2 g^2$, both of which are experimentally measurable. Although this is the asymptotic form of the scattering amplitude arising from scattering of low-momentum particles with $k\,|r_B| \ll 1$[50], this form is form is exact for the two-channel model.[5] Hence, at higher relative energy the two-channel model breaks down in its ability to describe the full momentum dependence of the scattering amplitude.

Feshbach resonances in ultracold gases are most frequently provided by hyperfine couplings between atoms whose valence electrons reside in singlet and triplet configurations.[6] The singlet potential is generally much deeper than the triplet potential, and also generally appears above the threshold of the triplet potential. Hence, the singlet forms the energetically closed channel. Because of the difference in magnetic moment between the singlet and triplet states, the energetic difference between their scattering thresholds may be tuned with a magnetic field. Denoting the detuning ν between a bound state of the closed channel and the scattering threshold of the open channel in terms of the difference in magnetic moment $\delta\mu \equiv \mu_{\text{closed}} - \mu_{\text{open}}$, the magnetic field strength B, and the critical magnetic field strength B_c where the detuning vanishes,

$$\nu\,(B) = \delta\mu\,(B - B_c)\;, \tag{1.5}$$

we may parameterize the dependence of the s-wave scattering length on B as [38]

$$a_s\,(B) = a_{\text{bg}}\left(1 - \frac{\Delta}{B - B_0}\right)\,. \tag{1.6}$$

Here, $\Delta = \Gamma_0/\delta\mu$ with Γ_0 defining the strength of the resonance, $B_0 = B_c + \delta B$ where $\delta B = \delta E/\delta\mu$ is an interchannel interaction-induced shift, a_{bg} is the background

---

[5] Note that the effective range given in the context of scattering of slow particles, $r_\star$, is related to the effective range defined here as $r_\star = -2r_b$.

[6] The spin states of real atoms are never purely singlet or triplet, but rather singlet-dominated or triplet-dominated.

scattering length the open channel would have in the absence of the closed channel bound state, and we have neglected any inelastic processes.

A common parameterization of the interactions in dilute atomic gases is provided by the regularized s-wave pseudopotential [51, 52]

$$U\left(\mathbf{r}\right)=\frac{2\pi\hbar^2 a_s}{\mu}\delta\left(\mathbf{r}\right)\partial_r r\,. \tag{1.7}$$

This pseudopotential is exact in the low-energy limit $E\to 0$, and provides an excellent approximation provided that $k\left|a_s\right|\ll 1$ and $kr_B\ll 1$ [38]. Hence, the controllability of the scattering length translates quite readily into controllability of the two-body interactions of the gas. We will return to this parameterization in Chapter 2, when we discuss Hubbard models.

1.2 Production of Ultracold Molecules

Molecules have a complex internal structure of vibrational and rotational energy levels which has no counterpart in alkali atoms.[7] Transitions between the internal states of a molecule are not typically governed by strict selection rules as in atoms, but rather by the square modulus of the wave function overlap between the states in question. These overlaps are known as *Franck-Condon factors*. This complicated internal structure makes direct laser cooling of a molecule challenging, as it is very difficult to devise a closed pumping cycle. Several laser cooling schemes have been proposed for specific molecules [53–56], and enhancement of laser cooling has been predicted when molecules are placed within an optical resonator cavity [57–59], with laser-cooled ions [60], or with atoms in Rydberg states [61, 62]. Experimental progress in direct laser cooling of molecules has been made in rare cases, most notably Strontium Fluoride [63]. Other *direct* methods of cooling which start from preformed molecules and attempt to extract energy from them include buffer gas cooling [64] and deceler-

---

[7] An overview of molecular degrees of freedom is provided in Sec. 1.3.

ation [65] by means of electric [66–68], magnetic [69, 70], or optical [71–73] fields. In the buffer gas cooling technique, hot molecules of the desired species are loaded into a chamber containing a cryogenic noble gas, most commonly Helium or Neon, and allowed to equilibrate. The latter technique amounts to essentially running a particle accelerator in reverse. That is, translational energy is extracted by imposing a field gradient on the molecule. The advantage of direct means such as buffer gas cooling or deceleration is that they are applicable to a wide array of species. However, direct methods to date have only produced molecules in the cold regime of temperatures $T \sim$10mK-1K which does not allow access to the fully quantum degenerate regime.

The most successful methods for producing ultracold, high-density samples of molecules have been *indirect* methods which form molecules from atoms which have themselves been cooled to ultracold temperatures. Because of the large effort in cooling alkali atoms [28–30], most of the molecules formed in this fashion are bialkali molecules. Early attempts at the indirect production of molecules focused on *photo-associating* two atoms scattering in S states into a bound state of the S+P excited potential [74, 75]. Molecules can transition to the ground (electronic) state by spontaneous decay [76, 77]. However, the most successful method for creating ultracold molecular samples in very deeply bound levels is by *magneto-association* of ultracold atoms into bound *Feshbach molecules* [38, 78] by sweeping across a Feshbach resonance. Generally, the magneto-association process creates molecules which are weakly bound but translationally ultracold, and the processes has been optimized so as to have nearly unit efficiency. Magneto-association has been achieved for heteronuclear species [79–86] as well as two-component fermionic species [87–93] and single-component bosons [94, 95]. In the latter two cases, these molecules have been observed to condense and form a molecular BEC. This tunability enables the study [91–93, 96–98] of the crossover from a BEC of diatomic molecules to a Cooper paired dilute Fermi gas, a phenomenon known as the *BEC-BCS crossover* [99–103].

Part III of this thesis focuses on solving the crossover problem in an optical lattice at the two-particle level using numerically exact techniques, and then using this numerical solution to derive an effective many-body model for the crossover in the lattice. Although the continuum limit of the BEC-BCS crossover problem has been solved, the lattice problem has been the subject of considerable debate. The low-energy effective theory describing a two-channel model in the lattice takes the form of a multi-channel model between unpaired atoms in the lowest band and a set of *dressed molecules* which form an effective closed channel. However, as with the relevant degrees of freedom of the other strongly correlated systems mentioned in the introductory paragraphs, the effective closed channel bears little resemblance to the microscopic closed channel, instead consisting of an infinite summation over Bloch bands from both the open and closed microscopic channels.

While the molecules which are produced through Feshbach association are translationally ultracold, they are often very highly internally excited. For ultracold atoms, the presence of excitations in the internal degrees of freedom do not cause concern, as these degrees of freedom are either hyperfine states which can be manipulated via optical pumping or electronic excitations which are so large in energy as to be effectively frozen out. For molecules, the presence of a rich internal structure of rotational and vibrational levels, together with the absence of strict selection rules for transitions, makes the isolation of a single molecular state a more daunting task. Remarkable progress has been made in transferring a collection of Feshbach molecules to a low-lying internal state via the stimulated Raman adiabatic passage (STIRAP) procedure [82, 86, 95, 104–107]. In this procedure, Feshbach molecules are transferred coherently to a much more deeply bound state via a two-photon process chosen such that the Franck-Condon factors of the intermediate state with both the target deeply bound state and the Feshbach molecular state are large. This process has been demonstrated to have 90% one-way efficiency in the case of KRb [106]. Finally, by using the mixing of hyperfine and rotational states induced by a nuclear quadrupole coupling, one can transfer the molecules into the lowest hyperfine state at a given magnetic field, resulting in absolute ground state molecules [108].

1.3 Classifications and Few-Body Physics of Ultracold Molecules

STIRAP has been the most successful method for producing ultracold, high phase-space density samples of molecules. Because STIRAP requires that the constituent atoms are already ultracold before they are assembled into molecules, ultracold molecules to date have been mostly alkali metal dimers [82, 85, 86, 109, 110], with mixed alkali metal-alkaline earth molecules on the horizon. The natural energy scales governing the various degrees of freedom of an alkali metal dimer molecule span nearly twelve orders of magnitude, and can be classified as in Figure 1.1. The largest energy scales are set by the electronic degrees of freedom and are of order a few electron-volts. The electronic degrees of freedom are involved in the coupling of the molecule to the optical lattice via the polarizability tensor [111], but these fields are far detuned from any resonances so that the molecule remains in its electronic ground state. The molecular term symbol denoting the electronic state is of the form $Q^{2S+1}\Lambda^{\pm}$, where S is the total electronic spin, Λ is the absolute value of the projection of the total orbital angular momentum along the internuclear axis, and ± denotes the parity under reflections in a plane containing the internuclear axis [50]. The term $Q = X$ for the ground electronic potential and then $A, B, C, \dots$ in energetically ascending order for excited electronic potentials of the same S, Λ multiplicity as the ground state and $a, b, c, \dots$ for excited electronic potentials of different multiplicity. When convenient, we will leave off Q to discuss the multiplet structure of a particular level without specifying how it is ordered with respect to the others. In this thesis we focus on molecules with ${}^1\Sigma$ ground states.

The next largest degrees of freedom are the vibrational degrees of freedom, which are classified by a quantum number v. The spacings between vibrational energy levels are of order a few THz, and can be neglected at ultracold temperatures. The next relevant degrees of freedom are the rotational modes of the molecule which have spacings on the order of a GHz. The rotational modes will be of particular interest for us because they are the energetically lowest-lying dipole-accessible excitations of $^1\Sigma$ molecules. We denote the operator of rotational angular momentum as $\hat{\mathbf{N}}$, with eigenkets $\hat{\mathbf{N}}^2|NM_N\rangle = N(N+1)|NM_N\rangle$. Here M_N is the projection of $\mathbf{N}$ along the space-fixed z-axis. In the presence of a DC electric field N is no longer a good quantum number, but the eigenstates are adiabatically connected to states in zero field. Hence, we use the notation $\bar{N}$ to represent the eigenstate in a DC field which is adiabatically connected to rotational state N in zero field. Finally, at the bottom of the energy hierarchy are the nuclear spins given by the operators $\hat{\mathbf{I}}_1$ and $\hat{\mathbf{I}}_2$ with projections M_1 and M_2 along the space-fixed z axis. Here, 1 and 2 refer to the constituents of the molecule in the order that the molecule is named, e. g. Rb is 1 and Cs 2 for RbCs. The interactions governing the nuclear spins are of order 100Hz-1kHz. In terms of temperature, 1kHz corresponds to roughly 50 nanoKelvin, and so hyperfine structure is typically thermally populated while the other internal structure is not. While these energies are small compared to the other scales of the problem, the precise, state-selective nature of the STIRAP process requires us to take into account the hyperfine structure of the molecules [108].

Explicitly, a $^1\Sigma$ alkali dimer molecule may be characterized by the parameters displayed in Table 1.1 [108, 111–113]. The numerical values of these parameters for the three most experimentally relevant species at the time of the writing of this thesis [82, 109, 110] are also collected in Table 1.1.[8] The physical origins of these terms and their role in the microscopic Hamiltonian are given in Chapters 3 and 4.

---

[8] The dynamic polarizability of LiCs has not yet been calculated, to our knowledge. We estimate that the ratio of the perpendicular and parallel polarizabilities will be similar to that of KRb and RbCs.

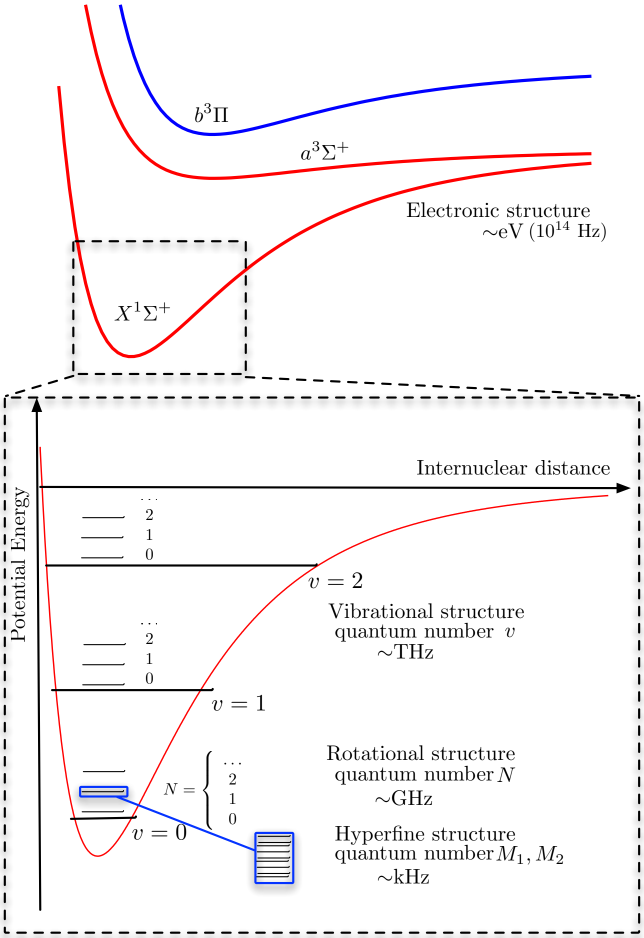


Figure 1.1: Intrinsic energy scales of alkali metal dimers. In descending order, the energy scales of a $^1\Sigma$ diatomic molecule are excitations of the electronic state, vibrational modes of the nuclei about the equilibrium internuclear separation, rotation of the molecule about its center of mass, and couplings between the nuclear spins and the other angular momenta of the molecule. These energy scales span nearly twelve orders of magnitude.

Table 1.1: Coupling constants defining a $^1\Sigma$ molecular species. Most of the values in this table have been obtained by DFT or other ab-initio theoretical means [111–113] and have not yet been verified by experiment. The exceptions are the rotational constant, dipole moment, and nuclear quadrupole couplings for KRb [108].

Description	Symbol	^{87}Rb^{133}Cs	^{40}K^{87}Rb	^{7}Li^{133}Cs
Rotational constant	B_N(GHz)	0.488	1.1139	5.636
Permanent dipole moment	d(Debye)	1.25	0.566	5.52
Nuclear spin 1	I_1	3/2	4	3/2
Nuclear spin 2	I_2	7/2	3/2	7/2
Nuclear g-factor 1	g_1	1.834	-0.324	2.171
Nuclear g-factor 2	g_2	0.738	1.834	0.738
Nuclear quadrupole coupling 1	$(eqQ)_1$(MHz)	-0.872	0.45	0.0185
Nuclear quadrupole coupling 2	$(eqQ)_2$(MHz)	0.051	-1.41	0.188
Rotation-nuclear spin coupling 1	c_1(Hz)	98.4	-24.1	32
Rotation-nuclear spin coupling 2	c_2(Hz)	194.1	420.1	3014
Tensor nuclear spin-spin coupling	c_3(Hz)	192.4	-48.2	140
Scalar nuclear spin-spin coupling	c_4(Hz)	17345.4	-2030.4	1610
Rotational g-factor	g_r	0.0062	0.014	0.0106
Nuclear shielding factor 1	σ_1(ppm)	3531	1321	108.2
Nuclear shielding factor 2	σ_2(ppm)	6367	3469	6242.5
Parallel polarizability	$\alpha_\parallel$(au)	3033.97215	2116.77398	?
Perpendicular Polarizability	$\alpha_\perp$(au)	675.962926	471.613009	$\sim 0.222798\alpha_\parallel$

What are the new features of ultracold molecules relevant to many-body physics? The dipole-dipole interaction

$$\hat{H}_{\mathrm{DD}}\left(\mathbf{R}\right)=\frac{\hat{\mathbf{d}}\cdot\hat{\mathbf{d}}-3\left(\hat{\mathbf{d}}\cdot\mathbf{e}_R\right)\left(\mathbf{e}_R\cdot\hat{\mathbf{d}}\right)}{R^3}\,, \tag{1.8}$$

where $\mathbf{R}$ is the vector connecting the two dipoles, is the source of much interest in polar molecules as it provides interactions which are both long range and anisotropic. The anisotropy has been observed in studying the stereodynamics of ultracold collisions [114, 115]. In addition, because of the anisotropy, angular momentum is not conserved during low-energy scattering. This adds a short-range contribution in the s-wave channel [116–118] which gives rise to a weak shape resonance in all even-ℓ channels. The given form of the dipole-dipole potential breaks down at short distances where dispersion and chemical effects become relevant. We account for this by imposing a short distance cutoff b of order r_{vdW}, the van der Waals length, on the dipole-dipole potential and account for the short-distance behavior by adding a contribution [119, 120]

$$\frac{4\pi\hbar^2 a\left(d\right)}{m}\delta\left(\mathbf{r}\right) \tag{1.9}$$

to the short-range pseudopotential. Here $a\left(d\right)$ is the dipole-dependent scattering length which is provided by the low-energy limit of the scattering matrix. Such a pseudopotential has been shown [119] to reproduce the correct physics away from any shape resonances.

In addition to the dipole-dipole interaction, molecules have a rich internal state space which can be tunably accessed using AC microwave fields. Thus, in addition to the ability to fine-tune fields to produce spin-like models for quantum simulation [121, 122], one can also access regimes where a large number of internal degrees of freedom are interacting over a disparate range of timescales and thus build a simulator of a quantum complex system [123]. The means by which one can tune the internal state space using external fields will be covered in Chapters 3 and 4.

Molecules also display complex short-range physics due both to the large number of internal states and large energy scales at short range [124] and the possibility of chemical reactions [125, 126]. In order to discuss short range physics, we first clarify the scales which classify processes as being either short or long range. We define the dipole length $r_{\mathrm{d}} = (md^2/\hbar^2)$ as the separation where the dipolar energy of two particles is comparable to the relative collision energy of two particles with a wavelength of r_d. We also define the van der Waals length $r_{\mathrm{vdW}} = \left[2\pi/\Gamma\left(1/4\right)^2\right]\left(2\mu C_6/\hbar^2\right)^{1/4}$, where C_6 is the coefficient of the $1/R^6$ dispersion potential with R the intermolecular separation, $\mu = M/2$ the reduced mass of two molecules of mass M, and $\Gamma\left(x\right)$ is the Gamma function, as the separation where the dipolar interaction becomes comparable to the short-range dispersive interaction [38, 127]. The scale at which chemical reactions become relevant is R_e, the bond length, which is smaller than 1nm for the species considered here [128]. The associated energy scale is on the order of 100THz, the chemical bond scale. The van der Waals length r_{vdW} is the next relevant length scale, ranging from 6nm for KRb to 30 and 50nm for LiCs and RbCs, respectively. The dipole length is very large compared to all of these, and increases with increasing dipole moment. The short-range region where chemical and state-changing collisions become relevant corresponds to $R < r_{\mathrm{vdW}}$.

A complete picture of low-energy scattering for these molecules is provided by multichannel quantum defect theory (MCQDT) [129] which uses the large separation of scales to define dimensionless parameters s and y which characterize the short range phase shift and chemical reactivity [130]. LiCs and KRb are both highly reactive [131]; there is unit probability of loss at short range. Thus, the collision rates, both elastic and inelastic, depend only on the long range potential and there can be no scattering resonances. Universal formulae exist for such species [131–133] and agree well with experimental measurements of molecular lifetimes [125]. Also, hyperfine spin appears not to be relevant in the scattering of highly reactive molecules [125, 131]. For non-reactive species such as RbCs and NaK, the collisions are non-universal, and may have electric field dependent scattering resonances [134].

Recent work has shown that the short-range physics, both inelastic and elastic, also depends strongly on an applied electric field [114, 133, 135, 136]. In particular, it has been well established that the stability of a molecular ensemble increases with increasing dipole moment in quasi-2D traps [133, 137]. Somewhat surprisingly, the electric field does not appear to be relevant to the stability of a molecular gas in a 3D optical lattice [126]. The detailed description of the short range scattering is outside the scope of this work. We characterize the short range physics by a complex scattering length a_s describing both elastic and inelastic scattering, which can be computed using MCQDT. We note that large real or imaginary parts of a_s lead to an effective hard-core condition in an optical lattice where only one molecule can exist per lattice site due to strong resonant interactions or the continuous quantum Zeno effect [138], respectively.

1.4 Simulation Methods

Analytical solutions of the many-particle Schrödinger equation are rare, and so often one resorts to numerical techniques. One of the most successful methods for the quantum many-body problem is density functional theory (DFT), which relies on the fact that the ground state energy is a unique functional of the ground state density [20, 21]. However, this energy functional is only known exactly for a free electron gas, and so approximate functionals are used for interacting systems. The computational procedure of DFT consists of minimizing the energy generated from an approximate functional by changing the density. The density corresponding to the minimum energy is expected to most closely represent the true ground state density. While in principle the many-body wave function of the ground state is itself also a functional of the ground state density, this functional is not known, and so the predictive power of DFT is limited to observables which depend only on the density. Hence, DFT is not generically useful for detecting order in strongly correlated systems, which generally requires two-point correlation functions such as the density-density correlation function $\langle \hat{n}_i \hat{n}_j \rangle$. Also, approximate functionals are only available for Coulomb-interacting electronic systems, and so DFT is not an appropriate starting point for systems which interact through different potentials such as molecules interacting through the dipole-dipole potential. Finally, we note that no improvements made to DFT can turn it into a "black box" method applicable to any interacting electronic system, even in principle, as the existence of an efficient approximation to the universal functionals relevant to electronic systems would imply that the hardest problems for quantum computers to solve would be efficiently solvable by classical computers [139].[9] This is thought to be impossible, although no formal proof has yet been provided.

The simplest method to solve the many-body Schrödinger equation is to form a matrix representation of the Hamiltonian operator and numerically diagonalize it. This method is referred to as *exact diagonalization*. The fact that the Hilbert space of a many-body system grows exponentially with the number of constituents in the system implies that this procedure is only practical for systems with very few constituents. Some improvement can be made by considering *sparse eigensolver methods* such as the Lanczos [140] or Davidson [141] algorithms which use only a procedure applying the Hamiltonian to a given vector to find solutions corresponding to extremal eigenvalues. By carefully accounting for all symmetries of the system, for example point group symmetries of the lattice, U(1) symmetry corresponding to conservation of number of particles, or SU(2) spin symmetry, the non-zero values of

---

[9] Explicitly, it would imply that a *Quantum Merlin-Arthur* (QMA)-complete problem lies in P, where P is the class of problems which can be solved in polynomial time by a deterministic Turing machine. QMA is the quantum analog to the classical complexity class NP.

the Hamiltonian can be codified and calculated on-the-fly to optimize performance. Even with all of these optimizations, the cutting edge of modern exact diagonalization is limited to roughly 40 two-component spins, or qubits, or a half-filled electronic system on a square lattice with 20 sites.

Another popular class of methods for the many-body problem are *Quantum Monte Carlo* (QMC) methods. Path-integral QMC methods use the *worldline* mapping from a d dimensional quantum system to a $d+1$ dimensional classical system [4, 142] and then use the classical Metropolis algorithm [143] to generate the equilibrium expectations of observables by sampling worldline configurations. The most common path integral QMC methods for strongly correlated systems are loop QMC [144] and the worm algorithm [145]. Modern implementations of these algorithms can simulate millions of particles at low temperatures, even when strongly correlated. However, path integral QMC suffers from the *sign problem* for interacting fermionic systems in dimensions greater than one or *frustrated* systems which have an extensive classical ground state degeneracy. In these systems, there exist configuration updates of the worldlines which amount to negative probabilities in the classical Monte Carlo scheme, and so Monte Carlo can no longer be applied. If the sign is ignored in the update procedure, an exponentially growing cancellation in the sign expectation leads to an exponential growth of errors rather than a statistically limited behavior of errors as in convergent Monte Carlo [146]. A similar *phase problem* exists when applying QMC to the unitary dynamics of a many-body system. There also exist variational QMC methods such as diffusion Monte Carlo [147] or variational Monte Carlo [148] in which a trial wavefunction is optimized using a Monte Carlo procedure. These algorithms suffer from bias in the choice of the variational ansatz, and also are not generally applicable to dynamics. Diagrammatic Monte Carlo [149] samples terms[10] in a series rather than configuration updates. If the series is a perturbation series, the associated

---

[10] Series expansions often employ a diagrammatic notation for the terms in the series, hence the name diagrammatic Monte Carlo.

diagrams are Feynman diagrams. Bold-line diagrammatic Monte Carlo [150–152] uses re-summation techniques [153] over subsets of diagrams to improve the efficiency, and do not suffer from sign problem in some cases. Such methods may also be applicable to real-time dynamics [154].

A more recent method for strongly correlated systems is *Dynamical mean-field theory* (DMFT) [155–159], in which the full many-body problem is mapped onto an effective impurity problem for one of the constituents with a self-consistently defined bath describing the coupling of this impurity to the surrounding system. The sign problem in the associated impurity problem can be better controlled due to the small size of the impurity, and the mean-field approximation can be assuaged by considering clusters of sites rather than single-site impurities and extrapolating in the size of the cluster. DMFT is most useful in dimensions $D > 2$, and becomes exact in the limit that $D \to \infty$. DMFT requires the analytical mapping from the original many-body problem to the impurity problem, and so has only been formulated for specific interaction potentials such as the those appearing in the Bose-Hubbard [160] and Fermi-Hubbard [156] models or for Bose-Fermi mixtures [159]. Hence, at present, generic applications in DMFT require new development on a case-by-case basis. Both DMFT and most flavors of QMC have the advantage that they incorporate finite temperature naturally.

The simulation methods that we apply for many-body studies in this thesis are variational methods based on matrix product states (MPSs), which are covered in detail in Part IV of this thesis. MPSs are a class of states which are generated through the process of a real-space renormalization group procedure known as the density-matrix renormalization group. The main convergence parameter χ used in an MPS simulation is a cutoff in the entanglement between any two complementary subsections. One of the greatest advantages of MPS methods is that, like exact diagonalization, MPS methods produce wavefunctions, and so a vast array of properties may be computed. Furthermore, MPS methods can be formulated for any microscopic degrees of freedom and for any interactions, making them suitable for generic implementation. Finally, one can devise efficient variational MPS algorithms for the dynamics of systems. The flexibility of MPS algorithms to adapt to different degrees of freedom, a variety of interactions, and simulation of time evolution makes them best suited for our purposes. The main drawback of MPS methods is that they work best in 1D and at zero temperature. Higher dimensional generalizations of MPS algorithms exist, but their numerical conditioning is much worse than that of MPS algorithms at present. Additionally, generalizations to finite temperature exist, but they also scale worse than their zero-temperature counterparts [161–164].

With MPS methods having been used successfully for nearly 20 years, why do molecules necessitate new code and algorithm development? The first reason is that Hubbard models for dipolar molecules have long-range interactions as discussed in Chapter 2. While DMRG-type methods have been devised for long-range interacting systems [163, 165], these methods are often Hamiltonian-specialized and inefficient. A relatively new characterization of Hamiltonians based on *matrix product operators* [166–168] (MPOs) enables for long-range interactions to be accommodated efficiently. Furthermore, as discussed in Chapter 7, by using a small set of finite state automaton rules for the construction of Hamiltonian terms from local operators, a wide array of Hamiltonians can be provided as input in a consistent form to an MPS program. This completely eradicates the need for Hamiltonian-specialized implementation. The need for flexibility in an MPS program is vital for studying molecules; as more complex molecules become cooled to quantum degeneracy a wider array of more complex many-body models are expected. Hence, flexible code prevents the need to "reinvent the wheel" as new many-body models become relevant. The MPS algorithms presented in this thesis make no reference to the microscopic constituents of the model under study or to the range or nature of their interactions. Rather, the only requirements are that the system be quasi one-dimensional and the Hamiltonian be expressed as an MPO.

Long-range interactions allow for more complex translation-breaking orders than finite-range interactions. An example of this is the “devil’s staircase” of insulating phases at rational fillings of bosons interacting with $1/r^3$ interactions [169, 170]. The insulating states can be true long-range ordered crystals in 1D as opposed to e. g. superfluids in 1D which are prevented from long range ordering by the Mermin-Wagner theorem [171, 172]. Broken translational order causes special difficulties for MPS algorithms, as these algorithms are stated most naturally for open boundary conditions. For a system with true long-range translational order, the presence of open boundaries strongly affects the bulk behavior of the system even several hundreds of sites away from the boundaries. To avoid this difficulty, it is necessary to work directly in the limit of an infinite lattice, in which we assume that the many-body state has a periodically repeating unit cell. An algorithm for variationally finding the unit cell of the ground state of an infinite system is provided in Chapter 8.

Finally, even for the simplest diatomic molecules with $^1\Sigma$ ground states the internal state space which is accessible can be significant due to a large hyperfine manifold. A powerful way of breaking a large Hilbert space into its smaller relevant components is through the explicit conservation of symmetries. In Chapter 6 the structure of symmetry-adapted MPSs is elucidated for the simplest case of Abelian symmetries, in which all of the irreducible representations are one-dimensional. Remarkably, the conservation of Abelian symmetries can be implemented in a completely generic way, and an arbitrary number of such symmetries may be simultaneously conserved. The only place in which the particular symmetry group is relevant is in determining how two quantum numbers transform under the group operation. Implementation details are provided in Chapter 6.

1.5 Related Work by Other Groups

The strongest connection of our work on polar molecules to other groups is the dipole-dipole interaction. The most detailed many-body studies of dipole-dipole interactions in ultracold atomic systems have focused on strongly magnetic atoms such as Chromium, Erbium, Europium, and Dysprosium. Chromium was the first of these atoms to be Bose-condensed [173, 174], with Dysprosium [175] and Erbium [176] only having been Bose-condensed very recently. Fermionic isotopes of Dysprosium have also been brought to quantum degeneracy [177]. The dipolar interaction of Chromium is about one-sixth its short range interaction, and so the dipolar effects are generally perturbative in nature. Still, clear signatures of the dipole-dipole interaction have been observed, such as d-wave expansion following the collapse of a Chromium BEC [178]. For broad reviews on the physics of dipolar gases, we refer the reader to Refs. [179, 180].

More closely related to our work are studies of dipolar particles in harmonic traps [119, 120, 181–186] and lattices [170, 187–193]. Most of the work done in harmonic traps is based on the mean-field Gross-Pitaevskii formalism which assumes the presence of a condensate. In the lattice commensurability effects can drive the system out of a condensed phase and into an insulating phase with a periodicity commensurate with the lattice. Thus, a variety of techniques have been brought to bear on the lattice problem. The quantum phases of bosonic dipoles in optical lattices have been investigated for various 2D geometries, including on the square lattice using mean field techniques [189, 191] and via QMC on the square [170] and triangular lattices [192]. A wide variety of quantum phases are possible including checkerboard solid and supersolid phases in addition to the superfluid and Mott insulator phases present for short-range interacting bosons. Supersolids are characterized by coexisting translational and phase order, and have been the subject of intense study [194]. Studies beyond the ground state properties include the characterization and stability properties of metastable states [187, 188]. The presence of a lattice has also been proposed to enhance the production of molecules through indirect means [190]. The scattering properties of a dipolar gas strongly confined in two dimensions but free to move along the third have been studied [183], paralleling investigations of confinement-induced resonances appearing in the short range case [195]. Finally, the parameters of a Luttinger liquid theory have been postulated for general power-law interactions in 1D [169], including an analysis of the Berezinskii-Kosterlitz-Thouless transition to pinned phases in the presence of a weak lattice.

The dipole-dipole interaction is not the only interesting feature of polar molecules. The presence of a permanent electric dipole moment allows for transitions between rotational states in an AC microwave field. An early suggestion based on this observation was provided by Demille [196] who proposed that polar molecules in a one-dimensional trap could be used for quantum computation. Many other groups have proposed using the internal structure to build quantum simulators of spin models [197, 198] or to study other condensed matter phenomena such as the Holstein model [199], exciton physics [200, 201], the physics of liquid crystals [202, 203], and string orders related to the Haldane phase of the antiferromagnetic spin-1 Heisenberg model [204, 205]. The internal structure can also be tuned through external fields in order to produce a desired interaction potential for a specific molecular state in a dressed-state picture [206–208]. More exotic proposals involve loading atoms into self-assembled lattices of dipolar molecules [209, 210] or tuning three-body interactions via external fields [211, 212]. A great deal of recent interest has been garnered for dipolar molecules in bilayer geometries where the dipolar interaction within a layer is repulsive, stabilizing the gases within the layers, but the interaction between molecules in different layers is attractive. This can induce exotic interlayer pairing [213, 214] and induce soliton filaments in a stack of such layers [215].

The most closely related works to our own are those which simulate quantum magnetism, particularly a long-range generalization of the t-J model, using the internal structure of polar molecules [121, 122, 216]. These works have an approach similar to ours, in which the properties of the many-body models are related to those of the few-body physics through microscopic analysis. The main difference between these works and ours is motivation. The t-J work specifically fine-tunes the external fields in order to achieve a quantum simulator of a known model. Our work, on the other hand, begins from a near-term experimental setup and asks what the naturally arising many-body Hamiltonian is and what its many-body features are.

There is also a body of work by other authors focusing on the problem addressed by the Fermi resonance Hamiltonian (FRH) which is the topic of Part III. The FRH maps a resonance model in the continuum onto a resonance model in the lattice by identifying a set of dressed molecules which form the effective closed channel in the lattice. To our knowledge the first time that an effective closed channel was used for a many-body model in a cold atoms context was in Ref. [217], although such terms had been used phenomenologically in the study of high-T_c superconductivity for many years prior [218–220]. The analogous high-T_c model, known as the "cooperon model," is still a subject of current research [221].

A proper description of the physics of the two-channel model requires a renormalization of the theory to remove divergences associated with a point-like boson [222, 223]. The exact solution for two particles interacting via a Feshbach resonance in a (possibly anisotropic) harmonic trap was obtained in Ref. [224]. The importance of intra- as well as inter-band coupling terms was stressed, and the theory was properly renormalized to remove divergences from using a point-like boson. The authors of Ref. [225] consider a lattice two-channel model, and determine the properties of the dressed molecule by considering deep lattices and replacing the lattice with a single harmonic well. The harmonic trap approximation both artificially leads to separability of the center of mass motion from both the relative motion and internal structure and underestimates the extent of Wannier functions, often by an order of magnitude. This implies that qualitative properties of the tunneling, as well as its general order of magnitude, cannot be accounted for using this approach. Duan has derived effective two-channel models using both a projection operator formalism [226] and general symmetry considerations [227]. He considers the dressed molecule to be the exact solution of an on-site Schrödinger equation, and then couples in many-body physics using atom-molecule couplings between neighboring sites. He then considers the case where one of the on-site eigenstates is close to the scattering continuum of two particles in a specific band n, and then projects the Hamiltonian onto the Hilbert space of empty sites, singly occupied sites with a fermion in band n, and doubly occupied sites containing a dressed molecule. Such a Hamiltonian cannot describe the full BEC-BCS crossover, as it restricts the dressed molecules to behave as hard-core bosons whereas deep in the BEC side the molecules are tightly bound, weakly interacting bosons. This work also does not give a prescription for solving the on-site problem, but references the exact solution in the harmonic trap. More recent work [228] uses the numerical solution from a double-well potential to avoid some of the shortcomings of the harmonic oscillator approximation. While this work captures some of the physics of the lattice at the nearest-neighbor level, it does not capture the full quasimomentum dependence of the lattice Hamiltonian.

Büchler was the first to give the exact solution for two fermions interacting through a zero-range Feshbach resonance in an optical lattice, properly accounting for the effects of higher bands and renormalization [229]. He then showed that when the interaction term U of the single-band Hubbard model was determined from the scattering properties of this exact two-body solution self-consistently, the Hubbard model still failed to reproduce the correct physics even for moderate s-wave scattering length, or very far from the actual pole of the Feshbach resonance. This has motivated our approach of using a two-channel model instead of a single-channel model such as the Hubbard model. His discussion of the two-body solution focused on states with zero total quasimomentum, although the theory encompasses states with arbitrary total quasimomentum.

Very recent work by von Stecher *et. al.* [230] focuses on an effective two-channel model near a lattice resonance. Instead of solving the on-site problem exactly they project the two-body Hamiltonian outside of the scattering continuum of two fermions in bands n and m which gives rise to the resonance, solve this projected Hamiltonian exactly, and then use the eigenstates of this projected Hamiltonian as a dressed closed channel. This is approach is very similar in spirit to ours. In contrast to our work, low dimensionality is assumed from the outset and so this approach breaks down when the energy associated with the Feshbach coupling becomes larger than the energy associated with the transverse confinement. Instead, our model treats the population in transverse excited state as being fixed by the two-body solution and thus part of the dressed molecule. This allows any imposed conditions of reduced dimensionality to be controlled only by the transverse tunneling and coupling rates of the dressed molecules.

Finally, with regards to the open source work, an open source version of DMRG exists as a part of the Algorithms and Libraries for Physics Simulations project (ALPS) [231–233]. Open source versions of 1D time dependent DMRG also exist [234, 235]. To the best of our knowledge, the two open source coding projects described in this thesis, open source time-evolving block decimation [236] and the time-evolving block decimation routines included as part of the ALPS package, are the only open source codes which work directly on matrix product states.

1.6 Outline

The present thesis is divided broadly into six parts. Each chapter within a part represents either a publication, in which case the chapter begins with an abstract, or material which places the publications in context and provide background information. Materials relevant to the open source coding projects or educational development discussed in Part V are placed in appendices to avoid interrupting the flow of the thesis. Part I is an introduction to the topics of this thesis, and includes the present chapter. This part continues in Chapter 2, which discusses the general route from a few-body Hamiltonian to a many-body Hamiltonian appropriate for describing strongly interacting particles in a lattice.

Part II of this thesis is devoted to the *molecular Hubbard Hamiltonian*, a model for the low-energy physics of $^1\Sigma$ alkali dimer molecules loaded into an optical lattice. Chapter 3 describes the first attempt at deriving such a Hamiltonian by considering only the rotational degrees of freedom. Special focus was put on describing the dynamics of molecules following a sudden turning on of a microwave AC field coupling the lowest rotational level to higher rotational levels. Using time-dependent matrix product state methods, it was found that coherent Rabi oscillations between two internal states driven at a single-molecule resonance were exponentially damped with an emergent timescale. This effect was termed *quantum dephasing.* As experiments began to produce ultracold molecules in the absolute vibrational, rotational, and hyperfine ground state, this model was revised to include the hyperfine structure of the molecules. We refer to the resulting Hamiltonian as the *hyperfine* molecular Hubbard Hamiltonian, which is discussed in Chapter 4. Chapter 4 studies the strong static electric and magnetic field limit of this Hamiltonian.

In Part III of this thesis we study the problem of the pairing of two-component fermionic atoms via a Feshbach resonance in an optical lattice. As the scattering length for two particles in the open channel diverges near resonance, the pseudopotential model for interactions, amounting to integrating out the closed channel, breaks down. Furthermore, as the strength of the pairing interaction g becomes stronger than the band gap, the restriction of the open channel to the lowest band is no longer appropriate at short distances. Hence, the Hubbard model [237] involving tunneling of two-component fermions in the lowest band with on-site s-wave inter-component interactions fails to correctly describe the system.

In our approach, we project the complete two-channel model into the basis of Bloch functions appropriate to the lattice problem. A scaling analysis enables us to extract the bound state properties of two particles to the limit of an infinite number of bands. By carefully partitioning the Hilbert space into low-energy and high-energy sectors and performing this numerically exact two-particle analysis on the high-energy sector, we derive the *Fermi resonance Hamiltonian* (FRH), a lattice model which is applicable to resonances of any width and any scattering length. The FRH takes the form of a multichannel resonance model between fermions in the lowest band of the open channel and a closed channel consisting of "dressed molecules." The use of the full lattice solution, the extrapolation to an infinite number of bands, and the proper regularization of the Feshbach coupling are all vital to the proper quantitative and qualitative description of the physics.

Part IV of the present thesis is devoted to matrix product states (MPSs) and variational algorithms associated with them. Chapter 6 gives an overview of MPSs. While this chapter is largely conceptual, it also contains details on symmetry adapted MPSs. Chapter 7 provides a thorough overview of algorithms using MPSs as variational ansätze for the eigenstates of arbitrary 1D Hamiltonians on finite lattices. Furthermore, it presents algorithms for simulating the dynamics of arbitrary time-dependent Hamiltonians whose error bounds do not depend on the smoothness properties of the Hamiltonian in question and do not require any Hamiltonian-specialized implementation. Chapter 8 discusses a variational algorithm for finding the ground state of an infinite system variationally using a translationally invariant MPS ansatz. Chapter 9 discusses how algorithms for time evolution of MPSs may be used to simulate MPSs as finite temperature, either through entangling MPSs to a fictitious reservoir or through sampling the characteristic states which arise at finite temperature. This chapter also provides information on how to formulate MPS algorithms for periodic boundary conditions which correctly describe the entanglement structure of periodic states.

In Part V of the thesis we discuss open source coding projects and educational materials. The open source coding projects include open source time-evolving block decimation [236], a stand-alone implementation of time-evolving block decimation, as well as an implementation of time-evolving block decimation as part of the ALPS open source package [231–233]. The educational materials include tutorials on the open source codes which are aimed at the level of graduate students performing research in strongly correlated physics and documents intended for use within the Carr theoretical physics research group for students who may have very little background in quantum mechanics or numerical methods. The manuals and documentation for the open source codes, as well as the educational materials, are reprinted in the appendices.

Finally, the thesis concludes in Chapter 12 with suggestions for future work. This chapter together with the appendices form Part VI of the thesis. All of the code used to generate the numerical data in this thesis, including the most recent versions of the open source codes in both ALPS and stand-alone open source TEBD, are included as a CD with the thesis. The contents of this CD, as well as implementation details for the included codes, are provided in Appendix A.

The publications included in this thesis are

1. Chapter 3. *Emergent Timescales in Entangled Quantum Dynamics of Ultracold Molecules in Optical Lattices*, M. L. Wall and L. D. Carr, New J. Phys. **11** 055027 (2009), doi:10.1088/1367-2630/11/5/055027. Permission is provided by the Creative Commons Attribution-Non-Commercial-ShareAlike 3.0 license according to the New Journal of Physics copyright statement.

2. Chapter 4. *Hyperfine molecular Hubbard Hamiltonian*, M. L. Wall and L. D. Carr, Physical Review A **82**, 013611 (2010), doi:10.1103/PhysRevA.82.013611. Copyright (2010) by the American Physical Society. Permission is provided by the American Physical Society according to the APS copyright policy.

3. Chapter 5. *Microscopic Model for Feshbach Interacting Fermions in an Optical Lattice with Arbitrary Scattering Length and Resonance Width*, M. L. Wall and L. D. Carr, Phys. Rev. Lett. **109**, 055302 (2012), doi:10.1103/PhysRevLett.109.055302. Copyright (2012) by the American Physical Society. Permission is provided by the American Physical Society according to the APS copyright policy.

4. Chapter 7. *Out of equilibrium dynamics with Matrix Product States*, M. L. Wall and L. D. Carr, New J. Physics, under review, arXiv:1205.1020v1. Permission is provided by the Creative Commons Attribution-Non-Commercial-ShareAlike 3.0 license according to the New Journal of Physics copyright statement.

5. Chapter 9. *Finite Temperature Matrix Product State Algorithms and Applications*, M. L. Wall and L. D. Carr, Chapter in ”Quantum Gases: Finite Temperature and Non-Equilibrium Dynamics” (Vol. 1 Cold Atoms Series), N. P. Proukakis, S. A. Gardiner, M. J. Davis and M. H. Szymanska, eds. (Imperial College Press, 2012), arXiv:1008.4303v1. Permission is provided by the Creative Commons Attribution-Non-Commercial-ShareAlike 3.0 license according to the arXiv copyright statement.

6. Material appearing in Appendix C first appeared on the wiki pages for the ALPS collaboration hosted at http://alps.comp-phys.org and was written solely by the author of the present thesis. Permission to reprint these materials is provided by the ALPS collaboration.

1.7 References Cited

[1] P. W. Anderson. More is Different. *Science*, 177:393–396, 1972.

[2] P. W. Anderson. *Basic Notions of Condensed Matter Physics*. Addison-Wesley, New York, 1984.

[3] P. Nozières and D. Pines. *The Theory of Quantum Liquids*, volume II. Addison-Wesley, New York, 1990.

[4] S. Sachdev. *Quantum Phase Transitions*. Cambridge University Press, New York, 1999.

[5] R. B. Laughlin. Anomalous Quantum Hall Effect: An Incompressible Quantum Fluid with Fractionally Charged Excitations. *Phys. Rev. Lett.*, 50:1395–1398, May 1983. doi: 10.1103/PhysRevLett.50.1395. URL `http://link.aps.org/doi/10.1103/PhysRevLett.50.1395`.

[6] B. I. Halperin. Statistics of Quasiparticles and the Hierarchy of Fractional Quantized Hall States. *Phys. Rev. Lett.*, 52:1583–1586, Apr 1984. doi: 10.1103/PhysRevLett.52.1583. URL `http://link.aps.org/doi/10.1103/PhysRevLett.52.1583`.

[7] Daniel Arovas, J. R. Schrieffer, and Frank Wilczek. Fractional Statistics and the Quantum Hall Effect. *Phys. Rev. Lett.*, 53:722–723, Aug 1984. doi: 10.1103/PhysRevLett.53.722. URL `http://link.aps.org/doi/10.1103/PhysRevLett.53.722`.

[8] Kenneth G. Wilson. The renormalization group: Critical phenomena and the Kondo problem. *Rev. Mod. Phys.*, 47:773–840, 1975.

[9] Steven R. White. Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.*, pages 2863–2866, 1992.

[10] Stellan Östlund and Stefan Rommer. Thermodynamic Limit of Density Matrix Renormalization. *Phys. Rev. Lett.*, 75(19):3537–3540, Nov 1995. doi: 10.1103/PhysRevLett.75.3537.

[11] Stefan Rommer and Stellan Östlund. Class of ansatz wave functions for one-dimensional spin systems and their relation to the density matrix renormalization group. *Phys. Rev. B*, 55(4):2164–2181, Jan 1997. doi: 10.1103/PhysRevB.55.2164.

[12] Guifré Vidal. Efficient Classical Simulation of Slightly Entangled Quantum Computations. *Phys. Rev. Lett.*, 91(14):147902, Oct 2003. doi: 10.1103/PhysRevLett.91.147902.

[13] Guifré Vidal. Efficient Simulation of One-Dimensional Quantum Many-Body Systems. *Phys. Rev. Lett.*, 93(4):040502, Jul 2004. doi: 10.1103/PhysRevLett.93.040502.

[14] A. J. Daley, C. Kollath, U. Schollwöck, and G. Vidal. Time-dependent density-matrix renormalization-group using adaptive effective Hilbert spaces. *Journal of Statistical Mechanics: Theory and Experiment*, 2004(04):P04005, 2004. URL http://stacks.iop.org/1742-5468/2004/i=04/a=P04005.

[15] F. Verstraete, D. Porras, and J. I. Cirac. Density Matrix Renormalization Group and Periodic Boundary Conditions: A Quantum Information Perspective. *Phys. Rev. Lett.*, 93(22):227205, Nov 2004. doi: 10.1103/PhysRevLett.93.227205.

[16] F. Verstraete and J. I. Cirac. Renormalization algorithms for Quantum Many-Body systems in two and higher dimensions. *arXiv:cond-mat/0407066v1*, 2004.

[17] F. Verstraete, V. Murg, and J.I. Cirac. Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems. *Advances in Physics*, 57(2):143–224, 2008. doi: 10.1080/14789940801912366. URL http://www.tandfonline.com/doi/abs/10.1080/14789940801912366.

[18] G. Vidal. Entanglement Renormalization. *Phys. Rev. Lett.*, 99:220405, 2007.

[19] J. Bardeen, L. N. Cooper, and J. R. Schrieffer. Theory of Superconductivity. *Phys. Rev.*, 108:1175–1204, 1957.

[20] P. Hohenberg and W. Kohn. Inhomogeneous Electron Gas. *Phys. Rev.*, 136:B864–B871, 1964.

[21] W. Kohn and L. J. Sham. Self-Consistent Equations Including Exchange and Correlation Effects. *Phys. Rev.*, 140:A1133–A1138, Nov 1965. doi: 10.1103/PhysRev.140.A1133. URL http://link.aps.org/doi/10.1103/PhysRev.140.A1133.

[22] SIESTA package for real space density functional theory: http://www.uam.es/departamentos/ciencias/fismateriac/siesta/.

[23] ABINIT package for plane wave density functional theory: http://www.abinit.org/.

[24] OCTOPUS package for time-dependent density functional theory: http://www.tddft.org/programs/octopus/wiki/index.php/Main_Page.

[25] H. J. Metcalf and P. van der Straten. *Laser cooling*. Springer-Verlag, New York, 1999.

[26] W. Ketterle and N. J. van Druten. Evaporative cooling of atoms. *Adv. Atom. Mol. Opt. Phys.*, 37:181, 1996.

[27] C.J. Pethick and H. Smith. *Bose-Einstein Condensation in Dilute Gases*. Cambridge University Press, Cambridge, 2002.

[28] M. H. Anderson, J. R. Ensher, M. R. Matthews, C. E. Wieman, and E. A. Cornell. Observation of Bose-Einstein condensation in a dilute atomic vapor. *Science*, 269:198–201, 1995.

[29] K. B. Davis, M.-O. Mewes, M. R. Andrews, N. J. van Druten, D. S. Durfee, D. M. Kurn, and W. Ketterle. Bose-Einstein Condensation in a Gas of Sodium Atoms. *Phys. Rev. Lett.*, 75:3969–3972, 1995.

[30] C. C. Bradley, C. A. Sackett, J. J. Tollett, and R. G. Hulet. Evidence of Bose-Einstein Condensation in an Atomic Gas with Attractive Interactions. *Phys. Rev. Lett.*, 75:1687, 1995.

[31] C. A. Sackett, H. T. C. Stoof, and R. G. Hulet. Growth and Collapse of a Bose-Einstein Condensate with Attractive Interactions. *Phys. Rev. Lett.*, 80:2031–2034, 1998.

[32] B. DeMarco and D. S. Jin. Onset of Fermi degeneracy in a trapped atomic gas. *Science*, 285:1703–1706, 1999.

[33] G. Roati, F. Riboli, G. Modugno, and M. Inguscio. Fermi-Bose Quantum Degenerate 40K-87Rb Mixture with Attractive Interaction. *Phys. Rev. Lett.*, 89:150403–1–4, 2002.

[34] Z. Hadzibabic, S. Gupta, C. A. Stan, C. H. Schunck, M. W. Zwierlein, K. Dieckmann, and W. Ketterle. Fifty-fold Improvement in the Number of Quantum Degenerate Fermionic Atoms. *Phys. Rev. Lett.*, 91:160401–1–4, 2003.

[35] M. J. Holland, B. DeMarco, and D. S. Jin. Evaporative cooling of a two-component degenerate fermi gas. *Phys. Rev. A*, 61:53610, 2000.

[36] M. Greiner, O. Mandel, T. Esslinger, T. W. Hansch, and I. Bloch. Quantum phase transition from a superfluid to a Mott insulator in a gas of ultracold atoms. *Nature*, 415:39–44, 2002.

[37] M. Greiner, O. Mandel, T.W. Hansch, and I. Bloch. Collapse and revival of the matter wave field of a Bose-Einstein condensate. *Nature*, 419:51–54, 2002.

[38] Cheng Chin, Rudolf Grimm, Paul Julienne, and Eite Tiesinga. Feshbach resonances in ultracold gases. *Rev. Mod. Phys.*, 82:1225–1286, Apr 2010. doi: 10.1103/RevModPhys.82.1225. URL `http://link.aps.org/doi/10.1103/RevModPhys.82.1225`.

[39] Peter J. Martin, Bruce G. Oldaker, Andrew H. Miklich, and David E. Pritchard. Bragg scattering of atoms from a standing light wave. *Phys. Rev. Lett.*, 60:515–518, Feb 1988. doi: 10.1103/PhysRevLett.60.515. URL `http://link.aps.org/doi/10.1103/PhysRevLett.60.515`.

[40] P. B. Blakie and C. W. Clark. Wannier states and Bose-Hubbard parameters for 2D optical lattices. *J. Phys. B: At. Mol. Opt. Phys.*, 37:1391–1404, 2004.

[41] Thilo Stöferle, Henning Moritz, Christian Schori, Michael Köhl, and Tilman Esslinger. Transition from a strongly Interacting 1D Superfluid to a Mott Insulator. *Phys. Rev. Lett.*, 92:130403, Mar 2004. doi: 10.1103/PhysRevLett.92.130403. URL `http://link.aps.org/doi/10.1103/PhysRevLett.92.130403`.

[42] B. Paredes, A. Widera, V. Murg, O. Mandel, S. Folling, I. Cirac, G. V. Shlyapnikov, and T. W. Hansch. Tonks-Girardeau gas of ultracold atoms in an optical lattice. *Nature*, 429:277–281, 2005.

[43] Olaf Mandel, Markus Greiner, Artur Widera, Tim Rom, Theodor W. Hänsch, and Immanuel Bloch. Coherent Transport of Neutral Atoms in Spin-Dependent Optical Lattice Potentials. *Phys. Rev. Lett.*, 91:010407, 2003.

[44] M. Sanpera A. Lewenstein, V. Ahufinger, B. Damski, A. Sen De, and U. Sen. Ultracold atomic gases in optical lattices: Mimicking condensed matter physics and beyond. *Adv. Phys.*, 56:243–379, 2007.

[45] R. P. Feynman. Simulating Physics with Computers. *Int. J. Theor. Phys.*, 21:467, 1982.

[46] D. Jaksch, C. Bruder, J. I. Cirac, C. W. Gardiner, and P. Zoller. Cold Bosonic Atoms in Optical Lattices. *Phys. Rev. Lett.*, 81(15):3108–3111, Oct 1998. doi: 10.1103/PhysRevLett.81.3108.

[47] M. P. A. Fisher, P. B. Weichman, G. Grinstein, and D. S. Fisher. Boson localization and the superfluid-insulator transition. *Phys. Rev. B*, 40:546–570, 1989.

[48] Herman Feshbach. A unified theory of nuclear reactions. II. *Annals of Physics*, 19(2):287 – 313, 1962. ISSN 0003-4916. doi: DOI:10.1016/0003-4916(62)90221-X. URL `http://www.sciencedirect.com/science/article/pii/000349166290221X`.

[49] D S Petrov, C Salomon, and G V Shlyapnikov. Diatomic molecules in ultracold Fermi gases-novel composite bosons. *Journal of Physics B: Atomic, Molecular and Optical Physics*, 38(9):S645, 2005. URL `http://stacks.iop.org/0953-4075/38/i=9/a=014`.

[50] L. D. Landau and E. M. Lifshitz. *Quantum Mechanics (Non-relativistic Theory)*, volume 3. Pergamon Press, Tarrytown, New York, 1977.

[51] T. D. Lee, Kerson Huang, and C. N. Yang. Eigenvalues and Eigenfunctions of a Bose System of Hard Spheres and Its Low-Temperature Properties. *Phys. Rev.*, 106(6):1135–1145, Jun 1957. doi: 10.1103/PhysRev.106.1135.

[52] Zbigniew Idziaszek and Tommaso Calarco. Pseudopotential Method for Higher Partial Wave Scattering. *Phys. Rev. Lett.*, 96:013201, Jan 2006. doi: 10.1103/PhysRevLett.96.013201. URL `http://link.aps.org/doi/10.1103/PhysRevLett.96.013201`.

[53] J. T. Bahns, W. C. Stwalley, and P. L. Gould. Laser cooling of molecules: A sequential scheme for rotation, translation, and vibration. *JOURNAL OF CHEMICAL PHYSICS*, 104(24):9689–9697, JUN 22 1996. ISSN 0021-9606. doi: {10.1063/1.471731}.

[54] M. D. Di Rosa. Laser-cooling molecules. *The European Physical Journal D - Atomic, Molecular, Optical and Plasma Physics*, 31:395–402, 2004. ISSN 1434-6060. URL `http://dx.doi.org/10.1140/epjd/e2004-00167-2`. 10.1140/epjd/e2004-00167-2.

[55] Benjamin K. Stuhl, Brian C. Sawyer, Dajun Wang, and Jun Ye. Magneto-optical Trap for Polar Molecules. *Phys. Rev. Lett.*, 101:243002, Dec 2008. doi: 10.1103/PhysRevLett.101.243002. URL `http://link.aps.org/doi/10.1103/PhysRevLett.101.243002`.

[56] F Robicheaux. A proposal for laser cooling of OH molecules. *Journal of Physics B: Atomic, Molecular and Optical Physics*, 42(19):195301, 2009. URL `http://stacks.iop.org/0953-4075/42/i=19/a=195301`.

[57] Giovanna Morigi, Pepijn W. H. Pinkse, Markus Kowalewski, and Regina de Vivie-Riedle. Cavity Cooling of Internal Molecular Motion. *Phys. Rev. Lett.*, 99:073001, Aug 2007. doi: 10.1103/PhysRevLett.99.073001. URL http://link.aps.org/doi/10.1103/PhysRevLett.99.073001.

[58] Benjamin L. Lev, András Vukics, Eric R. Hudson, Brian C. Sawyer, Peter Domokos, Helmut Ritsch, and Jun Ye. Prospects for the cavity-assisted laser cooling of molecules. *Phys. Rev. A*, 77:023402, Feb 2008. doi: 10.1103/PhysRevA.77.023402. URL http://link.aps.org/doi/10.1103/PhysRevA.77.023402.

[59] Margareta Wallquist, Peter Rabl, Mikhail D Lukin, and Peter Zoller. Theory of cavity-assisted microwave cooling of polar molecules. *New Journal of Physics*, 10(6):063005, 2008. URL http://stacks.iop.org/1367-2630/10/i=6/a=063005.

[60] Zbigniew Idziaszek, Tommaso Calarco, and Peter Zoller. Ion-assisted ground-state cooling of a trapped polar molecule. *Phys. Rev. A*, 83:053413, May 2011. doi: 10.1103/PhysRevA.83.053413. URL http://link.aps.org/doi/10.1103/PhysRevA.83.053413.

[61] S. D. Huber and H. P. Büchler. Dipole-Interaction-Mediated Laser Cooling of Polar Molecules to Ultracold Temperatures. *Phys. Rev. Lett.*, 108:193006, May 2012. doi: 10.1103/PhysRevLett.108.193006. URL http://link.aps.org/doi/10.1103/PhysRevLett.108.193006.

[62] B. Zhao, A. W. Glaetzle, G. Pupillo, and P. Zoller. Atomic Rydberg Reservoirs for Polar Molecules. *Phys. Rev. Lett.*, 108:193007, May 2012. doi: 10.1103/PhysRevLett.108.193007. URL http://link.aps.org/doi/10.1103/PhysRevLett.108.193007.

[63] E. S. Shuman, J. F. Barry, and D. DeMille. Laser cooling of a diatomic molecule. *Nature*, 467(7317):820–823, Oct 2010. ISSN 0028-0836. doi: 10.1038/nature09443. URL http://dx.doi.org/10.1038/nature09443.

[64] Dima Egorov, Thierry Lahaye, Wieland Schöllkopf, Bretislav Friedrich, and John M. Doyle. Buffer-gas cooling of atomic and molecular beams. *Phys. Rev. A*, 66:043401, 2002.

[65] Sebastiaan Y. T. van de Meerakker, Hendrick L. Bethlem, Nicolas Vanhaecke, and Gerard Meijer. Manipulation and control of molecular beams. *Chemical Reviews*, 0(0):null, 0. doi: 10.1021/cr200349r. URL http://pubs.acs.org/doi/abs/10.1021/cr200349r.

[66] Hendrick L. Bethlem, Giel Berden, and Gerard Meijer. Decelerating Neutral Dipolar Molecules. *Phys. Rev. Lett.*, 83:1558–1561, Aug 1999. doi: 10.1103/PhysRevLett.83.1558. URL http://link.aps.org/doi/10.1103/PhysRevLett.83.1558.

[67] J. R. Bochinski H. J. Lewandowski B. C. Sawyer E. R. Hudson and J. Ye. Efficient Stark deceleration of cold polar molecules. *Eur. J. Phys. D: At. Mol. Opt. Phys.*, 31:351–358, 2004.

[68] M. Strebel, S. Spieler, F. Stienkemeier, and M. Mudrich. Guiding slow polar molecules with a charged wire. *Phys. Rev. A*, 84:053430, Nov 2011. doi: 10.1103/PhysRevA.84.053430. URL http://link.aps.org/doi/10.1103/PhysRevA.84.053430.

[69] S. D. Hogan, D. Sprecher, M. Andrist, N. Vanhaecke, and F. Merkt. Zeeman deceleration of H and D. *Phys. Rev. A*, 76:023412, Aug 2007. doi: 10.1103/PhysRevA.76.023412. URL http://link.aps.org/doi/10.1103/PhysRevA.76.023412.

[70] Stephen D. Hogan, Michael Motsch, and Frederic Merkt. Deceleration of supersonic beams using inhomogeneous electric and magnetic fields. *Phys. Chem. Chem. Phys.*, 13:18705–18723, 2011. doi: 10.1039/C1CP21733J. URL http://dx.doi.org/10.1039/C1CP21733J.

[71] Fulton R., Bishop A. I., Shneider M. N., and Barker P. F. Controlling the motion of cold molecules with deep periodic optical potentials. *Nat Phys*, 2 (7):465–468, July 2006. ISSN 1745-2473. doi: 10.1038/nphys339. URL http://dx.doi.org/10.1038/nphys339.

[72] A I Bishop, L Wang, and P F Barker. Creating cold stationary molecular gases by optical stark deceleration. *New Journal of Physics*, 12(7):073028, 2010. URL http://stacks.iop.org/1367-2630/12/i=7/a=073028.

[73] J. F. Barry, E. S. Shuman, E. B. Norrgard, and D. DeMille. Laser Radiation Pressure Slowing of a Molecular Beam. *Phys. Rev. Lett.*, 108:103002, Mar 2012. doi: 10.1103/PhysRevLett.108.103002. URL http://link.aps.org/doi/10.1103/PhysRevLett.108.103002.

[74] Y. B. Band and P. S. Julienne. Ultracold-molecule production by laser-cooled atom photoassociation. *Phys. Rev. A*, 51:R4317–R4320, Jun 1995. doi: 10.1103/PhysRevA.51.R4317. URL http://link.aps.org/doi/10.1103/PhysRevA.51.R4317.

[75] Kevin M. Jones, Eite Tiesinga, Paul D. Lett, and Paul S. Julienne. Ultracold photoassociation spectroscopy: Long-range molecules and atomic scattering. *Rev. Mod. Phys.*, 78:483–535, May 2006. doi: 10.1103/RevModPhys.78.483. URL `http://link.aps.org/doi/10.1103/RevModPhys.78.483`.

[76] T. Takekoshi, B. M. Patterson, and R. J. Knize. Observation of Optically Trapped Cold Cesium Molecules. *Phys. Rev. Lett.*, 81:5105–5108, Dec 1998. doi: 10.1103/PhysRevLett.81.5105. URL `http://link.aps.org/doi/10.1103/PhysRevLett.81.5105`.

[77] Andrew J. Kerman, Jeremy M. Sage, Sunil Sainis, Thomas Bergeman, and David DeMille. Production and State-Selective Detection of Ultracold RbCs Molecules. *Phys. Rev. Lett.*, 92:153001, Apr 2004. doi: 10.1103/PhysRevLett.92.153001. URL `http://link.aps.org/doi/10.1103/PhysRevLett.92.153001`.

[78] Thorsten Köhler, Krzysztof Góral, and Paul S. Julienne. Production of cold molecules via magnetically tunable Feshbach resonances. *Rev. Mod. Phys.*, 78: 1311–1361, Dec 2006. doi: 10.1103/RevModPhys.78.1311. URL `http://link.aps.org/doi/10.1103/RevModPhys.78.1311`.

[79] C. Ospelkaus, S. Ospelkaus, L. Humbert, P. Ernst, K. Sengstock, and K. Bongs. Ultracold Heteronuclear Molecules in a 3D Optical Lattice. *Physical Review Letters*, 97(12):120402, 2006. doi: 10.1103/PhysRevLett.97.120402. URL `http://link.aps.org/abstract/PRL/v97/e120402`.

[80] J. J. Zirbel, K.-K. Ni, S. Ospelkaus, J. P. D'Incao, C. E. Wieman, J. Ye, and D. S. Jin. Collisional Stability of Fermionic Feshbach Molecules. *Phys. Rev. Lett.*, 100:143201, Apr 2008. doi: 10.1103/PhysRevLett.100.143201. URL `http://link.aps.org/doi/10.1103/PhysRevLett.100.143201`.

[81] J. J. Zirbel, K.-K. Ni, S. Ospelkaus, T. L. Nicholson, M. L. Olsen, P. S. Julienne, C. E. Wieman, J. Ye, and D. S. Jin. Heteronuclear molecules in an optical dipole trap. *Phys. Rev. A*, 78:013416, Jul 2008. doi: 10.1103/PhysRevA.78.013416. URL `http://link.aps.org/doi/10.1103/PhysRevA.78.013416`.

[82] K.-K. Ni, S. Ospelkaus, M. H. G. de Miranda, A. Peér, B. Neyenhuis, J. J. Zirbel, S. Kotochigova, P. S. Julienne, D. S. Jin, and Jun Ye. A High Phase-Space-Density Gas of Polar Molecules. *Science*, 322:231–235, 2008.

[83] C. Klempt, T. Henninger, O. Topic, M. Scherer, L. Kattner, E. Tiemann, W. Ertmer, and J. J. Arlt. Radio-frequency association of heteronuclear Feshbach molecules. *Phys. Rev. A*, 78:061602, Dec 2008. doi: 10.1103/PhysRevA.78.061602. URL `http://link.aps.org/doi/10.1103/PhysRevA.78.061602`.

[84] A.-C. Voigt, M. Taglieber, L. Costa, T. Aoki, W. Wieser, T. W. Hänsch, and K. Dieckmann. Ultracold Heteronuclear Fermi-Fermi Molecules. *Phys. Rev. Lett.*, 102(2):020405, Jan 2009. doi: 10.1103/PhysRevLett.102.020405.

[85] Myoung-Sun Heo, Tout T. Wang, Caleb A. Christensen, Timur M. Rvachov, Dylan A. Cotta, Jae-Hoon Choi, Ye-Ryoung Lee, and Wolfgang Ketterle. Formation of Ultracold Fermionic NaLi Feshbach Molecules. *e-print arXiv.org:1205.5304*, 2012.

[86] Cheng-Hsun Wu, Jee Woo Park, Peyman Ahmadi, Sebastian Will, and Martin W. Zwierlein. Ultracold Fermionic Feshbach Molecules of ^{23}Na^{40}K. *e-print arXiv.org:1206.5023*, 2012.

[87] S. Jochim, M. Bartenstein, A. Altmeyer, G. Hendl, C. Chin, J. H. Denschlag, and R. Grimm. Pure Gas of Optically Trapped Molecules Created from Fermionic Atoms. *Phys. Rev. Lett.*, 91:240402–1–4, 2003.

[88] S. Jochim, M. Bartenstein, A. Altmeyer, S. Riedl, C. Chin, J. H. Denschlag, and R. Grimm. Bose-Einstein condensation of molecules. *Science*, 302:2102–2103, 2003.

[89] M. Greiner, C. A. Regal, and D. S. Jin. Emergence of a molecular Bose-Einstein condensate from a Fermi gas. *Nature*, 426:437–540, 2003.

[90] K. Xu, T. Mukaiyama, J. R. Abo-Shaeer, J. K. Chin, D. E. Miller, and W. Ketterle. Formation of Quantum-Degenerate Sodium Molecules. *Phys. Rev. Lett.*, 91:210402–1–4, 2003.

[91] C. A. Regal, M. Greiner, and D. S. Jin. Observation of Resonance Condensation of Fermionic Atom Pairs. *Phys. Rev. Lett.*, 92:040403–1–4, 2004.

[92] M. W. Zwierlein, C. A. Stan, C. H. Schunck, S. M. F. Raupach, A. J. Kerman, and W. Ketterle. Condensation of Pairs of Fermionic Atoms near a Feshbach Resonance. *Phys. Rev. Lett.*, 92:120403–1–4, 2004.

[93] T. Bourdel, L. Khaykovich, J. Cubizolles, J. Zhang, F. Chevy, M. Teichmann, L. Tarruell, S. J. J. M. F. Kokkelmans, and C. Salomon. Experimental Study of the BEC-BCS Crossover Region in Lithium 6. *Phys. Rev. Lett.*, 93:050401–1–4, 2004.

[94] Jens Herbig, Tobias Kraemer, Michael Mark, Tino Weber, Cheng Chin, Hanns-Christoph Ngerl, and Rudolf Grimm. Preparation of a Pure Molecular Quantum Gas. *Science*, 301(5639):1510–1513, 2003. doi: 10.1126/science.1088876. URL `http://www.sciencemag.org/content/301/5639/1510.abstract`.

[95] J. G. Danzl, E. Haller, M. Gustavsson, M. J. Mark, R. Hart, N. Bouloufa, O. Dulieu, H. Ritsch, and H.-C Nägerl. Quantum gas of deeply bound ground state molecules. *Science*, 321:1062, 2008.

[96] M. Bartenstein, A. Altmeyer, S. Riedl, S. Jochim, C. Chin, J. H. Denschlag, and R. Grimm. Crossover from a molecular Bose-Einstein condensate to a degenerate Fermi gas. *Phys. Rev. Lett.*, 92:120401–1–4, 2004.

[97] J. Kinast, S. L. Hemmer, M. E. Gehm, A. Turlapov, and J. E. Thomas. Evidence for Superfluidity in a Resonantly Interacting Fermi Gas. *Phys. Rev. Lett.*, 92: 150402–1–4, 2004.

[98] Stephan Dürr, Thomas Volz, Andreas Marte, and Gerhard Rempe. Observation of Molecules Produced from a Bose-Einstein Condensate. *Phys. Rev. Lett.*, 92: 020406, Jan 2004. doi: 10.1103/PhysRevLett.92.020406. URL `http://link.aps.org/doi/10.1103/PhysRevLett.92.020406`.

[99] D. M. Eagles. Possible Pairing without Superconductivity at Low Carrier Concentrations in Bulk and Thin-Film Superconducting Semiconductors. *Phys. Rev.*, 186:456–463, 1969.

[100] A. J. Leggett. Diatomic Molecules and Cooper Pairs. In A. Pekalski and R. Przystawa, editors, *Modern Trends in the theory of Condensed Matter*. Springer-Verlag, Berlin, 1980.

[101] P. Noziéres and S. Schmitt-Rink. Bose condensation in an attractive fermion gas: From weak to strong coupling superconductivity. *Journal of Low Temperature Physics*, 59:195–211, 1985. ISSN 0022-2291. URL `http://dx.doi.org/10.1007/BF00683774`. 10.1007/BF00683774.

[102] Qijin Chen, Jelena Stajic, Shina Tan, and K. Levin. Bcs-bec crossover: From high temperature superconductors to ultracold superfluids. *Physics Reports*, 412(1):1 – 88, 2005. ISSN 0370-1573. doi: DOI:10.1016/j.physrep.2005.02.005. URL `http://www.sciencedirect.com/science/article/pii/S0370157305001067`.

[103] Allan Adams, Lincoln D. Carr, Thomas Schaefer, Peter Steinberg, and John E. Thomas. Strongly Correlated Quantum Fluids: Ultracold Quantum Gases, Quantum Chromodynamic Plasmas, and Holographic Duality, 2012. URL `http://arxiv.org/abs/1205.5180`.

[104] K. Winkler, F. Lang, G. Thalhammer, P. v. d. Straten, R. Grimm, and J. Hecker Denschlag. Coherent Optical Transfer of Feshbach Molecules to a Lower Vibrational State. *Phys. Rev. Lett.*, 98(4):043201, Jan 2007. doi: 10.1103/PhysRevLett.98.043201.

[105] S. Ospelkaus, A. Pe'er, K. K. Ni, J. J. Zirbel, B. Neyenhuis, S. Kotochigova, P. S. Julienne, J. Ye, and D. S. Jin. Ultracold dense gas of deeply bound heteronuclear molecules, 2008.

[106] S. Ospelkaus, K. K. Ni, M. H. G. de Miranda, A. Pe'er, B. Nyenhuis, D. Wang, S. Kotochigova, P. S. Julienne, D. S. Jin, , and Jun Ye. Ultracold polar molecules near quantum degeneracy. *Faraday Discuss.*, 142:351–359, 2009.

[107] J. G. Danzl, M. J. Mark, E. Haller, M. Gustavsson, R. Hart, J. Aldegunde, J. M. Hutson, and H.-C Nägerl. An ultracold high-density sample of rovibronic ground-state molecules in an optical lattice. *Nature Phys.*, 6:265–270, 2010.

[108] S. Ospelkaus, K.-K. Ni, G. Quéméner, B. Neyenhuis, D. Wang, M. H. G. de Miranda, J. L. Bohn, J. Ye, and D. S. Jin. Controlling the Hyperfine State of Rovibronic Ground-State Polar Molecules. *Phys. Rev. Lett.*, 104(3):030402, Jan 2010. doi: 10.1103/PhysRevLett.104.030402.

[109] J. Deiglmayr, A. Grochola, M. Repp, K. Mörtlbauer, C. Glück, J. Lange, O. Dulieu, R. Wester, and M. Weidemüller. Formation of Ultracold Polar Molecules in the Rovibrational Ground State. *Phys. Rev. Lett.*, 101(13):133004, Sep 2008. doi: 10.1103/PhysRevLett.101.133004.

[110] H.W. Cho, D.J. McCarron, D. L. Jenkin, M. P. Köppinger, and S. L. Cornish. A high phase-space density mixture of ^{87}Rb and ^{133}Cs: towards ultracold heteronuclear molecules. *The European Physical Journal D - Atomic, Molecular, Optical and Plasma Physics*, pages 1–7. ISSN 1434-6060. URL http://dx.doi.org/10.1140/epjd/e2011-10716-1. 10.1140/epjd/e2011-10716-1.

[111] Svetlana Kotochigova and David DeMille. Electric-field-dependent dynamic polarizability and state-insensitive conditions for optical trapping of diatomic polar molecules. *Phys. Rev. A*, 82:063421, Dec 2010. doi: 10.1103/PhysRevA.82.063421. URL http://link.aps.org/doi/10.1103/PhysRevA.82.063421.

[112] J. Aldegunde, Ben A. Rivington, Piotr S. Żuchowski, and Jeremy M. Hutson. Hyperfine energy levels of alkali-metal dimers: Ground-state polar molecules in electric and magnetic fields. *Phys. Rev. A*, 78:033434, 2008.

[113] Hong Ran, J Aldegunde, and Jeremy M Hutson. Hyperfine structure in the microwave spectra of ultracold polar molecules. *New Journal of Physics*, 12(4): 043015, 2010. URL http://stacks.iop.org/1367-2630/12/i=4/a=043015.

[114] K. K. Ni, S. Ospelkaus, D. Wang, G. Quemener, B. Neyenhuis, M. H. G. de Miranda, J. L. Bohn, J. Ye, and D. S. Jin. Dipolar collisions of polar molecules in the quantum regime. *Nature (London)*, 464:1324, 2010.

[115] M. H. G. de Miranda, A. Chotia, B. Neyenhuis, D. Wang, G. Quemener, S. Ospelkaus, J. L. Bohn, J. Ye, and D. S. Jin. Controlling the quantum stereodynamics of ultracold bimolecular reactions. *Nat Phys*, 7(6):502–507, Jun 2011. ISSN 1745-2473. doi: 10.1038/nphys1939. URL http://dx.doi.org/10.1038/nphys1939.

[116] M. Marinescu and L. You. Controlling Atom-Atom Interaction at Ultralow Temperatures by dc Electric Fields. *Phys. Rev. Lett.*, 81:4596–4599, 1998.

[117] S. Yi and L. You. Trapped atomic condensates with anisotropic interactions. *Phys. Rev. A*, 61:041604, Mar 2000. doi: 10.1103/PhysRevA.61.041604. URL http://link.aps.org/doi/10.1103/PhysRevA.61.041604.

[118] B. Deb and L. You. Low-energy atomic collision with dipole interactions. *Phys. Rev. A*, 64:022717, Jul 2001. doi: 10.1103/PhysRevA.64.022717. URL http://link.aps.org/doi/10.1103/PhysRevA.64.022717.

[119] D. C. E. Bortolotti, S. Ronen, J. L. Bohn, and D. Blume. Scattering Length Instability in Dipolar Bose-Einstein Condensates. *Phys. Rev. Lett.*, 97:160402, Oct 2006. doi: 10.1103/PhysRevLett.97.160402. URL http://link.aps.org/doi/10.1103/PhysRevLett.97.160402.

[120] Shai Ronen, Daniele C. E. Bortolotti, D. Blume, and John L. Bohn. Dipolar Bose-Einstein condensates with dipole-dependent scattering length. *Phys. Rev. A*, 74:033611, Sep 2006. doi: 10.1103/PhysRevA.74.033611. URL http://link.aps.org/doi/10.1103/PhysRevA.74.033611.

[121] Alexey V. Gorshkov, Salvatore R. Manmana, Gang Chen, Jun Ye, Eugene Demler, Mikhail D. Lukin, and Ana Maria Rey. Tunable Superfluidity and Quantum Magnetism with Ultracold Polar Molecules. *Phys. Rev. Lett.*, 107:115301, Sep 2011. doi: 10.1103/PhysRevLett.107.115301. URL http://link.aps.org/doi/10.1103/PhysRevLett.107.115301.

[122] Alexey V. Gorshkov, Salvatore R. Manmana, Gang Chen, Eugene Demler, Mikhail D. Lukin, and Ana Maria Rey. Quantum magnetism with polar alkali-metal dimers. *Phys. Rev. A*, 84:033619, Sep 2011. doi: 10.1103/PhysRevA.84.033619. URL http://link.aps.org/doi/10.1103/PhysRevA.84.033619.

[123] M. L. Wall and L. D. Carr. Hyperfine molecular Hubbard Hamiltonian. *Phys. Rev. A*, 82(1):013611, Jul 2010. doi: 10.1103/PhysRevA.82.013611.

[124] Michael Mayle, Brandon P. Ruzic, and John L. Bohn. Statistical aspects of ultracold resonant scattering. *Phys. Rev. A*, 85:062712, Jun 2012. doi: 10.1103/PhysRevA.85.062712. URL http://link.aps.org/doi/10.1103/PhysRevA.85.062712.

[125] S. Ospelkaus, K.-K. Ni, D. Wang, M. H. G. de Miranda, B. Neyenhuis, G. Quéméner, P. S. Julienne, J. L. Bohn, D. S. Jin, and J. Ye. Quantum-State Controlled Chemical Reactions of Ultracold Potassium-Rubidium Molecules. *Science*, 327(5967):853–857, 2010. doi: 10.1126/science.1184121. URL http://www.sciencemag.org/content/327/5967/853.abstract.

[126] Amodsen Chotia, Brian Neyenhuis, Steven A. Moses, Bo Yan, Jacob P. Covey, Michael Foss-Feig, Ana Maria Rey, Deborah S. Jin, and Jun Ye. Long-Lived Dipolar Molecules and Feshbach Molecules in a 3D Optical Lattice. *Phys. Rev. Lett.*, 108:080405, Feb 2012. doi: 10.1103/PhysRevLett.108.080405. URL http://link.aps.org/doi/10.1103/PhysRevLett.108.080405.

[127] G. F. Gribakin and V. V. Flambaum. Calculation of the scattering length in atomic collisions using the semiclassical approximation. *Phys. Rev. A*, 48:546–553, Jul 1993. doi: 10.1103/PhysRevA.48.546. URL http://link.aps.org/doi/10.1103/PhysRevA.48.546.

[128] M Aymar and O Dulieu. Calculation of accurate permanent dipole moments of the lowest ${}^{1,3}\Sigma^+$ states of heteronuclear alkali dimers using extended basis sets. *JOURNAL OF CHEMICAL PHYSICS*, 122(20), MAY 22 2005. ISSN 0021-9606. doi: {10.1063/1.1903944}.

[129] James F. E. Croft, Alisdair O. G. Wallis, Jeremy M. Hutson, and Paul S. Julienne. Multichannel quantum defect theory for cold molecular collisions. *Phys. Rev. A*, 84:042703, Oct 2011. doi: 10.1103/PhysRevA.84.042703. URL http://link.aps.org/doi/10.1103/PhysRevA.84.042703.

[130] Zbigniew Idziaszek, Goulven Quéméner, John L. Bohn, and Paul S. Julienne. Simple quantum model of ultracold polar molecule collisions. *Phys. Rev. A*, 82:020703, Aug 2010. doi: 10.1103/PhysRevA.82.020703. URL http://link.aps.org/doi/10.1103/PhysRevA.82.020703.

[131] Paul S. Julienne, Thomas M. Hanna, and Zbigniew Idziaszek. Universal ultracold collision rates for polar molecules of two alkali-metal atoms. *Phys. Chem. Chem. Phys.*, 13:19114–19124, 2011. doi: 10.1039/C1CP21270B. URL `http://dx.doi.org/10.1039/C1CP21270B`.

[132] Zbigniew Idziaszek and Paul S. Julienne. Universal Rate Constants for Reactive Collisions of Ultracold Molecules. *Phys. Rev. Lett.*, 104:113202, Mar 2010. doi: 10.1103/PhysRevLett.104.113202. URL `http://link.aps.org/doi/10.1103/PhysRevLett.104.113202`.

[133] Andrea Micheli, Zbigniew Idziaszek, Guido Pupillo, Mikhail A. Baranov, Peter Zoller, and Paul S. Julienne. Universal Rates for Reactive Ultracold Polar Molecules in Reduced Dimensions. *Phys. Rev. Lett.*, 105:073202, Aug 2010. doi: 10.1103/PhysRevLett.105.073202. URL `http://link.aps.org/doi/10.1103/PhysRevLett.105.073202`.

[134] Thomas M. Hanna, Eite Tiesinga, William F. Mitchell, and Paul S. Julienne. Resonant control of polar molecules in individual sites of an optical lattice. *Phys. Rev. A*, 85:022703, Feb 2012. doi: 10.1103/PhysRevA.85.022703. URL `http://link.aps.org/doi/10.1103/PhysRevA.85.022703`.

[135] Goulven Quéméner and John L. Bohn. Electric field suppression of ultracold confined chemical reactions. *Phys. Rev. A*, 81:060701, Jun 2010. doi: 10.1103/PhysRevA.81.060701. URL `http://link.aps.org/doi/10.1103/PhysRevA.81.060701`.

[136] Goulven Quéméner and John L. Bohn. Strong dependence of ultracold chemical rates on electric dipole moments. *Phys. Rev. A*, 81:022702, Feb 2010. doi: 10.1103/PhysRevA.81.022702. URL `http://link.aps.org/doi/10.1103/PhysRevA.81.022702`.

[137] Goulven Quéméner, John L. Bohn, Alexander Petrov, and Svetlana Kotochigova. Universalities in ultracold reactions of alkali-metal polar molecules. *Phys. Rev. A*, 84:062703, Dec 2011. doi: 10.1103/PhysRevA.84.062703. URL `http://link.aps.org/doi/10.1103/PhysRevA.84.062703`.

[138] N. Syassen, D. M. Bauer, M. Lettner, T. Volz, D. Dietze, J. J. García-Ripoll, J. I. Cirac, G. Rempe, and S. Dürr. Strong Dissipation Inhibits Losses and Induces Correlations in Cold Molecular Gases. *Science*, 320(5881):1329–1331, 2008. doi: 10.1126/science.1155309. URL `http://www.sciencemag.org/content/320/5881/1329.abstract`.

[139] N. Schuch and F. Verstraete. Computational complexity of interacting electrons and fundamental limitations of density functional theory. *Nat Phys*, 5(10):732–735, Oct 2009. doi: 10.1038/nphys1370. URL `http://dx.doi.org/10.1038/nphys1370`.

[140] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins Studies in Mathematical Sciences. The Johns Hopkins University Press, Baltimore, 3 edition, 1996.

[141] Ernest R. Davidson. The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices. *Journal of Computational Physics*, 17(1):87 – 94, 1975. ISSN 0021-9991. doi: 10.1016/0021-9991(75)90065-0. URL `http://www.sciencedirect.com/science/article/pii/0021999175900650`.

[142] R. P. Feynman. Atomic Theory of the λ Transition in Helium. *Phys. Rev.*, 91:1291–1301, Sep 1953. doi: 10.1103/PhysRev.91.1291. URL `http://link.aps.org/doi/10.1103/PhysRev.91.1291`.

[143] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087, 1953.

[144] H. G. Evertz. The loop algorithm. *Advances in Physics*, 52(1):1–66, 2003. doi: 10.1080/0001873021000049195. URL `http://www.tandfonline.com/doi/abs/10.1080/0001873021000049195`.

[145] N. Prokof'ev, B. Svistunov, and I. Tupitsyn. Exact, complete, and universal continuous-time worldline Monte Carlo approach to the statistics of discrete quantum systems. *Journal of Experimental and Theoretical Physics*, 87:310–321, 1998. ISSN 1063-7761. URL `http://dx.doi.org/10.1134/1.558661`. 10.1134/1.558661.

[146] Matthias Troyer and Uwe-Jens Wiese. Computational Complexity and Fundamental Limitations to Fermionic Quantum Monte Carlo Simulations. *Phys. Rev. Lett.*, 94:170201, May 2005. doi: 10.1103/PhysRevLett.94.170201. URL `http://link.aps.org/doi/10.1103/PhysRevLett.94.170201`.

[147] R.C Grimm and R.G Storer. Monte-Carlo solution of Schrödinger's equation. *Journal of Computational Physics*, 7(1):134 – 156, 1971. ISSN 0021-9991. doi: 10.1016/0021-9991(71)90054-4. URL `http://www.sciencedirect.com/science/article/pii/0021999171900544`.

[148] W. L. McMillan. Ground State of Liquid He^4. *Phys. Rev.*, 138:A442–A451, Apr 1965. doi: 10.1103/PhysRev.138.A442. URL `http://link.aps.org/doi/10.1103/PhysRev.138.A442`.

[149] Nikolai V. Prokof'ev and Boris V. Svistunov. Polaron Problem by Diagrammatic Quantum Monte Carlo. *Phys. Rev. Lett.*, 81:2514–2517, Sep 1998. doi: 10.1103/PhysRevLett.81.2514. URL `http://link.aps.org/doi/10.1103/PhysRevLett.81.2514`.

[150] Nikolay Prokof'ev and Boris Svistunov. Bold Diagrammatic Monte Carlo Technique: When the Sign Problem Is Welcome. *Phys. Rev. Lett.*, 99:250201, Dec 2007. doi: 10.1103/PhysRevLett.99.250201. URL `http://link.aps.org/doi/10.1103/PhysRevLett.99.250201`.

[151] N. V. Prokof'ev and B. V. Svistunov. Bold diagrammatic Monte Carlo: A generic sign-problem tolerant technique for polaron models and possibly interacting many-body problems. *Phys. Rev. B*, 77:125101, Mar 2008. doi: 10.1103/PhysRevB.77.125101. URL `http://link.aps.org/doi/10.1103/PhysRevB.77.125101`.

[152] K. Van Houcke, F. Werner, E. Kozik, N. Prokof/'ev, B. Svistunov, M. J. H. Ku, A. T. Sommer, L. W. Cheuk, A. Schirotzek, and M. W. Zwierlein. Feynman diagrams versus Fermi-gas Feynman emulator. *Nat Phys*, 8(5):366–370, May 2012. ISSN 1745-2473. doi: 10.1038/nphys2273. URL `http://dx.doi.org/10.1038/nphys2273`.

[153] A. L. Fetter and J. D. Walecka. *Quantum Theory of Many-Particle Systems*. McGraw-Hill, Inc., New York, 1971.

[154] Marco Schiró. Real-time dynamics in quantum impurity models with diagrammatic Monte Carlo. *Phys. Rev. B*, 81:085126, Feb 2010. doi: 10.1103/PhysRevB.81.085126. URL `http://link.aps.org/doi/10.1103/PhysRevB.81.085126`.

[155] Walter Metzner and Dieter Vollhardt. Correlated Lattice Fermions in $d = \infty$ Dimensions. *Phys. Rev. Lett.*, 62:324–327, Jan 1989. doi: 10.1103/PhysRevLett.62.324. URL `http://link.aps.org/doi/10.1103/PhysRevLett.62.324`.

[156] Antoine Georges and Gabriel Kotliar. Hubbard model in infinite dimensions. *Phys. Rev. B*, 45:6479–6483, Mar 1992. doi: 10.1103/PhysRevB.45.6479. URL `http://link.aps.org/doi/10.1103/PhysRevB.45.6479`.

[157] Antoine Georges, Gabriel Kotliar, Werner Krauth, and Marcelo J. Rozenberg. Dynamical mean-field theory of strongly correlated fermion systems and the limit of infinite dimensions. *Rev. Mod. Phys.*, 68:13–125, Jan 1996. doi: 10.1103/RevModPhys.68.13. URL `http://link.aps.org/doi/10.1103/RevModPhys.68.13`.

[158] G. Kotliar, S. Y. Savrasov, K. Haule, V. S. Oudovenko, O. Parcollet, and C. A. Marianetti. Electronic structure calculations with dynamical mean-field theory. *Rev. Mod. Phys.*, 78:865–951, Aug 2006. doi: 10.1103/RevModPhys.78.865. URL `http://link.aps.org/doi/10.1103/RevModPhys.78.865`.

[159] I. Titvinidze, M. Snoek, and W. Hofstetter. Generalized dynamical mean-field theory for Bose-Fermi mixtures in optical lattices. *Phys. Rev. B*, 79:144506, 2009.

[160] Wen-Jun Hu and Ning-Hua Tong. Dynamical mean-field theory for the Bose-Hubbard model. *Phys. Rev. B*, 80:245110, Dec 2009. doi: 10.1103/PhysRevB.80.245110. URL `http://link.aps.org/doi/10.1103/PhysRevB.80.245110`.

[161] Adrian E. Feiguin and Steven R. White. Finite-temperature density matrix renormalization using an enlarged Hilbert space. *Phys. Rev. B*, 72(22):220401, Dec 2005. doi: 10.1103/PhysRevB.72.220401.

[162] Steven R. White. Minimally Entangled Typical Quantum States at Finite Temperature. *Phys. Rev. Lett.*, 102(19):190601, May 2009. doi: 10.1103/PhysRevLett.102.190601.

[163] E M Stoudenmire and Steven R White. Minimally entangled typical thermal state algorithms. *New Journal of Physics*, 12(5):055026, 2010. URL `http://stacks.iop.org/1367-2630/12/i=5/a=055026`.

[164] Michael L. Wall and Lincoln D. Carr. Finite temperature matrix product state algorithms and applications. *"Chapter in 'Non-Equilibrium Quantum Gases at Finite Temperatures', ed. M. Davis, S. Gardiner, N. Nygaard, N. Proukakis, and M. Szymanska (World Scientific, 2011); e-print http://arxiv.org/abs/1008.4303"*, 2011.

[165] J Schachenmayer, I Lesanovsky, A Micheli, and A J Daley. Dynamical crystal creation with polar molecules or Rydberg atoms in optical lattices. *New Journal of Physics*, 12(10):103044, 2010. URL `http://stacks.iop.org/1367-2630/12/i=10/a=103044`.

[166] Ian P. McCulloch. From density-matrix renormalization group to matrix product states. *Journal of Statistical Mechanics: Theory and Experiment*, 2007 (10):P10014, 2007. URL `http://stacks.iop.org/1742-5468/2007/i=10/a=P10014`.

[167] Gregory M. Crosswhite, A. C. Doherty, and Guifré Vidal. Applying matrix product operators to model systems with long-range interactions. *Phys. Rev. B*, 78:035116, Jul 2008. doi: 10.1103/PhysRevB.78.035116. URL `http://link.aps.org/doi/10.1103/PhysRevB.78.035116`.

[168] B Pirvu, V Murg, J I Cirac, and F Verstraete. Matrix product operator representations. *New Journal of Physics*, 12(2):025012, 2010. URL `http://stacks.iop.org/1367-2630/12/i=2/a=025012`.

[169] M. Dalmonte, G. Pupillo, and P. Zoller. One-Dimensional Quantum Liquids with Power-Law Interactions: The Luttinger Staircase. *Phys. Rev. Lett.*, 105: 140401, Sep 2010. doi: 10.1103/PhysRevLett.105.140401. URL `http://link.aps.org/doi/10.1103/PhysRevLett.105.140401`.

[170] B. Capogrosso-Sansone, C. Trefzger, M. Lewenstein, P. Zoller, and G. Pupillo. Quantum Phases of Cold Polar Molecules in 2D Optical Lattices. *Phys. Rev. Lett.*, 104:125301, 2010.

[171] N. D. Mermin and H. Wagner. Absence of Ferromagnetism or Antiferromagnetism in One- or Two-Dimensional Isotropic Heisenberg Models. *Phys. Rev. Lett.*, 17:1133–1136, Nov 1966. doi: 10.1103/PhysRevLett.17.1133. URL `http://link.aps.org/doi/10.1103/PhysRevLett.17.1133`.

[172] P. C. Hohenberg. Existence of Long-Range Order in One and Two Dimensions. *Phys. Rev.*, 158:383–386, Jun 1967. doi: 10.1103/PhysRev.158.383. URL `http://link.aps.org/doi/10.1103/PhysRev.158.383`.

[173] A. Griesmaier, J. Werner, S. Hensler, J. Stuhler, , and T. Pfau. Bose-Einstein Condensation of Chromium. *Phys. Rev. Lett.*, 94:160401–1–4, 2005.

[174] Q. Beaufils, R. Chicireanu, T. Zanon, B. Laburthe-Tolra, E. Maréchal, L. Vernac, J.-C. Keller, and O. Gorceix. All-optical production of chromium Bose-Einstein condensates. *Phys. Rev. A*, 77:061601, Jun 2008. doi: 10.1103/PhysRevA.77.061601. URL `http://link.aps.org/doi/10.1103/PhysRevA.77.061601`.

[175] Mingwu Lu, Nathaniel Q. Burdick, Seo Ho Youn, and Benjamin L. Lev. Strongly Dipolar Bose-Einstein Condensate of Dysprosium. *Phys. Rev. Lett.*, 107:190401, Oct 2011. doi: 10.1103/PhysRevLett.107.190401. URL http://link.aps.org/doi/10.1103/PhysRevLett.107.190401.

[176] K. Aikawa, A. Frisch, M. Mark, S. Baier, A. Rietzler, R. Grimm, and F. Ferlaino. Bose-Einstein Condensation of Erbium. *Phys. Rev. Lett.*, 108:210401, May 2012. doi: 10.1103/PhysRevLett.108.210401. URL http://link.aps.org/doi/10.1103/PhysRevLett.108.210401.

[177] Mingwu Lu, Nathaniel Q. Burdick, and Benjamin L. Lev. Quantum Degenerate Dipolar Fermi Gas. *Phys. Rev. Lett.*, 108:215301, May 2012. doi: 10.1103/PhysRevLett.108.215301. URL http://link.aps.org/doi/10.1103/PhysRevLett.108.215301.

[178] T. Lahaye, J. Metz, B. Fröhlich, T. Koch, M. Meister, A. Griesmaier, T. Pfau, H. Saito, Y. Kawaguchi, and M. Ueda. *d*-wave Collapse and Explosion of a Dipolar Bose-Einstein Condensate. *Phys. Rev. Lett.*, 101:080401, 2008.

[179] M. A. Baranov. Theoretical progress in many-body physics with ultracold dipolar gases. *Physics Reports*, 464(3):71–111, 2008.

[180] T Lahaye, C Menotti, L Santos, M Lewenstein, and T Pfau. The physics of dipolar bosonic quantum gases. *Reports on Progress in Physics*, 72(12):126401, 2009. URL http://stacks.iop.org/0034-4885/72/i=12/a=126401.

[181] L. Santos, G. V. Shlyapnikov, P. Zoller, and M. Lewenstein. Bose-Einstein condensation in trapped dipolar gases. *Phys. Rev. Lett.*, 85:1791–1794, 2000. Erratum: L. Santos, G. V. Shlyapnikov, P. Zoller, and M. Lewenstein, Erratum: Phys. Rev. Lett. 85, 1791 (2000).

[182] S. Yi, L. You, and H. Pu. Quantum Phases of Dipolar Spinor Condensates. *Phys. Rev. Lett.*, 93:040403–1–4, 2004.

[183] S. Sinha and L. Santos. Cold Dipolar Gases in Quasi-One-Dimensional Geometries. *Phys. Rev. Lett.*, 99:140406, 2007.

[184] Shai Ronen, Daniele C. E. Bortolotti, and John L. Bohn. Radial and Angular Rotons in Trapped Dipolar Gases. *Phys. Rev. Lett.*, 98:030406, Jan 2007. doi: 10.1103/PhysRevLett.98.030406. URL http://link.aps.org/doi/10.1103/PhysRevLett.98.030406.

[185] K. Kanjilal, John L. Bohn, and D. Blume. Pseudopotential treatment of two aligned dipoles under external harmonic confinement. *Phys. Rev. A*, 75:052705, May 2007. doi: 10.1103/PhysRevA.75.052705. URL `http://link.aps.org/doi/10.1103/PhysRevA.75.052705`.

[186] Ryan M. Wilson, Shai Ronen, John L. Bohn, and Han Pu. Manifestations of the Roton Mode in Dipolar Bose-Einstein Condensates. *Phys. Rev. Lett.*, 100:245302, Jun 2008. doi: 10.1103/PhysRevLett.100.245302. URL `http://link.aps.org/doi/10.1103/PhysRevLett.100.245302`.

[187] C. Menotti, C. Trefzger, and M. Lewenstein. Metastable States of a Gas of Dipolar Bosons in a 2D Optical Lattice. *Phys. Rev. Lett.*, 98:235301, 2007.

[188] C. Trefzger, C. Menotti, and M. Lewenstein. Ultracold dipolar gas in an optical lattice: The fate of metastable states. *Phys. Rev. A*, 78:043604, Oct 2008. doi: 10.1103/PhysRevA.78.043604. URL `http://link.aps.org/doi/10.1103/PhysRevA.78.043604`.

[189] K. Goral, L. Santos, and M. Lewenstein. Quantum Phases of Dipolar Bosons in Optical Lattices. *Phys. Rev. Lett.*, 88:170406–1–4, 2002.

[190] B. Damski, L. Santos, E. Tiemann, M. Lewenstein, S. Kotochigova, P. Julienne, and P. Zoller. Creation of a dipolar superfluid in optical lattices. *Phys. Rev. Lett.*, 90:110401–1–4, 2003.

[191] Ippei Danshita and Carlos A. R. Sá de Melo. Stability of Superfluid and Supersolid Phases of Dipolar Bosons in Optical Lattices. *Phys. Rev. Lett.*, 103:225301, Nov 2009. doi: 10.1103/PhysRevLett.103.225301. URL `http://link.aps.org/doi/10.1103/PhysRevLett.103.225301`.

[192] L. Pollet, J. D. Picon, H. P. Büchler, and M. Troyer. Supersolid Phase with Cold Polar Molecules on a Triangular Lattice. *Phys. Rev. Lett.*, 104:125302, Mar 2010. doi: 10.1103/PhysRevLett.104.125302. URL `http://link.aps.org/doi/10.1103/PhysRevLett.104.125302`.

[193] Ryan M. Wilson and John L. Bohn. Emergent structure in a dipolar Bose gas in a one-dimensional lattice. *Phys. Rev. A*, 83:023623, Feb 2011. doi: 10.1103/PhysRevA.83.023623. URL `http://link.aps.org/doi/10.1103/PhysRevA.83.023623`.

[194] Massimo Boninsegni and Nikolay V. Prokof'ev. *Colloquium* : Supersolids: What and where are they? *Rev. Mod. Phys.*, 84:759–776, May 2012. doi: 10.1103/RevModPhys.84.759. URL `http://link.aps.org/doi/10.1103/RevModPhys.84.759`.

[195] M. Olshanii. Atomic Scattering in the Presence of an External Confinement and a Gas of Impenetrable Bosons. *Phys. Rev. Lett.*, 81:938–941, 1998.

[196] D. DeMille. Quantum Computation with Trapped Polar Molecules. *Phys. Rev. Lett.*, 88:067901, 2002.

[197] A. Micheli, G. K. Brennen, and P. Zoller. A toolbox for lattice-spin models with polar molecules. *Nat Phys*, 2(5):341–347, May 2006. ISSN 1745-2473. doi: 10.1038/nphys287. URL `http://dx.doi.org/10.1038/nphys287`.

[198] Gavin K Brennen, Andrea Micheli, and Peter Zoller. Designing spin-1 lattice models using polar molecules. *New J. Phys.*, 9:138, 2007.

[199] Felipe Herrera and Roman V. Krems. Tunable Holstein model with cold polar molecules. *Phys. Rev. A*, 84:051401, Nov 2011. doi: 10.1103/PhysRevA.84.051401. URL `http://link.aps.org/doi/10.1103/PhysRevA.84.051401`.

[200] J Pérez-Ríos, F Herrera, and R V Krems. External field control of collective spin excitations in an optical lattice of $^2\Sigma$ molecules. *New Journal of Physics*, 12(10):103007, 2010. URL `http://stacks.iop.org/1367-2630/12/i=10/a=103007`.

[201] Felipe Herrera, Marina Litinskaya, and Roman V. Krems. Tunable disorder in a crystal of cold polar molecules. *Phys. Rev. A*, 82:033428, Sep 2010. doi: 10.1103/PhysRevA.82.033428. URL `http://link.aps.org/doi/10.1103/PhysRevA.82.033428`.

[202] Jorge Quintanilla, Sam T. Carr, and Joseph J. Betouras. Metanematic, smectic, and crystalline phases of dipolar fermions in an optical lattice. *Phys. Rev. A*, 79:031601, Mar 2009. doi: 10.1103/PhysRevA.79.031601. URL `http://link.aps.org/doi/10.1103/PhysRevA.79.031601`.

[203] Chungwei Lin, Erhai Zhao, and W. Vincent Liu. Liquid crystal phases of ultracold dipolar fermions on a lattice. *Phys. Rev. B*, 81:045115, Jan 2010. doi: 10.1103/PhysRevB.81.045115. URL `http://link.aps.org/doi/10.1103/PhysRevB.81.045115`.

[204] Emanuele G. Dalla Torre, Erez Berg, and Ehud Altman. Hidden Order in 1D Bose Insulators. *Phys. Rev. Lett.*, 97:260401, Dec 2006. doi: 10.1103/PhysRevLett.97.260401. URL `http://link.aps.org/doi/10.1103/PhysRevLett.97.260401`.

[205] J. P. Kestner, Bin Wang, Jay D. Sau, and S. Das Sarma. Prediction of a gapless topological Haldane liquid phase in a one-dimensional cold polar molecular lattice. *Phys. Rev. B*, 83:174409, May 2011. doi: 10.1103/PhysRevB.83.174409. URL `http://link.aps.org/doi/10.1103/PhysRevB.83.174409`.

[206] G. Pupillo, A. Micheli, H. P. Buchler, and P. Zoller. *Cold molecules: Creation and applications*, chapter 12. Taylor & Francis, 2009.

[207] A. V. Gorshkov, P. Rabl, G. Pupillo, A. Micheli, P. Zoller, M. D. Lukin, and H. P. Büchler. Suppression of inelastic collisions between polar molecules with a repulsive shield. *Phys. Rev. Lett.*, 101:073201, 2008.

[208] A. Micheli, G. Pupillo, H. P. Büchler, and P. Zoller. Cold polar molecules in two-dimensional traps: Tailoring interactions with external fields for novel quantum phases. *Phys. Rev. A*, 76:043604, 2007.

[209] G. Pupillo, A. Griessner, A. Micheli, M. Ortner, D.-W. Wang, and P. Zoller. Cold atoms and molecules in self-assembled dipolar lattices. *Phys. Rev. Lett.*, 100:050402, 2008.

[210] M Ortner, A Micheli, G Pupillo, and P Zoller. Quantum simulations of extended Hubbard models with dipolar crystals. *New J. Phys.*, 11:055045, 2009.

[211] H. P. Büchler, A Micheli, and P Zoller. Three-body interactions with cold polar molecules. *Nature Phys.*, 3:726–731, 2007.

[212] Kai P. Schmidt, Julien Dorier, and Andreas M. Läuchli. Solids and Supersolids of Three-Body Interacting Polar Molecules on an Optical Lattice. *Phys. Rev. Lett.*, 101:150405, Oct 2008. doi: 10.1103/PhysRevLett.101.150405. URL `http://link.aps.org/doi/10.1103/PhysRevLett.101.150405`.

[213] C. Trefzger, C. Menotti, and M. Lewenstein. Pair-Supersolid Phase in a Bilayer System of Dipolar Lattice Bosons. *Phys. Rev. Lett.*, 103:035304, Jul 2009. doi: 10.1103/PhysRevLett.103.035304. URL `http://link.aps.org/doi/10.1103/PhysRevLett.103.035304`.

[214] Andrew C. Potter, Erez Berg, Daw-Wei Wang, Bertrand I. Halperin, and Eugene Demler. Superfluidity and Dimerization in a Multilayered System of Fermionic Polar Molecules. *Phys. Rev. Lett.*, 105:220406, Nov 2010. doi: 10.1103/PhysRevLett.105.220406. URL `http://link.aps.org/doi/10.1103/PhysRevLett.105.220406`.

[215] Kazimierz Łakomy, Rejish Nath, and Luis Santos. Spontaneous crystallization and filamentation of solitons in dipolar condensates. *Phys. Rev. A*, 85:033618, Mar 2012. doi: 10.1103/PhysRevA.85.033618. URL `http://link.aps.org/doi/10.1103/PhysRevA.85.033618`.

[216] Ryan Barnett, Dmitry Petrov, Mikhail Lukin, and Eugene Demler. Quantum Magnetism with Multicomponent Dipolar Molecules in an Optical Lattice. *Phys. Rev. Lett.*, 96:190401, 2006.

[217] Eddy Timmermans, Paolo Tommasini, Mahir Hussein, and Arthur Kerman. Feshbach resonances in atomic Bose-Einstein condensates. *Physics Reports*, 315(1-3):199 – 230, 1999. ISSN 0370-1573. doi: DOI:10.1016/S0370-1573(99)00025-3. URL `http://www.sciencedirect.com/science/article/pii/S0370157399000253`.

[218] J. Ranninger and S. Robaszkiewicz. Superconductivity of locally paired electrons. *Physica B*, 135:468, 1985.

[219] R. Friedberg and T. D. Lee. Gap energy and long-range order in the boson-fermion model of superconductivity. *Phys. Rev. B*, 40(10):6745–6762, Oct 1989. doi: 10.1103/PhysRevB.40.6745.

[220] J. Ranninger and J. M. Robin. The boson-fermion model of high-T_c superconductivity doping dependence. *Physica C*, 253:279–291, 1995.

[221] Kai-Yu Yang, E. Kozik, Xin Wang, and M. Troyer. Diagrammatic quantum Monte Carlo solution of the two-dimensional cooperon-fermion model. *Phys. Rev. B*, 83:214516, Jun 2011. doi: 10.1103/PhysRevB.83.214516. URL `http://link.aps.org/doi/10.1103/PhysRevB.83.214516`.

[222] M. Holland, S. J. J. M. F. Kokkelmans, M. L. Chiofalo, and R. Walser. Resonance superfluidity in a quantum degenerate Fermi gas. *Phys. Rev. Lett.*, 87:120406–1–4, 2001.

[223] J. N. Milstein, S. J. J. M. F. Kokkelmans, and M. J. Holland. Resonance theory of the crossover from Bardeen-Cooper-Schrieffer superfluidity to Bose-Einstein condensation in a dilute Fermi gas. *Phys. Rev. A*, 66:043604–1–6, 2002.

[224] Roberto B. Diener and Tin-Lun Ho. Fermions in Optical Lattices Swept across Feshbach Resonances. *Phys. Rev. Lett.*, 96(1):010402, Jan 2006. doi: 10.1103/PhysRevLett.96.010402.

[225] D. B. M. Dickerscheid, U. Al Khawaja, D. van Oosten, and H. T. C. Stoof. Feshbach resonances in an optical lattice. *Phys. Rev. A*, 71:043604, Apr 2005. doi: 10.1103/PhysRevA.71.043604. URL `http://link.aps.org/doi/10.1103/PhysRevA.71.043604`.

[226] L.-M. Duan. Effective Hamiltonian for Fermions in an Optical Lattice across a Feshbach Resonance. *Phys. Rev. Lett.*, 95(24):243202, 2005.

[227] L.-M. Duan. General Hubbard model for strongly interacting fermions in an optical lattice and its phase detection. *EPL*, 81(2):20001, 2008. URL `http://stacks.iop.org/0295-5075/81/i=2/a=20001`.

[228] J. P. Kestner and L.-M. Duan. Effective single-band models for strongly interacting fermions in an optical lattice. *Phys. Rev. A*, 81(4):043618, Apr 2010. doi: 10.1103/PhysRevA.81.043618.

[229] Hans Peter Büchler. Microscopic derivation of Hubbard parameters for cold atomic gases. *Phys. Rev. Lett.*, 104(9):090402, Mar 2010. doi: 10.1103/PhysRevLett.104.090402.

[230] Javier von Stecher, Victor Gurarie, Leo Radzihovsky, and Ana Maria Rey. Lattice-Induced Resonances in One-Dimensional Bosonic Systems. *Phys. Rev. Lett.*, 106(23):235301, Jun 2011. doi: 10.1103/PhysRevLett.106.235301.

[231] Algorithms and libraries for physics simulations: `http://alps.comp-phys.org`.

[232] A. F. Albuquerque, F. Alet, P. Corboz, P. Dayal, A. Feiguin, S. Fuchs, L. Gamper, E. Gull, S. Gurtler, A. Honecker, R. Igarashi, M. Korner, A. Kozhevnikov, A. Lauchli, S. R. Manmana, M. Matsumoto, I. P. McCulloch, F. Michel, R. M. Noack, G. Pawlowski, L. Pollet, T. Pruschke, U. Schollwöck, S. Todo, S. Trebst, M. Troyer, P. Werner, and S. Wessel. The ALPS project release 1.3: Open-source software for strongly correlated systems. *Journal of Magnetism and Magnetic Materials*, 310:1187–1193, 2007.

[233] B. Bauer, L. D. Carr, H. G. Evertz, A. Feiguin, J. Freire, S. Fuchs, L. Gamper, J. Gukelberger, E. Gull, S. Guertler, A. Hehn, R. Igarashi, S. V. Isakov, D. Koop, P. N. Ma, P. Mates, H. Matsuo, O. Parcollet, G. Pawłowski, J. D. Picon, L. Pollet, E. Santos, V. W. Scarola, U. Schollwöck, C. Silva, B. Surer, S. Todo, S. Trebst, M. Troyer, M. L. Wall, P. Werner, and S. Wessel. The ALPS project release 2.0: open source software for strongly correlated systems. *Journal of Statistical Mechanics: Theory and Experiment*, 2011(05):P05001, 2011. URL `http://stacks.iop.org/1742-5468/2011/i=05/a=P05001`.

[234] Powder with power open source time-dependent DMRG code: http://www.qti.sns.it/dmrg/phome.html.

[235] DMRG++ open source DMRG code: http://www.ornl.gov/~gz1/dmrgPlusPlus/.

[236] Open source time-evolving block decimation. http://physics.mines.edu/downloads/software/tebd/.

[237] Tilman Esslinger. Fermi-Hubbard Physics with Atoms in an Optical Lattice. *Annual Review of Condensed Matter Physics*, 1(1):129–152, 2010. doi: 10.1146/annurev-conmatphys-070909-104059. URL http://www.annualreviews.org/doi/abs/10.1146/annurev-conmatphys-070909-104059.

CHAPTER 2

MODELS FOR STRONGLY CORRELATED LATTICE PHYSICS

In this chapter we outline the general procedure of deriving an effective low-energy many-body lattice model appropriate for strong correlations from few-body physics. The replacement of the full many-body Hamiltonian with an effective model can be justified as the result of a renormalization group iteration [1]. Using a renormalization group analysis, a Hamiltonian can be projected into a suitably chosen low-energy subspace and the high-energy modes integrated out, resulting in an effective model for only the low-energy modes. Under this renormalization procedure, certain *irrelevant* interactions are suppressed relative to those which grow or stay constant.[11] The hope is that the renormalized theory contains fewer relevant single particle states or simpler interactions. Additionally, rather than integrating out the high-energy degrees of freedom, one can also *dress* them with interactions such that the new dressed high-energy degrees of freedom have a more easily understood structure or simpler couplings to the relevant low-energy degrees of freedom. This dressing is the main idea underlying Feynman diagram re-summation techniques such as the random phase approximation [2], and is also the key idea in the derivation of the Fermi resonance Hamiltonian discussed in Part III.

Generally, a renormalization group analysis of an interacting many-body system is difficult. For particles in deep lattices,[12] the presence of a band gap provides an energy scale which naturally separates the low-energy and high-energy degrees of freedom. If all other energy scales, e.g., temperature, interactions, etc., are smaller than this band gap, then the particles will remain in the lowest Bloch band, drastically simplifying

---

[11] Interactions that grow and remain constant are called *relevant* and *marginal*, respectively.

[12] Deep in this context means that the strength of the lattice is comparable to or larger than the kinetic energy. For ultracold gases with lattice potential $V(x) = V_0 \sin^2(\pi x/a)$, a lattice may be considered deep in typical cases for $V_0/E_R \geq 2$, with $E_R = \hbar^2\pi^2/2ma^2$ the recoil energy.

the structure of the theory without the need for a more complex renormalization group analysis. Lattice Hamiltonians obtained under these assumptions are known as *Hubbard models* [3, 4], in analogy with the lattice model proposed by Hubbard

$$\hat{H}=-t\sum_{\langle i,j\rangle}\sum_{\sigma\in\{\uparrow,\downarrow\}}\left[\hat{a}_{i\sigma}^{\dagger}\hat{a}_{j\sigma}+\text{h.c.}\right]+U\sum_{i}\hat{n}_{i\uparrow}\hat{n}_{i\downarrow}\,. \tag{2.1}$$

The first term represents tunneling of fermions with spin projection σ between nearest-neighbor lattice sites i and j with associated energy t. The latter term represents interactions between spin-up and spin-down fermions which occur only when two particles occupy the same lattice site. For electrons in solids which interact via a screened Coulomb potential, the reduction of the interactions to a single on-site term is a drastic oversimplification, and so the Hubbard model represents a toy model which is not expected to capture the microscopic physics.[13] Additionally, the restriction to only the lowest Bloch band of the lattice does not hold for f-electron metals in which the interaction parameters are generally greater than the band splitting. However, for neutral ultracold gases in optical lattices, the interactions can be well-modeled by a contact pseudopotential in many cases, and so Eq. (2.1) may represent an accurate microscopic many-body Hamiltonian. The remaining issue is to identify the parameters t and U from few-body physics.

2.1 The Single-Particle Problem: Bloch States and Wannier Functions

Let us consider a general many-body Hamiltonian consisting of a single-particle Hamiltonian $\hat{H}_1$ and two-body interactions $\hat{H}_2$, written in second quantization as

$$\hat{H}=\int d\mathbf{r}\hat{\psi}^{\dagger}\left(\mathbf{r}\right)\hat{H}_1\hat{\psi}\left(\mathbf{r}\right)+\frac{1}{2}\int d\mathbf{r}\int d\mathbf{r}'\hat{\psi}^{\dagger}\left(\mathbf{r}\right)\hat{\psi}^{\dagger}\left(\mathbf{r}'\right)\hat{H}_2\hat{\psi}\left(\mathbf{r}'\right)\hat{\psi}\left(\mathbf{r}\right)\,. \tag{2.2}$$

For systems which include a lattice potential, we may write $\hat{H}_1$ as

---

[13] For a detailed exposition of the assumptions leading to the Hubbard model in solid state systems, see Ref. [4]

$$\hat{H}_1 = \hat{H}_{\rm kin} + \hat{H}_{\rm latt} + \hat{H}_{\rm internal}\,, \tag{2.3}$$

where $\hat{H}_{\rm kin}$ is the kinetic energy operator, $\hat{H}_{\rm latt}$ is the coupling of a particle to the lattice, and $\hat{H}_{\rm internal}$ is the Hamiltonian describing the internal degrees of freedom in free space. We furthermore assume that all terms appearing in $\hat{H}_{\rm internal}$ have no spatial dependence on the lattice scale. When this is the case then the basis diagonalizing this Hamiltonian is spatially independent, depending only on the internal degrees of freedom of a single particle in free space. We will refer to the basis diagonalizing this Hamiltonian as $|\sigma\rangle$: $\hat{H}_{\rm internal}|\sigma\rangle = E^{\sigma}_{\rm internal}|\sigma\rangle$.

The coupling to the lattice is provided by the dynamical polarizability tensor $\tilde{\alpha}$ of the particle in question [5]. This tensor has nine operator-valued elements which can be indexed by pairs (p, p') with p and p' running over a space fixed basis $\{x, y, z\}$.[14] Using the states $|\sigma\rangle$ we can find a representation of $\tilde{\alpha}$ within the eigenspace of $\hat{H}_{\rm internal}$. We assume that the optical lattice consists of three independent retro-reflected[15] laser beams, each monochromatic with linear frequency ν, arranged in a simple cubic structure:

$$\mathbf{E}_{\rm opt}\left(\mathbf{r}, t\right) = 2e^{2\pi i\nu t}\left[E_x\boldsymbol{\epsilon}^x \sin\left(k_l x\right) + E_y\boldsymbol{\epsilon}^y \sin\left(k_l y\right) + E_z\boldsymbol{\epsilon}^z \sin\left(k_l z\right)\right]\,. \tag{2.4}$$

Here $k_l = \pi/a$ with $a = \lambda/2$ the lattice spacing and λ the wavelength of the optical field. In addition to the intensities $I_x = E_x^2$ etc. along each spatial direction, this optical field is also described by three complex vectors $\boldsymbol{\epsilon}^\nu$, $\nu = x, y, z$ giving the polarization of the x, y, and z fields, respectively. If we use the spherical representation of the polarization vectors $\boldsymbol{\epsilon}^\nu$ we have the Stark shift

---

[14] Equivalently, a spherical basis $\{-1, 0, 1\}$.

[15] This accounts for the factors of 2 in front of the field strengths.

$$V_{\sigma'\sigma}\left(\mathbf{r}\right) \equiv \langle\sigma'|\hat{H}_{\mathrm{latt}}\left(\mathbf{r}\right)|\sigma\rangle = -\sum_{\nu=x,y,z}\sum_{q_{\nu}q_{\nu'}}\left|E_{\nu}\right|^{2}\epsilon^{\nu\star}_{q_{\nu}}\langle\sigma'|\tilde{\alpha}_{q_{\nu}q'_{\nu}}|\sigma\rangle\epsilon^{\nu}_{q'_{\nu}}\sin^{2}\left(k_{l}x_{\nu}\right)\,. \tag{2.5}$$

This shift is what provides the lattice potential for ultracold gases.

For anisotropic systems such as rotating molecules, Eq. (2.5) is not generally diagonal in the internal degrees of freedom. This is discussed for rotational eigenstates in Chapter 3, and will be discussed in full generality in a forthcoming paper [6]. In this case, the problem is complicated by the fact that the three independent beams forming $\mathbf{E}_{\mathrm{opt}}$ generally compete for ordering of the internal state $|\sigma\rangle$. Thus, the lattice formed from three independent beams becomes non-separable due to the fact that the beams couple differently to the internal state. For simplicity, in the remainder of the present discussion we will assume that the matrix Eq. (2.5) is diagonal. In this case two simplifications occur. The first is that each $|\sigma\rangle$ obeys a Schrödinger equation which is decoupled from all other internal states. The second is that the lattice is separable in real space, and so we can solve each Cartesian direction separately.

As the single-particle Hamiltonian Eq. (2.3) is invariant under translations by a Bravais lattice vector, its eigenfunctions can be written in Bloch form [7]. We thus write the solutions of $\hat{H}_1$ in the form $\psi_{\sigma\mathbf{n}\mathbf{q}}\left(\mathbf{r}\right)$. Here, the quasimomentum $\mathbf{q}$ whose components q_{σ} lie within the first Brillouin zone (BZ) $[-k_l, k_l)$ denote how this eigenfunction transforms under translations, and the band index $\mathbf{n}$ distinguishes solutions with the same translational symmetry which differ in energy. Due to the separability of the lattice potential, we may write this solution as a product

$$\psi_{\sigma\mathbf{n}\mathbf{q}}\left(\mathbf{r}\right) = \prod_{\nu=x,y,z}\psi_{\sigma n_{\nu}q_{\nu}}\left(r_{\nu}\right)\,, \tag{2.6}$$

where the $\psi_{\sigma n_{\nu}q_{\nu}}\left(r_{\nu}\right)$ satisfy the 1D Schrödinger equations

$$\left[\frac{\hat{p}_{\nu}^{2}}{2m}+V_{\sigma\sigma}\left(r_{\nu}\right)\right]\psi_{\sigma n_{\nu}q_{\nu}}\left(x_{\nu}\right) = E_{\sigma q_{\nu}n_{\nu}}\psi_{\sigma n_{\nu}q_{\nu}}\left(r_{\nu}\right)\,. \tag{2.7}$$

Here and throughout, we will use boldface to denote three-component vectors and ordinary type to denote scalar quantities, e. g. $\mathbf{q}$ refers to a 3D quasimomentum and q a 1D quasimomentum. Also, we will leave off the ν subscripts when they are unnecessary, writing instead e. g. $\psi_{\sigma n q}\left(x\right)$ for a 1D Bloch function.

We now turn to the solution of the equations governing the 1D Bloch functions Eq. (2.7). As is well known [7], Bloch functions may be written in the form

$$\psi_{\sigma n q}\left(x\right)=\frac{1}{\sqrt{La}}e^{iqx}u_{\sigma n q}\left(x\right)\,, \tag{2.8}$$

where L is the number of (1D) unit cells and $u_{\sigma n q}\left(x\right)$ has the same periodicity as the potential. Because of this periodicity, these functions may be expanded in a Fourier series

$$u_{\sigma n q}\left(x\right)=\lim_{\ell\to\infty}\sum_{p=-\ell}^{\ell}c^{p}_{\sigma n q}e^{2\pi p i x/a}\,. \tag{2.9}$$

Here ℓ is a finite Fourier cutoff used in numerics. A cutoff of a few tens captures the lowest few bands to machine precision. Inserting this expansion into the 1D Schrödinger equation yields the finite eigenvalue equation for the coefficients $c^{p}_{\sigma n q}$:

$$\lim_{\ell\to\infty}\sum_{p'=-\ell}^{\ell}H_{pp'}c^{p'}_{\sigma n q}=E_{\sigma n q}c^{p}_{\sigma n q}\,, \tag{2.10}$$

$$H_{pp'}=\left[\left(2p+q/k_l\right)^2E_R+\frac{V}{2}\right]\delta_{pp'}-\frac{V}{4}\left(\delta_{p,p'+1}+\delta_{p,p'-1}\right)\,, \tag{2.11}$$

where we have defined the recoil energy $E_R\equiv\hbar^2k_l^2/2m$ and V is the coefficient of $\sin^2\left(k_l\nu\right)$ along the particular 1D direction. Numerically, this amounts to solving a real symmetric tridiagonal eigenproblem.

Bloch functions represent simultaneous eigenfunctions of translation and the single-particle Hamiltonian, and are hence highly delocalized. Typical interactions in strongly correlated systems, on the other hand, are often spatially local and so Bloch functions represent an inadequate basis for describing strong interactions. A more suitable basis is provided by Wannier functions, which are the quasimomentum Fourier transform conjugates to the Bloch functions

$$w_{\sigma\mathbf{n}}\left(\mathbf{r}-\mathbf{r}_i\right)=\frac{1}{\sqrt{L}}\sum_{\mathbf{q}\in\mathrm{BZ}}e^{-i\mathbf{q}\cdot\mathbf{r}_i}\psi_{\sigma n\mathbf{q}}\left(\mathbf{r}\right)\,, \tag{2.12}$$

where $\mathbf{r}_i$ represents the position of site i in the lattice. These functions are highly localized,[16] and represent a more appropriate basis in which to expand strongly interacting models. As is apparent from the notation of Eq. (2.12), a Wannier function centered around a particular lattice site $\mathbf{r}_i$ depends only on the distance from that site. We will also use the shorthand $w_{\sigma\mathbf{n}i}\left(\mathbf{r}\right)\equiv w_{\sigma\mathbf{n}}\left(\mathbf{r}-\mathbf{r}_i\right)$ to simplify some expressions.

The general route to a Hubbard model is to expand the field operator $\hat{\psi}\left(\mathbf{r}\right)$ in terms of the single-particle Wannier basis, resulting in a Hamiltonian of the form

$$\begin{aligned}\hat{H}=&\sum_{i,j}\sum_{\sigma n}\hat{a}^{\dagger}_{i\sigma\mathbf{n}}\langle i\sigma\mathbf{n}|\hat{H}_1|j\sigma\mathbf{n}\rangle\hat{a}_{j\sigma\mathbf{n}} \\ &+\frac{1}{2}\sum_{i_1i_2i'_1i'_2}\sum_{\mathbf{n}_1\mathbf{n}_2\mathbf{n}'_1\mathbf{n}'_2}\sum_{\sigma_1\sigma_2\sigma'_1\sigma'_2}\langle i_1\sigma_1\mathbf{n}_1;i_2\sigma_2\mathbf{n}_2|\hat{H}_2|i'_1\sigma'_1\mathbf{n}'_1;i'_2\sigma'_2\mathbf{n}'_2\rangle \\ &\times\hat{a}^{\dagger}_{i_1\sigma_i\mathbf{n}_1}\hat{a}^{\dagger}_{i_2\sigma_2\mathbf{n}_2}\hat{a}_{i'_2\sigma'_2\mathbf{n}'_2}\hat{a}_{i'_1\sigma'_1\mathbf{n}'_1}\,,\end{aligned} \tag{2.13}$$

where $\hat{a}^{\dagger}_{i\sigma\mathbf{n}}$ creates a particle in Wannier state $w_{\sigma\mathbf{n}}\left(\mathbf{r}-\mathbf{r}_i\right)$. The overlap integrals $\langle i\sigma\mathbf{n}|\hat{H}_1|j\sigma\mathbf{n}\rangle$ and $\langle i_1\sigma_1\mathbf{n}_1;i_2\sigma_2\mathbf{n}_2|\hat{H}_2|i'_1\sigma'_1\mathbf{n}'_1;i'_2\sigma'_2\mathbf{n}'_2\rangle$ are called *Hubbard parameters*, and are the point of contact between the microscopic, few-body physics and the many-body physics. In order for this transformation to be useful, we must truncate the sums appearing in Eq. (2.13) in some form. Often, the sums over the band indices $\mathbf{n}$ and internal states σ are set by selection rules and energetics from the few-body physics. For example, for transitions in an AC microwave field with circular polarization q, only

---

[16] More precisely, these functions can be made exponentially localized when subject to a one-dimensional inversion symmetric potential by an appropriate choice of phases on the Bloch functions [8]. The Wannier functions produced by this procedure are called maximally localized Wannier functions, and are used exclusively in this thesis unless indicated otherwise.

internal states for which $\langle \sigma' | \hat{d}_q | \sigma \rangle$ is nonzero will contribute, with d_q the projection of the dipole operator along the space-fixed spherical basis direction q. Similarly, interactions which transfer particles between bands on-site must preserve the parity of the bands, that is $(-1)^{n_1+n_2} = (-1)^{n_1'+n_2'}$. Additionally, it is most often assumed that the temperature, interaction scales, and all other scales of the problem are small compared to the band gap between the lowest two bands. This restricts all particles to remain in the lowest band, and so the summations over band indices vanish. To avoid overly cumbersome notation, we will leave off band indices when we discuss the two-particle interactions. We will also discuss the truncation of spatial sums separately for single-particle and short and long ranged two particle interactions.

2.1.1 Single-Particle Hubbard Parameters

The Hubbard parameters arising from the single-particle piece take the form

$$t_{\sigma\mathbf{n}}^{i,j} \equiv -\int d\mathbf{r} w_{\sigma\mathbf{n}}^{\star}\left(\mathbf{r}-\mathbf{r}_i\right)\left[\hat{H}_{\text{kin}} + \hat{H}_{\text{internal}} + \hat{H}_{\text{latt}}\left(\mathbf{r}\right)\right] w_{\sigma\mathbf{n}}\left(\mathbf{r}-\mathbf{r}_j\right) \tag{2.14}$$

$$= -\frac{1}{L^3}\sum_{\mathbf{q}\mathbf{q}'}\int d\mathbf{r} e^{i\mathbf{q}\cdot\mathbf{r}_i}\psi_{\sigma\mathbf{n}\mathbf{q}}^{\star}\left(\mathbf{r}\right)\left[\hat{H}_{\text{kin}} + \hat{H}_{\text{internal}} + \hat{H}_{\text{latt}}\left(\mathbf{r}\right)\right] e^{i\mathbf{q}'\cdot\mathbf{r}_j}\psi_{\sigma\mathbf{n}\mathbf{q}'}\left(\mathbf{r}\right) \tag{2.15}$$

$$= -\frac{1}{L^3}\sum_{\mathbf{q}\mathbf{q}'} e^{i\left(\mathbf{q}\cdot\mathbf{r}_i - \mathbf{q}'\cdot\mathbf{r}_j\right)} E_{\sigma\mathbf{n}\mathbf{q}}\delta_{\mathbf{q}\mathbf{q}'} \tag{2.16}$$

$$= -\frac{1}{L^3}\sum_{\mathbf{q}} e^{i\mathbf{q}\left(\mathbf{r}_i-\mathbf{r}_j\right)} E_{\sigma\mathbf{n}\mathbf{q}}\,. \tag{2.17}$$

The third line used the fact that the Bloch functions diagonalize the single-particle Hamiltonian and the orthonormality of the Bloch functions. This orthogonality is also why the parameters $t_{\sigma\mathbf{n}}^{i,j}$ are diagonal in σ and $\mathbf{n}$. The parameters $t_{\sigma\mathbf{n}}^{i,j}$ with $i \neq j$ are called *tunneling* or *hopping* parameters, as they are interpreted as a particle quantum mechanically tunneling through the lattice from site i to site j. These parameters take the simple form of a quasimomentum Fourier transform of the band structure in the relative coordinate. Their contribution to the Hamiltonian is

$$\hat{H}_{\text{tunneling}} = -\sum_{\sigma\mathbf{n}} \sum_{i\neq j} t^{i,j}_{\sigma\mathbf{n}} \hat{a}^{\dagger}_{i\sigma\mathbf{n}} \hat{a}_{j\sigma\mathbf{n}} \,. \tag{2.18}$$

The minus sign in this definition is customary and accounts for the minus sign in the definition of $t^{i,j}_{\sigma\mathbf{n}}$. It arises because the nearest-neighbor tunneling coefficient defined as such is positive. For $i = j$, $t^{i,i}_{\sigma\mathbf{n}}$ is minus the energy of a particle in band $\mathbf{n}$ and internal state σ, and so it is customary to reverse the sign and define

$$E_{\sigma\mathbf{n}} = \frac{1}{L^3} \sum_{\mathbf{q}} E_{\sigma\mathbf{n}\mathbf{q}} \,. \tag{2.19}$$

This parameter is simply the average of the band structure for fixed band number $\mathbf{n}$ and internal state σ. The complete single-particle Hubbard Hamiltonian is thus

$$\hat{H}_{\text{Hsp}} = \sum_{\sigma\mathbf{n}} E_{\sigma\mathbf{n}} \sum_{i} \hat{n}_{i\sigma\mathbf{n}} - \sum_{\sigma\mathbf{n}} \sum_{i\neq j} t^{i,j}_{\sigma\mathbf{n}} \hat{a}^{\dagger}_{i\sigma\mathbf{n}} \hat{a}_{j\sigma\mathbf{n}} \,, \tag{2.20}$$

where $\hat{n}_{i\sigma\mathbf{n}} \equiv \hat{a}^{\dagger}_{i\sigma\mathbf{n}} \hat{a}_{i\sigma\mathbf{n}}$ is the number operator on site i for band $\mathbf{n}$ and internal state σ. Noting that $\hat{a}^{\dagger}_{j\sigma\mathbf{n}} \hat{a}_{i\sigma\mathbf{n}}$ is the Hermitian conjugate of $\hat{a}^{\dagger}_{i\sigma\mathbf{n}} \hat{a}_{j\sigma\mathbf{n}}$, we must have $t^{j,i}_{\sigma\mathbf{n}} = t^{i,j}_{\sigma\mathbf{n}}$ for a Hermitian Hamiltonian.[17] This allows us to write

$$\hat{H}_{\text{Hsp}} = \sum_{\sigma\mathbf{n}} E_{\sigma\mathbf{n}} \sum_{i} \hat{n}_{i\sigma\mathbf{n}} - \sum_{\sigma\mathbf{n}} \sum_{j>i} t^{i,j}_{\sigma\mathbf{n}} \left[\hat{a}^{\dagger}_{i\sigma\mathbf{n}} \hat{a}_{j\sigma\mathbf{n}} + \text{h.c.} \right] \,. \tag{2.21}$$

For a 1D lattice of the form $V(x) = V_0 \sin^2(\pi x/a)$, we can parameterize the dependence of the lowest band tunneling on the lattice height V_0 as [9]

$$t^{ij}/E_R = A \left(\frac{V_0}{E_R} \right)^B \exp\left(-C\sqrt{V_0/E_R} \right) \,, \tag{2.22}$$

which is motivated by the large V_0/E_R limit of the tunneling computed using Mathieu functions [10]. A fit to numerically generated data is found to be quantitatively valid

---

[17] Strictly speaking, this implies that $t^{j,i}_{\sigma\mathbf{n}} = t^{i,j\,\star}_{\sigma\mathbf{n}}$, but for time-reversal invariant single-particle Hamiltonians we can always choose the eigenvalues, and thus the tunnelings, to be real.

in the range $V_0/E_R > 2$. The results are summarized in Table 2.1. These expressions can be used to determine a consistent order of approximation when truncating long-range interactions.

Table 2.1: Fit parameters for the tunneling at the nearest, next-nearest, and third nearest neighbor distances together with their asymptotic standard errors.

Term	A	B	C
$t^{i,i+1}$	1.363±0.0004	1.057±0.0005	2.117±0.0004
$-t^{i,i+2}$	2.491±0.0023	1.957±0.0011	4.361±0.0012
$t^{i,i+3}$	7.294±0.011	2.767±0.0015	6.534±0.0018

2.2 Two-Particle Hubbard Parameters

We now turn to determining the Hubbard parameters for the two-particle interaction. Interactions can be broadly classified into short range and long range terms, where short range terms are those which are well modeled by a pseudopotential of the form $U(\mathbf{r}) = g\delta(\mathbf{r})$ and long-range terms have nonlocal $\mathbf{r}$ dependence. We begin with an exposition of the short range terms.

2.2.1 Short-Range Interactions

The quantum-mechanical scattering from a spherically symmetric potential can be codified in terms of partial waves ℓ, where ℓ is the relative orbital angular momentum of the scattering particles. At low energies, the dominant partial waves are the s- and p-waves, corresponding to $\ell = 0$ and $\ell = 1$, respectively. For short range potentials, the contributions from these partial waves at low energies can be captured by pseudopotentials that depend only on a single parameter, the (s- or p-wave) *scattering length*. This is a readily calculated and measured quantity. The regularized s-wave pseudopotential is [11, 12]

$$U(\mathbf{r}) = \frac{4\pi\hbar^2 a_s}{m}\delta(\mathbf{r})\,\partial_r r\,, \tag{2.23}$$

where a_s is the s-wave scattering length. The term $\partial_r r$ is a regularization operator which incorporates boundary conditions on the spherically symmetric scattering problem, but is inconvenient for the lattice problem which is not naturally stated in spherical coordinates. In Cartesian coordinates, which are most natural for our simple cubic lattice, we have

$$U(\mathbf{r}) = \frac{4\pi\hbar^2 a_s}{m}\delta(x)\,\delta(y)\,\delta(z)\left[(x\partial_x + y\partial_y + z\partial_z) + 1\right] . \tag{2.24}$$

Thus, provided that our functions have sufficiently regular derivatives at $r \to 0$, the first term in brackets vanishes in this same limit. The regularity of the Wannier functions is assured by the fact that they are band-limited periodic functions, and so the pseudopotential becomes

$$U(\mathbf{r}) = \frac{4\pi\hbar^2 a_s}{m}\delta(\mathbf{r}) . \tag{2.25}$$

That is, we can neglect the regularization operator.

Using this pseudopotential in the second-quantized many-body Hamiltonian and expanding in terms of lowest band Wannier functions, we have

$$\begin{aligned}\hat{H}_{\mathrm{s-wave}} =& \frac{2\pi\hbar^2}{m}\int d\mathbf{r}\int d\mathbf{r}'\hat{\psi}^\dagger(\mathbf{r})\,\hat{\psi}^\dagger(\mathbf{r}')\,a_s\delta(\mathbf{r}-\mathbf{r}')\,\hat{\psi}(\mathbf{r}')\,\hat{\psi}(\mathbf{r}) && (2.26)\\ =& \frac{2\pi\hbar^2}{m}\sum_{i_1 i_2 i'_1 i'_2}\sum_{\sigma_1\sigma_2\sigma'_1\sigma'_2} a_s^{\sigma'_1\sigma'_2}\hat{a}^\dagger_{i_1\sigma_1}\hat{a}^\dagger_{i_2\sigma'_2}\hat{a}_{i'_2\sigma'_2}\hat{a}_{i'_1\sigma'_1}\\ &\times\int d\mathbf{r}w^\star_{i_1\sigma_1}(\mathbf{r})\,w^\star_{i_2\sigma_2}(\mathbf{r})\,w_{i'_2\sigma'_2}(\mathbf{r})\,w_{i'_1\sigma_1}(\mathbf{r}) , && (2.27)\end{aligned}$$

where $a_s^{\sigma\sigma'}$ is now the s-wave scattering length for particles in internal states σ and σ'. Note that this does not depend on the band or site indices, as it is a quantity calculated in free 3D space. All of the lattice physics is encapsulated in the Wannier overlap integral, which is essentially a geometrical factor. Each term in the summation can be interpreted as a scattering process where particles in states σ'_1 and σ'_2 at positions

i'_1 and i'_2 scatter via the s-wave pseudopotential into internal states σ_1 and σ_2 at positions i_1 and i_2, respectively. We will call the primed indices incoming and the unprimed indices outgoing. Because of the localization of the Wannier functions, terms with $i_1 \neq i_2 \neq i'_1 \neq i'_2$ are exponentially suppressed, and so the most common approximation is to keep only the $i_1 = i_2 = i'_1 = i'_2$ term. In the case of a single internal state, this leads to

$$\hat{H}_{\mathrm{s-wave}} = \frac{2\pi\hbar^2 a_s}{m}\int d\mathbf{r}\,|w_0(\mathbf{r})|^4 \sum_i \hat{n}_i(\hat{n}_i - 1)\ , \tag{2.28}$$

from which we can immediately read off the value of U appearing in the Bose-Hubbard model Eq. (1.2). It is important to note that the lattice affects the strength of the interactions through the overlap of the Wannier functions. Hence, for a fixed scattering length the interactions can be increased by making the lattice deeper. By replacing a single lattice site with a harmonic oscillator with oscillator length $a_{\mathrm{h.o.}} = a/\pi(V_0/E_R)^{1/4}$ and thus approximating the Wannier functions with harmonic oscillator eigenstates,[18] we find that these integrals all grow as roughly $(V_0/E_R)^{1/4}$ with the lattice depth V_0. Finally, in the case where the lattice potential is separable, the Wannier function overlaps also separate into three independent 1D integrals, and so can be computed numerically with high efficiency.

The situation is similar for p-wave interactions. The regularized p-wave pseudopotential is [12]

$$U(\mathbf{r}) = \frac{\pi\hbar^2 a_1^3}{\mu}\overleftarrow{\nabla}\delta(\mathbf{r})\overrightarrow{\nabla} r\partial_{rrr}r^2\,, \tag{2.29}$$

where the arrows indicate the action of the gradient operators. Using the Leibnitz formula we have

---

[18] The on-site interaction coefficient for delta-function interactions in the deep lattice limit is the only place that this approximation has any credence. For off-site interactions or tunneling parameters, the use of the approximation leads not only to quantitative errors of an order of magnitude or more, but also qualitative errors in the symmetries of the Hubbard parameters, see Chapter 5.

$$\partial_{rrr}\left(r^2 f\right) = \left(6\partial_r + 6r\partial_{rr} + r^2\partial_{rrr}\right) f\,, \tag{2.30}$$

and so the only term which possibly doesn't vanish as $r \to 0$ for sufficiently regular functions f is the first. We will denote all such terms which vanish as $r \to 0$ as van.terms. We have

$$\begin{aligned} \overrightarrow{\nabla} r\partial_{rrr}\left(r^2 f\right) &= 6\overrightarrow{\nabla}\left[x\partial_x + y\partial_y + z\partial_z\right] f + \text{van.terms}\,, && (2.31)\\ &= 6\overrightarrow{\nabla} f + \text{van.terms}\,, && (2.32) \end{aligned}$$

and so the regularized potential for our sufficiently regular Wannier functions becomes

$$U\left(\mathbf{r}\right) = \frac{6\pi\hbar^2 a_1^3}{\mu}\overleftarrow{\nabla}\delta\left(\mathbf{r}\right)\overrightarrow{\nabla}\,. \tag{2.33}$$

Expanding the second quantized Hamiltonian as above, we have

$$\begin{aligned} \hat{H}_{\mathrm{p-wave}} &= \frac{3\pi\hbar^2}{\mu}\int d\mathbf{r}\int d\mathbf{r}'\hat{\psi}^{\dagger}\left(\mathbf{r}\right)\hat{\psi}^{\dagger}\left(\mathbf{r}'\right) a_1^3\overleftarrow{\nabla}\delta\left(\mathbf{r}-\mathbf{r}'\right)\overrightarrow{\nabla}\hat{\psi}\left(\mathbf{r}'\right)\hat{\psi}\left(\mathbf{r}\right)\,, && (2.34)\\ &= \frac{3\pi\hbar^2}{\mu}\sum_{i_1 i_2 i_2' i_1'}\sum_{\sigma_1\sigma_2\sigma_2'\sigma_1'}\left(a_1^{\sigma_1'\sigma_2'}\right)^3 \hat{a}^{\dagger}_{i_1\sigma_1}\hat{a}^{\dagger}_{i_2\sigma_2}\hat{a}_{i_2'\sigma_2'}\hat{a}_{i_1'\sigma_1'}\int d\mathbf{r}\\ &\times\int d\mathbf{r}' w^{\star}_{i_1\sigma_1}\left(\mathbf{r}\right) w^{\star}_{i_2\sigma_2}\left(\mathbf{r}'\right)\overleftarrow{\nabla}_{\mathbf{r}-\mathbf{r}'}\delta\left(\mathbf{r}-\mathbf{r}'\right)\overrightarrow{\nabla}_{\mathbf{r}-\mathbf{r}'} w_{i_2'\sigma_2'}\left(\mathbf{r}'\right) w_{i_1'\sigma_1'}\left(\mathbf{r}\right)\,, && (2.35)\\ &= \frac{6\pi\hbar^2}{m}\sum_{i_1 i_2 i_2' i_1'}\sum_{\sigma_1\sigma_2\sigma_2'\sigma_1'}\left(a_1^{\sigma_1'\sigma_2'}\right)^3 \hat{a}^{\dagger}_{i_1\sigma_1}\hat{a}^{\dagger}_{i_2\sigma_2}\hat{a}_{i_2'\sigma_2'}\hat{a}_{i_1\sigma_1'}\int d\mathbf{r}\\ &\times\int d\mathbf{r}\left[\left(\nabla w^{\star}_{i_1\sigma_1}\left(\mathbf{r}\right)\right) w^{\star}_{i_2\sigma_2}\left(\mathbf{r}\right) - w^{\star}_{i_1\sigma_1}\left(\mathbf{r}\right)\left(\nabla w^{\star}_{i_2\sigma_2}\left(\mathbf{r}\right)\right)\right]\\ &\times\left[\left(\nabla w_{i_1'\sigma_1'}\left(\mathbf{r}\right)\right) w_{i_2'\sigma_2'}\left(\mathbf{r}\right) - w_{i_1'\sigma_1'}\left(\mathbf{r}\right)\left(\nabla w_{i_2'\sigma_2'}\left(\mathbf{r}\right)\right)\right]\,. && (2.36) \end{aligned}$$

Here $a_1^{\sigma\sigma'}$ is the p-wave scattering length for internal states σ and σ'. We note that the derivatives can be rigorously taken, i.e., without any discretization error, using the Fourier representation of the Bloch functions, Eq. (2.9). Also, note that this integral vanishes identically if all of the incoming or outgoing indices are the same. That is to say, there is no on-site p-wave interaction between identical particles.[19]

2.2.2 Long-Range Interactions

For alkali atoms, the dominant interaction is provided by a $1/R^6$ dispersive potential [13] and hence interactions are well modeled by a short-range pseudopotential. We now turn our attention to long-ranged interactions arising from the dipole-dipole interaction Eq. (1.8) which are relevant for molecules. We can recast the interaction Eq. (1.8) as the contraction of two rank-two spherical tensors as

$$\hat{H}_{\mathrm{DD}}\left(\mathbf{R}\right) = -\frac{\sqrt{6}}{R^3}C^{(2)}\left(\mathbf{R}\right)\cdot\left[\hat{\mathbf{d}}_1\otimes\hat{\mathbf{d}}_2\right]^{(2)}\,, \tag{2.37}$$

where

$$\left[\hat{\mathbf{d}}_1\otimes\hat{\mathbf{d}}_2\right]^{(2)}_q = \sum_m \langle 1,m,1,q-m|2,q\rangle\left(d_1\right)_m\left(d_2\right)_{q-m}\,, \tag{2.38}$$

$\langle j_1m_1j_2m_2|jm\rangle$ is a Clebsch-Gordan coefficient, $\left(d_j\right)_m$ represents the m^{th} component of the j^{th} dipole in a space-fixed spherical basis, and $C^{(2)}_m\left(\mathbf{R}\right)$ is an unnormalized spherical harmonic in the polar coordinates defined by the relative coordinate. We compute Hubbard parameters in the usual way by expanding the field operators in terms of Wannier functions, which yields the general terms

$$\frac{1}{2}\int d\mathbf{r}\int d\mathbf{r}'\hat{\psi}^{\dagger}\left(\mathbf{r}\right)\hat{\psi}^{\dagger}\left(\mathbf{r}'\right)\hat{H}_{\mathrm{DD}}\left(\mathbf{r}-\mathbf{r}'\right)\hat{\psi}\left(\mathbf{r}'\right)\hat{\psi}\left(\mathbf{r}\right) \tag{2.39}$$
$$= \frac{1}{2}\sum_{\sigma_1\sigma_2\sigma_2'\sigma_1'}\sum_{i_1i_2i_2'i_1'}V^{\sigma_1\sigma_2\sigma_2'\sigma_1'}_{i_1i_2i_2'i_1'}\hat{a}^{\dagger}_{i_1\sigma_1}\hat{a}^{\dagger}_{i_2\sigma_2}\hat{a}_{i_2'\sigma_2'}\hat{a}_{i_1'\sigma_1'}\,,$$
$$V^{\sigma_1\sigma_2\sigma_2'\sigma_1'}_{i_1i_2i_2'i_1'} = \sum_q D^{\sigma_1'\sigma_2'}_{q;\sigma_1\sigma_2}W^{\sigma_1\sigma_2\sigma_2'\sigma_1'}_{q;i_1i_2i_2'i_1'}\,,$$
$$D^{\sigma_1'\sigma_2'}_{q;\sigma_1\sigma_2} = \sqrt{6}\left(-1\right)^q\langle\sigma_1\sigma_2|\left[\hat{\mathbf{d}}_1\otimes\hat{\mathbf{d}}_2\right]^{(2)}_q|\sigma_1'\sigma_2'\rangle\,, \tag{2.40}$$

---

[19] This is also apparent in that the p-wave scattering length vanishes identically for identical particles.

$$W^{\sigma_1\sigma_2\sigma_2'\sigma_1'}_{q;i_1i_2i_2'i_1'}=-\int d\mathbf{r}F^{\sigma_1\sigma_1'}_{i_1i_1'}\left(\mathbf{r}\right)\int d\mathbf{r}'\frac{C^{(2)}_{-q}\left(\mathbf{r}'-\mathbf{r}\right)}{\left|\mathbf{r}'-\mathbf{r}\right|^3}F^{\sigma_2\sigma_2'}_{i_2i_2'}\left(\mathbf{r}'\right)\,. \tag{2.41}$$

In the last line we have defined $F^{\sigma\sigma'}_{ij}\left(\mathbf{r}\right)\equiv w^{\star}_{i\sigma}\left(\mathbf{r}\right)w_{j\sigma'}\left(\mathbf{r}\right)$.

Our evaluation of the Hubbard parameters now breaks into two pieces, one dealing only with the dipole moments' dependence on the internal state and the other with the spatial distribution of the Wannier functions. Here we focus only on the geometrical part of the integral. For convenience, however, we will anticipate the result that only the $q=0$ terms are relevant for molecules, take a single internal state σ and define

$$W_{i_1i_2i_2'i_1'}\equiv W^{\sigma\sigma\sigma\sigma}_{0;i_1i_2i_2'i_1'}\,, \tag{2.42}$$

$$F_{ij}\left(\mathbf{r}\right)\equiv F^{\sigma\sigma}_{ij}\left(\mathbf{r}\right)\,, \tag{2.43}$$

$$D\equiv D^{\sigma\sigma}_{0;\sigma\sigma}\,. \tag{2.44}$$

The integral in question,

$$W_{i_1i_2i_2'i_1'}=-\int d\mathbf{r}F_{i_1i_1'}\left(\mathbf{r}\right)\int d\mathbf{r}'\frac{C^{(2)}_{0}\left(\mathbf{r}'-\mathbf{r}\right)}{\left|\mathbf{r}'-\mathbf{r}\right|^3}F_{i_2i_2'}\left(\mathbf{r}'\right)\,, \tag{2.45}$$

is naively six-dimensional integral. However, noting that

$$-\int d\mathbf{r}'\frac{C^{(2)}_{0}\left(\mathbf{r}'-\mathbf{r}\right)}{\left|\mathbf{r}'-\mathbf{r}\right|^3}F_{i_2i_2'}\left(\mathbf{r}'\right) \tag{2.46}$$

is the convolution of the dipole-dipole potential with the function $F_{i_2i_2'}\left(\mathbf{r}'\right)$, we can use the convolution theorem to find

$$W_{i_1i_2i_2'i_1'}=\int d\mathbf{r}F_{i_1i_1'}\left(\mathbf{r}\right)\mathcal{F}^{-1}\left[\mathcal{F}\left[-\frac{C^{(2)}_{0}\left(\mathbf{r}\right)}{\left|\mathbf{r}\right|^3}\right]\left(\mathbf{k}\right)\mathcal{F}\left[F_{i_2i_2'}\left(\mathbf{r}\right)\right]\left(\mathbf{k}\right)\right]\left(\mathbf{r}\right)\,, \tag{2.47}$$

where $\mathcal{F}\left[g\left(\mathbf{r}\right)\right]\left(\mathbf{k}\right)$ is the Fourier transform of $g\left(\mathbf{r}\right)$. The integral is now a series of two 3D Fourier transforms followed by an 3D integral in real space. The Fourier transform approach to compute dipolar integrals has also been used for dipolar Gross-

Pitaevskii equations [14, 15]. Using the expansion of a plane wave in terms of spherical harmonics [16]

$$e^{-i\mathbf{k}\cdot\mathbf{r}} = 4\pi \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} (-i)^{\ell} j_{\ell}(kr) Y_{m}^{\ell\star}(\theta_r, \phi_r) Y_{m}^{\ell}(\theta_k, \phi_k) , \tag{2.48}$$

with $j_{\ell}(x)$ the spherical Bessel function of order ℓ and (θ_x, ϕ_x) the spherical angles in the x coordinate system defined with respect to the quantization axis, we find the Fourier transform of the dipole-dipole potential to be

$$\mathcal{F}\left[-2\frac{C_0^2(\mathbf{r})}{r^3}\right](\mathbf{k}) = \mathcal{F}\left[\frac{1-3\cos^2\theta}{r^3}\right](\mathbf{k}) = \frac{4\pi}{3}\left(\cos^2\theta_k - 1\right) , \tag{2.49}$$

with θ_k the polar angle in $\mathbf{k}$-space. We note that the introduction of a spherical cutoff b on the lower limit of the r integration results in the function

$$4\pi\left(\cos^2\theta_k - 1\right)\left[\frac{\sin kb}{(kb^3)} - \frac{\cos kb}{(kb)^2}\right] . \tag{2.50}$$

The corrections to Eq. (2.49) scale as $(kb)^2$, and so are negligible for small b and k in the range we typically consider. We can thus safely take $b \to 0$ in computing these integrals and work with Eq. (2.49).

Numerically, the Fourier transforms can be performed in $\mathcal{O}(N^3 \log N)$ time using the fast Fourier transform (FFT) algorithm, where N is the number of grid points discretizing each $1D$ domain. More precisely, consider each 1D domain to be a symmetric finite interval $[-L/2, L/2]$ and then let $n_g - 1$ be even, with n_g the number of grid points. We then introduce the points

$$x_k = -\frac{L}{2} + \frac{(k-1)L}{n_g - 1} , \tag{2.51}$$

where k runs from 1 to n_g and $x_1 \equiv x_{n_g}$ because of the periodicity of the domain. The discrete Fourier domain of the $n_g - 1$ points $x_1 \ldots x_{n_g-1}$ is represented by the values $\frac{2\pi}{L}\left[0, 1, \ldots, \frac{n_g-1}{2}-1, \frac{n_g-1}{2}, -\frac{n_g-1}{2}+1, \ldots, -2, -1\right]$. Hence, the spacing in Fourier space is controlled by L, the length of the domain in real space, and the extent of the domain in Fourier space is controlled by n_g/L, the inverse step size. Because the Wannier functions $w(x)$ on a finite domain are periodic and band-limited, their discrete and continuous Fourier transforms are related by a scaling constant provided we sample the entire domain at a frequency of at least twice the largest frequency component of $w(x)$ [17]. This prevents us from having to consider the more advanced interpolation schemes required for Fourier integrals of general functions [18]. Furthermore, as is known for spectral methods, the calculation in Fourier space converges exponentially fast in L provided that n_g/L is large enough to capture the full support of the function in Fourier space.[20] Let us define g to be the extent of the function in the discrete Fourier space. We can then choose $n_g = 2 * g * L + 1$ to satisfy all of the above considerations. We find that for typical $g \sim 5 - 7$ which satisfy the Nyquist condition, the real space integration is of acceptable precision using a high-order Simpson integrator [18].

As the proposed calculation scales as $\mathcal{O}\left(L^3\right)$, we would like to determine the smallest L such that we obtain the value of the Hubbard parameter in the limit as $L \to \infty$ to within a desired tolerance. Consider computing W_{0dd0} on a domain of length L. There is a size-independent contribution to the dipole-dipole parameter which scales as $\sim 1/d^p$, where p is some power, but there is another contribution which scales as $\sim 1/|L-d|^{p_2}$ coming from the periodic boundary conditions. Because of the slow power law decay, this term can be quite sizable, especially when d itself is large. The solution is to compute the integral for a few L and then fit to the form

$$I\left(L\right) = a + \frac{b}{|L-d|^p}\,. \tag{2.52}$$

---

[20] The support of the function in Fourier space can be determined by using Parseval's theorem on finite Fourier subintervals to determine that the norm is unity to a desired tolerance.

a gives the true dipole-dipole parameter in the limit as $L \to \infty$. Practical experience shows that taking 4 values of L starting at roughly $4d$[21] produces an excellent fit.

Naively, it may be surprising that the integration over the dipole-dipole potential with its $1/r^3$ behavior yields convergent results. The key point is that the dipole-dipole potential is also proportional to a rank-two spherical harmonic, and so picks out the components of the Wannier function product with d-wave symmetry. This component necessarily vanishes at the origin, and so there is no actual divergence in the integrand. Additionally, because of the strong anisotropy of the spherical harmonic, this Hubbard parameter is quite sensitive to transverse confinement. Confinement modifies not only the strength of the potential as in the short-range case, but also the power p of the potential. This result will be explored in greater detail in an upcoming publication [19].

Thus, the result of expanding the dipole-dipole interaction in terms of Wannier functions is a term

$$\hat{H} = \sum_{i<j} U^{i,j} \hat{n}_i \hat{n}_j \,, \tag{2.53}$$

where $U^{i,j}$ exhibits power-law behavior at large separations $|i-j|$ and is sensitive to transverse confinement. That is, Hubbard models for molecules involve long-ranged interactions, as opposed to models for atoms. For molecules with multiple internal states which have dipole-allowed transitions, the dipole-dipole potential also gives rise to an "exchange" contribution which describes the long-ranged propagation of a rotational quantum through the system [20–22].

The general procedure outlined in this chapter is used in Part II to derive effective models for heteronuclear bialkali molecules in optical lattices. However, the procedure here is not the only way in which a Hubbard model can be derived. Part III

---

[21] 4 was chosen by assuming that the decay was purely dipolar and then determining the L such that $1/d^3 > x/(L-d)^3$ for some scaling factor x. Taking $x \sim 100$ gives an estimate of $L \sim 4d$.

uses a different technique, in which the relevant Wannier functions are formed from bound states of a two-particle Hamiltonian. These two-particle bound states are then coupled to fermions in the lowest Bloch band through a Wannier function overlap of a pairing Hamiltonian. In this way, higher bands are included in the Hubbard model, but only in the configurations which are determined to be physically relevant from few-body physics.

2.3 References Cited

[1] R. Shankar. Renormalization-group approach to interacting fermions. *Rev. Mod. Phys.*, 66:129–192, Jan 1994. doi: 10.1103/RevModPhys.66.129. URL `http://link.aps.org/doi/10.1103/RevModPhys.66.129`.

[2] A. L. Fetter and J. D. Walecka. *Quantum Theory of Many-Particle Systems*. McGraw-Hill, Inc., New York, 1971.

[3] J. Hubbard. Electron correlations in narrow energy bands. *Proc. R. Soc. London A*, 276:238, 1963.

[4] A. Auerbach. *Interacting Electrons and Quantum Magnetism*. Springer, Berlin, 1994.

[5] K. D. Bonin and V. V. Kresin, editors. *Electric-Dipole Polarizabilities of Atoms, Molecules, and Clusters*. World Scientific, Singapore, 1988.

[6] M. L. Wall and L. D. Carr. In preparation.

[7] Neil Ashcroft and David Mermin. *Solid State Physics*. Saunders College Publishing, Orlando, 1976.

[8] W. Kohn. Analytic Properties of Bloch Waves and Wannier Functions. *Phys. Rev.*, 115(4):809–821, Aug 1959. doi: 10.1103/PhysRev.115.809.

[9] Ana Maria Rey. *Ultracold Bosonic Atoms in Optical Lattices*. PhD thesis, University of Maryland, 2004.

[10] Immanuel Bloch, Jean Dalibard, and Wilhelm Zwerger. Many-body physics with ultracold gases. *Rev. Mod. Phys.*, 80:885, 2008.

[11] T. D. Lee, Kerson Huang, and C. N. Yang. Eigenvalues and Eigenfunctions of a Bose System of Hard Spheres and Its Low-Temperature Properties. *Phys. Rev.*, 106(6):1135–1145, Jun 1957. doi: 10.1103/PhysRev.106.1135.

[12] Zbigniew Idziaszek and Tommaso Calarco. Pseudopotential Method for Higher Partial Wave Scattering. *Phys. Rev. Lett.*, 96:013201, Jan 2006. doi: 10.1103/PhysRevLett.96.013201. URL `http://link.aps.org/doi/10.1103/PhysRevLett.96.013201`.

[13] C.J. Pethick and H. Smith. *Bose-Einstein Condensation in Dilute Gases.* Cambridge University Press, Cambridge, 2002.

[14] Krzysztof Góral, Kazimierz Rzążewski, and Tilman Pfau. Bose-Einstein condensation with magnetic dipole-dipole forces. *Phys. Rev. A*, 61:051601, Mar 2000. doi: 10.1103/PhysRevA.61.051601. URL `http://link.aps.org/doi/10.1103/PhysRevA.61.051601`.

[15] D. C. E. Bortolotti, S. Ronen, J. L. Bohn, and D. Blume. Scattering Length Instability in Dipolar Bose-Einstein Condensates. *Phys. Rev. Lett.*, 97:160402, Oct 2006. doi: 10.1103/PhysRevLett.97.160402. URL `http://link.aps.org/doi/10.1103/PhysRevLett.97.160402`.

[16] K. Gottfried and T.-M. Yan. *Quantum Mechanics.* Springer, New York, 2nd edition, 2004.

[17] E. O. Brigham. *The Fast Fourier Transform and Applications.* Prentice Hall, Englewood Cliffs, NJ, 1988.

[18] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge Univ. Press, Cambridge, U.K., 1993.

[19] M. L. Wall and L. D. Carr. Confinement Effects in Dipolar Gases on Optical Lattices, 2012. In preparation.

[20] Alexey V. Gorshkov, Salvatore R. Manmana, Gang Chen, Jun Ye, Eugene Demler, Mikhail D. Lukin, and Ana Maria Rey. Tunable Superfluidity and Quantum Magnetism with Ultracold Polar Molecules. *Phys. Rev. Lett.*, 107:115301, Sep 2011. doi: 10.1103/PhysRevLett.107.115301. URL `http://link.aps.org/doi/10.1103/PhysRevLett.107.115301`.

[21] Alexey V. Gorshkov, Salvatore R. Manmana, Gang Chen, Eugene Demler, Mikhail D. Lukin, and Ana Maria Rey. Quantum magnetism with polar alkali-metal dimers. *Phys. Rev. A*, 84:033619, Sep 2011. doi: 10.1103/PhysRevA.84.033619. URL `http://link.aps.org/doi/10.1103/PhysRevA.84.033619`.

[22] Felipe Herrera and Roman V. Krems. Tunable Holstein model with cold polar molecules. *Phys. Rev. A*, 84:051401, Nov 2011. doi: 10.1103/PhysRevA.84.051401. URL `http://link.aps.org/doi/10.1103/PhysRevA.84.051401`.

PART II

THE MOLECULAR HUBBARD HAMILTONIAN

CHAPTER 3

EMERGENT TIMESCALES IN ENTANGLED QUANTUM DYNAMICS OF ULTRACOLD MOLECULES IN OPTICAL LATTICES

Abstract:[22] We derive a novel lattice Hamiltonian, the *Molecular Hubbard Hamiltonian*, which describes the essential many-body physics of closed-shell ultracold heteronuclear molecules in their absolute ground state in a quasi-one- dimensional optical lattice. The molecular Hubbard Hamiltonian is explicitly time dependent, making a dynamic generalization of the concept of quantum phase transitions necessary. Using the time-evolving block decimation algorithm to study entangled dynamics, we demonstrate that, in the case of hard-core bosonic molecules at half-filling, the molecular Hubbard Hamiltonian exhibits emergent timescales over which spatial entanglement grows, crystalline order appears and oscillations between rotational states self-damp into an asymptotic superposition. We show that these timescales are non-monotonic functions of the physical parameters describing the lattice.

3.1 Introduction

In recent years, ultracold atomic gases have provided near perfect realizations of condensed matter Hamiltonians, acting as *quantum simulators* [1, 2] that allow the study of complex condensed matter phenomena in a clean and highly controllable environment. Ultracold polar molecular gases, which have recently been brought to the edge of quantum degeneracy in their absolute ground state [3, 4], offer additional features over atomic gases, such as a large internal Hilbert space and a greater susceptibility to external fields via a permanent electric dipole. There have been a number of proposals on how to use ultracold molecular gases for mimicking well-known Hamilto-

---

[22] Published previously as *Emergent Timescales in Entangled Quantum Dynamics of Ultracold Molecules in Optical Lattices*, M. L. Wall and L. D. Carr, New J. Phys. **11** 055027 (2009).

nians such as spin-1 lattice models [5]. Ultracold molecules have also been suggested as a model system for the study of strongly correlated 2D quantum phases [6] or for quantum information processing schemes [7–9]. However, these proposals frequently involve complex and yet-to-be implemented experimental techniques. In this article, we instead focus on the completely *new* quantum many body physics which results naturally from the simplest quantum lattice experiments that can be performed in the immediate future with established techniques in ultracold molecular quantum gases.

Towards this end we derive a novel lattice Hamiltonian, which we refer to as the *Molecular Hubbard Hamiltonian* (MHH). The MHH describes the physics of an ultracold polar molecular gas in a 1D optical lattice that is oriented using a DC electric field, giving rise to a resonant dipole-dipole interaction, and is driven between rotational levels using a microwave AC field. In particular, new aspects of our derivation include explicit dependence of hopping energy on the molecular polarizability tensor. This in turn allows a determination of the tensor elements, an outstanding experimental issue, from the borders of the static phase diagram of the MHH, which are identical to those of the extended Bose-Hubbard Hamiltonian [10] when a single molecular rotational level is occupied.

Beyond the statics, the MHH naturally has a dynamical component due to the AC driving fields, as well as an internal structure in terms of rotational modes which is inherently different from spinor atomic systems [11, 12]. We study this dynamical aspect with Time-Evolving Block Decimation (TEBD) [13, 14], a newly developed entangled quantum dynamics algorithm which takes spatial entanglement (specifically, Schmidt number [15]) as a cut-off. We find two emergent timescales in the case of half-filling for hard core bosonic molecules. We emphasize that a *quantum* lattice model requires low filling (average number of particles per site), in contrast to a mean field lattice model, for which the filling would typically be quite high. Thus, although experiments can most easily access the mean field regime of hundreds of molecules per site with a single pair of counter-propagating laser beams, we look slightly ahead to the quantum regime, which will require two pairs of such beams in order to create an array of quasi-1D "tubes." A third pair is then used to create the lattice in each tube. This technique is already well established for ultracold atoms [16].

Dynamical aspects of quantum phase transitions are just beginning to be considered [17, 18], and have so far been a limited area of study restricted to mean field considerations, due to lack of numerical tools. With the recent advent of entangled quantum dynamics algorithms, namely TEBD, dynamical properties of many-body systems are becoming amenable to numerical study. For example, TEBD has been used to address key questions such as the dynamics of a quantum quench [19, 20] or the speed at which correlations propagate in a lattice [21]; these are not issues which can be studied with other dynamical methods such as dynamical mean field theory (DMFT).[23] We give a brief review of TEBD in Sec. 3.3.1. The reader interested in computational details can find them in Ref. [23].

The first main contribution of this paper is to present a careful derivation of the Molecular Hubbard Hamiltonian. This is done in Sec. 3.2, with some previously known aspects of molecular physics relegated to appendix 3.6. The second main contribution is to present emergent timescales for half filling; although we treat the case of hard core bosons, the MHH can also be applied to fermionic molecules. To this end, in Sec. 3.3 we first give a brief explanation of TEBD and the quantum measures we use. Then, in Sec. 3.4 we present and analyze our simulations, with an accompanying convergence study in appendix 3.7. Finally, in Sec. 3.5 we summarize.

3.2 The Molecular Hubbard Hamiltonian

The Molecular Hubbard Hamiltonian (MHH) is

---

[23] Time-dependent Density Functional Theory (TDFT) may be able to succeed in a partial analysis of entangled dynamics, which might even be complementary to methods such as TEBD, but this is not yet at all clear in the literature [22].

$$
\begin{aligned}
\hat{H} \;=\; & -\sum_{JJ'M} t_{JJ'M} \sum_{\langle i,i' \rangle} \left(\hat{a}^{\dagger}_{i',J'M} \hat{a}_{iJM} + \text{h.c.} \right) \\
& + \sum_{JM} E_{JM} \sum_{i} \hat{n}_{iJM} - \pi \sin\left(\omega t\right) \sum_{JM} \Omega_{JM} \sum_{i} \left(\hat{a}^{\dagger}_{iJ,M} \hat{a}_{iJ+1,M} + \text{h.c.} \right) \\
& + \frac{1}{2} \sum_{\substack{J_1,\, J_1',\, J_2,\, J_2' \\ M,\, M'}} U_{dd}{}^{\substack{J_1,\, J_1',\, J_2,\, J_2' \\ M,\, M'}} \sum_{\langle i,i' \rangle} \hat{a}^{\dagger}_{iJ_1M} \hat{a}_{iJ_1'M} \hat{a}^{\dagger}_{i'J_2M'} \hat{a}_{i'J_2'M'} . \,, \qquad (3.1)
\end{aligned}
$$

where $\hat{a}_{iJM}$ destroys a bosonic or fermionic molecule in the $|\mathcal{E}; JM\rangle$ state (defined below) on the i^{th} lattice site, and the bracket notation $\langle \ldots \rangle$ denotes that the sum is taken over nearest neighbors. The first term in Eq. (3.1) corresponds to hopping both between sites and molecular rotational states with quantum numbers J, M. The second term represents the rotational energy along with rotational state-dependent energy differences due to a DC electric field. The third term corresponds to an AC electric field, making this a driven system. The fourth term corresponds to electric dipole-dipole interactions. In the following subsections and appendix 3.6 we justify Eq. (3.1) with a careful derivation and present the energy scales of each term.

3.2.1 Derivation of the Molecular Hubbard Hamiltonian

The full molecular Hamiltonian in second quantization is

$$
\begin{aligned}
\hat{H} \;=\; & \int d^3r\, \hat{\psi}^{\dagger}\left(\mathbf{r}\right) \left[\hat{H}_{\text{kin}} + \hat{H}_{\text{rot}} + \hat{H}_{\text{DC}} + \hat{H}_{\text{AC}}\left(t\right) + \hat{H}_{\text{opt}}\left(\mathbf{r}\right) \right] \hat{\psi}\left(\mathbf{r}\right) \\
& + \int d^3r d^3r'\, \hat{\psi}^{\dagger}\left(\mathbf{r}\right) \hat{\psi}^{\dagger}\left(\mathbf{r}'\right) \hat{H}_{\text{dd}}\left(|\mathbf{r}-\mathbf{r}'|\right) \hat{\psi}\left(\mathbf{r}'\right) \hat{\psi}\left(\mathbf{r}\right) . \qquad (3.2)
\end{aligned}
$$

The terms on the first line correspond to single-molecule effects: kinetic energy, rotation, the DC electric field which orients the dipole, the AC microwave field which drives transitions between rotational levels, and the far off-resonant optical lattice potential, respectively. The second line is the two-molecule resonant dipolar energy. The field operators $\hat{\psi}$ can be either bosonic or fermionic. We focus on the bosonic case for brevity. There are five key assumptions underlying our derivation, as follows. We consider all five assumptions to be reasonable for present and near-future experiments.

1. We consider ultracold closed-shell polar heteronuclear diatomic molecules, characterized by permanent dipole moment d and rotational constant B. The most experimentally relevant bosonic species in this category are SrO, RbCs, and LiCs [6]. The individual molecules are assumed to be in their electronic and vibrational ground states, and it is assumed that none of these degrees of freedom can be excited at the large intermolecular separations and low temperatures/relative energies that we consider.
2. The molecule is assumed to have a $^1\Sigma$ ground state. The characteristic trapping potential length is chosen large enough compared to the internuclear axis to assume spherical symmetry, i.e. a locally constant potential.
3. We neglect any intramolecular interactions (e.g., hyperfine structure), as they are typically very small for $^1\Sigma$ molecules [24].
4. We consider only the lowest three rotational levels. All AC fields will be sufficiently weak to allow this assumption.
5. We work in the hard-core limit where at most one molecule is allowed per site. This is enforced by strong dipole-dipole interactions on-site. We consider the lattice spacing large enough to include only nearest-neighbor dipole-dipole interactions. Other short-range interactions such as exchange or chemical reactions or long range interactions such as dispersion and quadrupole-quadrupole interactions are not considered.

We proceed to follow the usual procedure [2] of expanding the field operators of our second-quantized Hamiltonian in a Wannier basis of single-molecule states centered at a particular discrete position $\mathbf{r}_i$:

$$\hat{\psi} \quad = \textstyle\sum_i \hat{a}_i w\left(\mathbf{r}-\mathbf{r}_i\right)\;, \tag{3.3}$$

where i is a site index and the sum is over all lattice sites. For our Wannier Basis we choose the single-molecule basis that diagonalizes the rotational and DC electric field Hamiltonians, spanned by kets $|\mathcal{E};JM\rangle$. In this basis, which we refer to as the "dressed basis" (the DC field "dresses" the rotational basis) we have the field operator expansion

$$\hat{\psi}_{JM} \quad = \textstyle\sum_i \hat{a}_{iJM} w_{JM}\left(\mathbf{r}-\mathbf{r}_i\right) \equiv \sum_i \hat{a}_{iJM}|\mathcal{E};JM\rangle_i\,. \tag{3.4}$$

We note that such a basis, while highly efficient for the hard core limit we consider, becomes progressively worse for higher filling factors, till in the mean field limit the single-molecule basis, whether dressed or not, is so poor that many bands must be considered. Here we do not include a band index for simplicity, although the generalization of Eq. (3.1) to include multiple bands is straightforward.

This choice of Wannier basis associates the terms in Eq. (3.1) to the terms in Eq. (3.2) as follows:

$$\begin{aligned}
t_{J,J',M} &\equiv -\int d\mathbf{r}\, w^\star_{JM}\left(\mathbf{r}-\mathbf{r}_i\right)\left[H_{\mathrm{kin}}+H_{\mathrm{opt}}\right] w_{J'M}\left(\mathbf{r}-\mathbf{r}_{i+1}\right), && (3.5)\\
E_{JM} &\equiv \int d\mathbf{r}\, w^\star_{JM}\left(\mathbf{r}-\mathbf{r}_i\right)\left[H_{\mathrm{rot}}+H_{\mathrm{DC}}\right] w_{JM}\left(\mathbf{r}-\mathbf{r}_i\right), && (3.6)\\
-\pi\Omega_{JM}\sin\left(\omega t\right) &\equiv \int d\mathbf{r}\, w^\star_{JM}\left(\mathbf{r}-\mathbf{r}_i\right)\left[H_{\mathrm{AC}}\right] w_{J+1,M}\left(\mathbf{r}-\mathbf{r}_i\right), && (3.7)\\
U_{dd}^{\substack{J_1,\,J_1',\,J_2,\,J_2'\\ M,\,M'}} &\equiv \int d\mathbf{r}d\mathbf{r}'\, w^\star_{J_1'M}\left(\mathbf{r}-\mathbf{r}_i\right) w^\star_{J_2'M'}\left(\mathbf{r}'-\mathbf{r}_{i+1}\right) && (3.8)\\
&\quad\times H_{dd}\left(\mathbf{r}-\mathbf{r}'\right) w_{J_1M}\left(\mathbf{r}-\mathbf{r}_i\right) w_{J_2M'}\left(\mathbf{r}'-\mathbf{r}_{i+1}\right)\;,
\end{aligned}$$

where the operators H_{kin}, H_{opt}, etc., are taken to be in position space representation. For the derivation of the single-molecule terms (rotational, DC electric field, and AC electric field) and discussion of the properties of our Wannier basis we refer the reader to appendix 3.6. In the following sections we present the derivation of the tunneling (hopping) and dipole-dipole terms, which have new aspects not heretofore appearing in the literature [25].

3.2.2 Tunneling

The tunneling term represents the sum of the molecular kinetic energy with the potential energy of the lattice. After expanding in the Wannier basis of Eq. (3.4), we find the effective tunneling Hamiltonian

$$\hat{H}_t^{\mathrm{eff}} = -\sum_{J,J',M} t_{JJ'M} \sum_{\langle i,i' \rangle} \left(\hat{a}^\dagger_{i,J'M} \hat{a}_{i',JM} + \mathrm{h.c.} \right) \tag{3.9}$$

where $t_{J,J',M}$ was defined in Eq. (3.5). To understand why this operator mixes states of different J, we note that the kinetic energy and (far off-resonant) optical lattice potential do not mix rotational eigenstates. Because our Wannier basis states are dressed and therefore superpositions of rotational eigenstates with different J, the tunneling operator in the dressed basis will mix J. Although the dressed basis makes the tunneling more complex to analyze, it simplifies other terms in the MHH, such as the DC term, and is in any case a more standard basis for analysis of the diatomic molecules we study here. Comparable basis changes are sometimes made in other quantum many body systems, where, for instance, particles and holes are mixed, or particles are paired. Note that, because we assume z-polarized fields, M is still a good quantum number. To discuss the actual form of the tunneling energies $\{t_{J,J',M}\}$ we must first examine the interaction of a diatomic molecule with the optical lattice.

3.2.3 Interaction with an Optical Lattice

The charge redistribution that occurs when a molecule is subjected to a static, spatially uniform electric field $\mathbf{E}$ is reflected in its dipole moment $\mathbf{d}$ via the polarizability series

$$d_j \quad = d_j^{(0)} + \alpha_{jk}E_k + \tfrac{1}{2!}\beta_{jkl}E_kE_l + \tfrac{1}{3!}\Gamma_{jklm}E_kE_lE_m + \dots \tag{3.10}$$

where the first, second, and third order coefficients α_{jk}, β_{jkl}, and Γ_{jklm} are elements of the polarizability, hyperpolarizability, and second hyperpolarizability tensors, respectively. The polarizability tensor is a symmetric rank-two tensor with no more than six independent elements (less if molecular symmetry is greater), and characterizes the lowest order dipole moment induced by an applied electric field. From this tensor we can form the scalar invariants

$$\bar{\alpha} \quad \equiv \tfrac{1}{3}\mathrm{Tr}\tilde{\alpha}\,, \tag{3.11}$$

$$(\Delta\alpha)^2 \quad \equiv \tfrac{1}{2}\left[3\mathrm{Tr}(\tilde{\alpha}^2) - (\mathrm{Tr}\tilde{\alpha})^2\right]\,, \tag{3.12}$$

referred to as the polarizability and the polarizability anisotropy, respectively. Note that we use the tilde to clarify that $\tilde{\alpha}$ with elements α_{jk} is a tensor, not a scalar – we reserve the accent circumflex (the "hat" symbol) for quantum operators. In linear molecules, such as diatomic molecules, the presence of only two distinct moments of inertia allows for the classification of $\tilde{\alpha}$ according to its components along and perpendicular to the internuclear axis, denoted $\alpha_\parallel$ and $\alpha_\perp$, respectively. In the presence of AC electric fields with frequency ω we speak of the dynamic polarizability tensor $\tilde{\alpha}\,(\omega)$, with the series of Eq. (3.10) being the zero frequency limit. The tensor $\tilde{\alpha}\,(\omega)$ is, in general, complex, with the real part inducing a dipole moment and the imaginary part accounting for power absorption by the dipole and out-of-phase dipole oscillation. In the case of Σ diatomic molecules in their electronic and vibrational ground states [25]

$$\tilde{\alpha}\,(\omega) \quad \equiv \alpha_\parallel\,(\omega)\,\mathbf{e}'_0 \otimes \mathbf{e}'_0 + \alpha_\perp\,(\omega)\sum_{\Lambda=\pm1}(-1)^\Lambda\,\mathbf{e}'_\Lambda \otimes \mathbf{e}'_{-\Lambda}\,, \tag{3.13}$$

where the $\mathbf{e}'_q$ are molecule-fixed spherical basis vectors. The parallel and perpendicular dynamic polarizabilities are

$$\alpha_{\|} = \textstyle\sum_{\pm} \sum_{\nu,v} \frac{\left|d_{\nu\Sigma(v)-X\Sigma(0)}\right|^2}{E_{\nu\Sigma(v)}-E_{X\Sigma(0)}\mp\hbar\omega}\,, \tag{3.14}$$

$$\alpha_{\perp} = \textstyle\sum_{\pm} \sum_{\nu,v} \frac{\left|d_{\nu\Pi(v)-X\Sigma(0)}\right|^2}{E_{\nu\Pi(v)}-E_{X\Sigma(0)}\mp\hbar\omega}\,, \tag{3.15}$$

respectively. In these expressions $d_{\nu\Lambda(v)-X\Sigma(0)}$ is the transition dipole moment from the ground state to the $\nu\Lambda\,(v)$ state (following the usual diatomic molecular notation, $\Lambda \in \{\Sigma, \Pi\} \equiv \{0, 1\}$ is the quantum number associated with the projection of the total electronic orbital angular momentum along the internuclear axis, i.e., in the molecule-fixed basis) and the sum over $\mp$ accounts for the near-resonant and typically far off-resonant terms.

Transforming $\tilde{\alpha}$ from the molecule-fixed basis to the space-fixed basis using the transformation discussed in appendix 3.6, we find

$$\begin{aligned}\tilde{\alpha}'\left(\omega_L\right) = &\sum_{p_1 p_2} \sum_{j=0,2} \sum_{m=-j}^{j} (2j+1) \begin{pmatrix} 1 & 1 & j \\ p_1 & p_2 & m \end{pmatrix} \sqrt{\frac{1}{(2-j)!\,(3+j)!}} \\ &\times \left[\alpha_{\|}\,(j+2)\,(j-1) - 4\alpha_{\perp}\right] C_m^{(j)} \mathbf{e}_{p_1} \otimes \mathbf{e}_{p_2}\,,\end{aligned} \tag{3.16}$$

where $C_m^{(j)}$ is an unnormalized spherical harmonic, $(\ldots)$ denotes the Wigner 3-j coefficient [26], and the $\mathbf{e}_p$ are space-fixed spherical basis vectors.

The interaction of the lattice with the molecule is represented by the Hamiltonian

$$H_{\text{opt}}\left(\mathbf{x}\right) = -\mathbf{E}^{\star}_{\text{opt}}\left(\mathbf{r}\right) \cdot \tilde{\alpha}'\left(\omega_L\right) \cdot \mathbf{E}_{\text{opt}}\left(\mathbf{r}\right)\ , \tag{3.17}$$

where $\mathbf{E}_{\text{opt}}\left(\mathbf{r}\right)$ is the electric field of the optical lattice. If the electric field has polarization p in the space-fixed spherical basis then we find

$$\begin{aligned} H_{\text{opt}}\left(\mathbf{x}\right) = & -\frac{\left|\mathbf{E}_{\text{opt}}\left(\mathbf{r}\right)\right|^2}{3}\Big[\left(\alpha_\parallel + 2\alpha_\perp\right)C_0^{(0)} \\ & + \left(-1\right)^p \frac{2}{\left(1-p\right)!\left(1+p\right)!}\left(\alpha_\parallel - \alpha_\perp\right)C_0^{(2)}\Big] . \end{aligned} \tag{3.18}$$

For light linearly polarized in the $\hat{x}$-direction we obtain

$$\begin{aligned} H_{\text{opt}} = & -\frac{\left|\mathbf{E}_{\text{opt}}\left(\mathbf{r}\right)\right|^2}{6}\Big[2\left(\alpha_\parallel + 2\alpha_\perp\right)C_0^{(0)} \\ & + \left(\alpha_\parallel - \alpha_\perp\right)\left(\sqrt{6}C_{-2}^{(2)} - 2C_0^{(2)} + \sqrt{6}C_2^{(2)}\right)\Big] , \end{aligned} \tag{3.19}$$

whereas for light linearly polarized in the $\hat{y}$-direction we find

$$\begin{aligned} H_{\text{opt}} = & \frac{\left|\mathbf{E}_{\text{opt}}\left(\mathbf{r}\right)\right|^2}{6}\Big[-2\left(\alpha_\parallel + 2\alpha_\perp\right)C_0^{(0)} \\ & + \left(\alpha_\parallel - \alpha_\perp\right)\left(\sqrt{6}C_{-2}^{(2)} + 2C_0^{(2)} + \sqrt{6}C_2^{(2)}\right)\Big] . \end{aligned} \tag{3.20}$$

Since $C_0^{(0)} = 1$, these terms give a state-independent energy shift. The $C_q^{(2)}$ terms produce a tensor shift. Because the depth (in energy) of a typical optical lattice is much smaller than the energy of transitions between rotational levels (of order B, as defined in 3.6.2), we can ignore far off-resonant Raman coupling between different J manifolds and use only the diagonal matrix elements. The $C_2^{(2)}$ term and the $C_{-2}^{(2)}$ will both mix M in the $J \geq 2$ manifolds, but do not affect the lowest two rotational levels, again, because we neglect Raman couplings. Thus x, y, and z polarizations all have the same Hamiltonian in this approximation. We can calculate the matrix elements of $C_0^{(2)}$ in the field free basis using the Wigner-Eckart theorem to find

$$\begin{aligned} \langle J'M'|H_{\text{opt}}\left(\mathbf{r}\right)|JM\rangle = & -\frac{\left|\mathbf{E}_{\text{opt}}\left(\mathbf{r}\right)\right|^2}{3}\delta_{JJ'}\delta_{MM'}\Big[\left(\alpha_\parallel + 2\alpha_\perp\right) \\ & + \left(-1\right)^p \frac{2}{\left(1-p\right)!\left(1+p\right)!}\left(\alpha_\parallel - \alpha_\perp\right)\frac{J\left(J+1\right)-3M^2}{\left(2J-1\right)\left(2J+3\right)}\Big] . \end{aligned} \tag{3.21}$$

In our effective Hamiltonian we choose right circular polarization for the z lattice, x polarization for the x lattice, and y polarization for the y lattice, where each lattice refers to a pair of counter-propagating laser beams used to create a standing wave. Other choices are of course also possible, and the mathematical derivation leads to a Hamiltonian similar in form to that of equation (3.1). However, it is possible for hopping to depend on two M indices, e.g. $t_{JJ'MM'}$.

We consider the fields making up the optical lattice to have sinusoidal spatial profiles, resulting in sine-squared intensity profiles. In addition, we assume that the y and z lattices are tight, meaning that the molecules are strongly confined at the potential minimum (for a red-detuned trap). This tight confinement allows us to approximate them via a Taylor series, e.g., $\sin^2(k_z z) \simeq k_z^2 z^2$ in the vicinity of the molecule. Using the above results, the matrix elements of the Hamiltonian for the optical lattice can be written

$$\begin{aligned}
&\langle J'M'|H_{\text{opt}}(\mathbf{r})|JM\rangle \\
&= -\frac{|\mathbf{E}_{\text{opt}}(\mathbf{y})|^2 k_y^2 y^2 + |\mathbf{E}_{\text{opt}}(\mathbf{x})|^2 \sin^2(k_x x)}{3}\left[\bar{\alpha} + 2\Delta\alpha \frac{J(J+1) - 3M^2}{(2J-1)(2J+3)}\right]\delta_{JJ'}\delta_{MM'} \\
&\quad - \frac{|\mathbf{E}_{\text{opt}}(\mathbf{z})|^2 k_z^2 z^2}{3}\left[\bar{\alpha} - \Delta\alpha \frac{J(J+1) - 3M^2}{(2J-1)(2J+3)}\right]\delta_{JJ'}\delta_{MM'} \qquad (3.22)
\end{aligned}$$

or, more compactly, as

$$\begin{aligned}
\langle J'M'|H_{\text{opt}}(\mathbf{r})|JM\rangle &= \left[-\alpha_{JM}^{(t)}|\mathbf{E}_{\text{opt}}(\mathbf{y})|^2 k_y^2 y^2 - \alpha_{JM}^{(t)}|\mathbf{E}_{\text{opt}}(\mathbf{x})|^2 \sin^2(k_x x)\right]\delta_{JJ'}\delta_{MM'} \\
&\quad - |\mathbf{E}_{\text{opt}}(\mathbf{z})|^2 \alpha_{JM}^{(z)} k_z^2 z^2 \delta_{JJ'}\delta_{MM'} \qquad (3.23)
\end{aligned}$$

by defining

$$\alpha_{JM}^{(t)} \equiv \frac{1}{3}\left[\bar{\alpha} + 2\Delta\alpha \frac{J(J+1) - 3M^2}{(2J-1)(2J+3)}\right], \qquad (3.24)$$

$$\alpha_{JM}^{(z)} \equiv \frac{1}{3}\left[\bar{\alpha} - \Delta\alpha \frac{J(J+1) - 3M^2}{(2J-1)(2J+3)}\right]. \qquad (3.25)$$

We now define, as is customary, the "lattice heights" in the x, y, and z directions, respectively, as

$$\begin{aligned} V_x^{(JM)} &\equiv -\left|\mathbf{E}_{\text{opt}}\left(\mathbf{x}\right)\right|^2 \alpha_{JM}^{(t)}, & (3.26)\\ V_y^{(JM)} &\equiv -\left|\mathbf{E}_{\text{opt}}\left(\mathbf{y}\right)\right|^2 \alpha_{JM}^{(t)}, & (3.27)\\ V_z^{(JM)} &\equiv -\left|\mathbf{E}_{\text{opt}}\left(\mathbf{z}\right)\right|^2 \alpha_{JM}^{(z)}\,. & (3.28)\end{aligned}$$

The tight confinement in the transverse (y and z) directions strongly suppresses tunneling in these directions, making the overall lattice effectively 1D along x.

From Eqs. (3.24)-(3.25), it is apparent that different rotational levels experience different trapping frequencies and different tunneling energies. To make this clearer, we parse our full field-free tunneling matrix element as

$$\begin{aligned} t_{JM} &\equiv -\int d\mathbf{r}\, w_{JM}^{\star}\left(\mathbf{r}-\mathbf{r}_i\right)\left[H_{\text{kin}}+H_{\text{opt}}\right] w_{JM}\left(\mathbf{r}-\mathbf{r}_{i+1}\right)\\ &= \int d\mathbf{r}\, w_{JM}^{\star}\left(\mathbf{r}-\mathbf{r}_i\right)\left[-H_{\text{kin}}+V_x^{(JM)}\sin^2\left(k_x x^2\right)\right] w_{JM}\left(\mathbf{r}-\mathbf{r}_{i+1}\right)\\ &\quad+\int d\mathbf{r}\, w_{JM}^{\star}\left(\mathbf{r}-\mathbf{r}_i\right)\left[V_y^{(JM)}k_y^2 y^2+V_z^{(JM)}k_z^2 z^2\right] w_{JM}\left(\mathbf{r}-\mathbf{r}_{i+1}\right)\,. \qquad (3.29)\end{aligned}$$

Because we consider tight traps such that the lattice heights in the y- and z-directions are much greater than the lattice height in thex-direction, $V_y^{(JM)} \sim V_z^{(JM)} \gg V_x^{(JM)}$, the contributions from the $V_y^{(JM)}$ and $V_z^{(JM)}$ terms are exponentially suppressed compared to the $V_z^{(JM)}$ and H_{kin} terms, and so we neglect them. This is equivalent to the array of tubes we discussed in section 1, where each tube is isolated from its neighbors. The matrix element becomes

$$t_{JM} \approx \int d\mathbf{r} w_{JM}^{\star}\left(\mathbf{r}-\mathbf{r}_i\right)\left[-H_{\text{kin}}+V_x^{(JM)}\sin^2\left(k_x x^2\right)\right] w_{JM}\left(\mathbf{r}-\mathbf{r}_{i+1}\right)\,. \qquad (3.30)$$

In the evaluation of the integral, Eq. (3.30) we assume that the Bloch function of a molecule in the sinusoidal optical lattice is a Mathieu function along x. This may seem to contradict our assumption of spherical symmetry in the above derivation.

Table 3.1: Values of the polarizabilities for LiCs in different rotational states $|JM\rangle$.

$\|JM\rangle$	$3\alpha^{(t)}_{JM}/\bar{\alpha}$	$3\alpha^{(z)}_{JM}/\bar{\alpha}$
$\|00\rangle$	1	1
$\|10\rangle$	1.715	0.642
$\|1\pm 1\rangle$	0.642	1.178
$\|20\rangle$	1.511	0.744
$\|2\pm 1\rangle$	1.255	0.872
$\|2\pm 2\rangle$	0.488	1.255

However, the assumption of spherical symmetry (i.e. a locally constant potential) need only hold on the order of an internuclear axis ($\sim 5\mathring{A}$) near the molecule. In contrast, on the order of the characteristic lattice length $\sqrt{\hbar/\mu\omega_{\text{opt}}}$ the rigid-rotor molecule is indistinguishable from a point particle (such as an alkali atom), and so spherical symmetry is not required. With this understanding, we recognize $t^{(0)}_{JM}$ as the expression for the hopping energy for point particles in optical lattices [27] with the additional feature that the lattice height along the quasi-1D direction $V_0 = V_x^{(JM)}$ is dependent on J through the polarizability tensor. Thus, altering the expression from the theory of point particles in optical lattices, we obtain the result

$$\frac{t^{(0)}_{JM}}{E_R} \approx A\left(\frac{V_x^{(JM)}}{E_R}\right)^B \exp\left(-C\sqrt{\frac{V_x^{(JM)}}{E_R}}\right), \tag{3.31}$$

where $A = 1.397$, $B = 1.051$, $C = 2.121$, and

$$E_R \equiv \hbar^2 k_x^2/2m \tag{3.32}$$

is the recoil energy.

Using tabulated values of the polarizabilities for LiCs[28] as given in Table 3.1, we find that, for a reasonable lattice height $V_x^{(00)}/E_R \simeq 10$, the tunneling term for the $|11\rangle$ state is only about 20% of that in the $|00\rangle$ state, as shown in Figure 3.1. For LiCs in a red-detuned optical lattice of wavelength $\lambda = 985$nm, $E_R = 2\pi \times 1.46\hbar$ kHZ. Typical values of the lattice heights are $V_x \sim 10E_R$, $V_y, V_z \sim 25E_R$ [29].

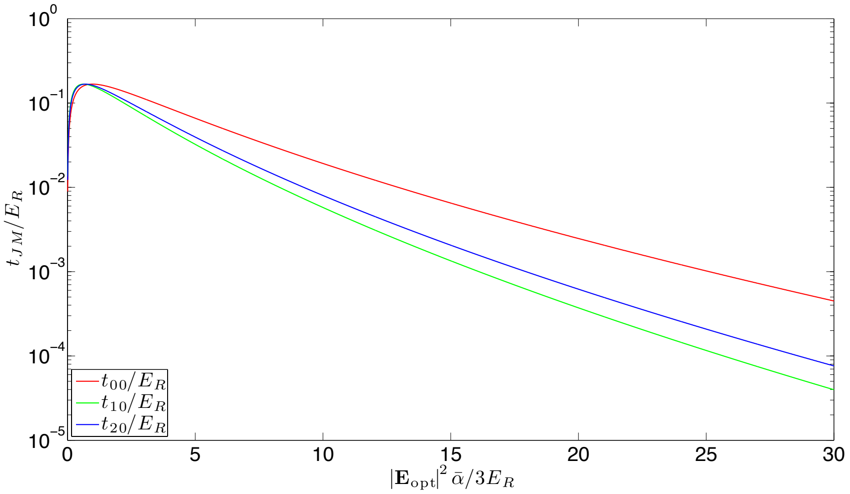


Figure 3.1: Dependence of the field-free tunneling (hopping) coefficient on rotational state and lattice height.

We reiterate that the above matrix elements and tunneling energies $\{t_{JM}\}$ have been computed in the field-free basis for simplicity. To transform to the dressed basis, we use the unitary matrix with dressed eigenvectors as columns, recovering Eq. (3.9), where the tunneling matrix element is no longer diagonal in J.

3.2.4 Dipole-Dipole Interactions

The induced dipoles from the DC field give rise to a resonant dipole-dipole interaction. The Hamiltonian for this interaction in the two-site dressed basis spanned by $|\mathcal{E}; J_1M_1J_2M_2\rangle$ is

$$\hat{H}_{\rm dd} = \frac{1}{2}\sum_{\substack{J_1,\, J_1',\, J_2,\, J_2'\\ M,\, M'}} U_{dd}^{\substack{J_1,\, J_1',\, J_2,\, J_2'\\ M,\, M'}} \sum\nolimits_{\langle i,i'\rangle} \hat{a}^{\dagger}_{iJ_1M}\hat{a}_{iJ_1'M}\hat{a}^{\dagger}_{i'J_2M'}\hat{a}_{i'J_2'M'}\,, \tag{3.33}$$

where we have defined

$$\begin{aligned} U_{dd}^{\substack{J_1,\, J_1',\, J_2,\, J_2'\\ M,\, M'}} &\equiv \int d\mathbf{r}d\mathbf{r}'\, w^{\star}_{J_1'M}\left(\mathbf{r}-\mathbf{r}_i\right) w^{\star}_{J_2'M'}\left(\mathbf{r}'-\mathbf{r}_{i+1}\right) \\ &\times H_{dd}\left(\mathbf{r}-\mathbf{r}'\right) w_{J_1M}\left(\mathbf{r}-\mathbf{r}_i\right) w_{J_2M'}\left(\mathbf{r}'-\mathbf{r}_{i+1}\right)\,, \end{aligned} \tag{3.34}$$

and for notational simplicity we have suppressed the $\mathcal{E}$ subscripts. Note that because of our choice of polarizations of the optical lattice and AC and DC electric fields, $M_1 = M_2 \equiv M$ and $M_1' = M_2' \equiv M'$.

The resonant dipole-dipole interaction between two permanent dipoles d_1 and d_2 whose respective centers of mass are separated by a vector $\mathbf{R}$ in the space-fixed frame is

$$\hat{H}_{dd} \;=\; \frac{\hat{\mathbf{d}}_{\mathbf{1}}\cdot\hat{\mathbf{d}}_{\mathbf{2}}-3\left(\mathbf{e}_R\cdot\hat{\mathbf{d}}_{\mathbf{1}}\right)\left(\hat{\mathbf{d}}_{\mathbf{2}}\cdot\mathbf{e}_R\right)}{R^3}\,, \tag{3.35}$$

where $\mathbf{e}_R$ is a unit vector in the direction of $\mathbf{R}$. Using standard angular momentum recoupling we recast this in spherical tensor notation as

$$H_{dd} \;=\; -\frac{\sqrt{6}}{R^3}\sum_{\mu}\left(-1\right)^{\mu}C^{(2)}_{-\mu}\left(\mathbf{R}\right)\left[\hat{\mathbf{d}}_1\otimes\hat{\mathbf{d}}_2\right]^{(2)}_{\mu}\,, \tag{3.36}$$

where $(T)^{(k)}_q$ denotes the component of the rank-k spherical tensor T that has projection q along $\mathbf{R}$, $C^{(j)}_m\left(\mathbf{R}\right)$ is an unnormalized spherical harmonic in the polar coordinates defined with respect to $\mathbf{R}$, and we have defined the tensor product of the vector operators $\hat{\mathbf{d}}_1$ and $\hat{\mathbf{d}}_2$ as

$$\left[\hat{\mathbf{d}}_1\otimes\hat{\mathbf{d}}_2\right]^{(k)}_{q}\equiv\sum_m\langle 1,m,1,q-m|kq\rangle\left(\hat{\mathbf{d}}_1\right)^{(1)}_{m}\left(\hat{\mathbf{d}}_2\right)^{(1)}_{q-m}\,. \tag{3.37}$$

In the last line, $\langle j_1,m_1,j_2,m_2|J,M\rangle$ is a Clebsch-Gordan coefficient. We now take matrix elements of Eq. (3.36) in the two dressed-molecule basis $|\mathcal{E};J_1M_1,J_2M_2\rangle$, where molecule 1 is on site i and molecule 2 is on site $i+1$, yielding

$$\begin{aligned}\langle\mathcal{E};J_1'M_1',J_2'M_2'|\hat{H}_{dd}|\mathcal{E};J_1M_1,J_2M_2\rangle &= -\frac{\sqrt{6}}{R^3}\sum_{\mu}\left(-1\right)^{\mu}C^{(2)}_{-\mu}\left(\mathbf{R}\right)\sum_m\langle 1,m,1,\mu-m|2\mu\rangle\\ \times\langle\mathcal{E};J_1'M_1'|\left(\hat{\mathbf{d}}_1\right)^{(1)}_{m}|\mathcal{E};J_1M_1\rangle\langle\mathcal{E};J_2'M_2'|\left(\hat{\mathbf{d}}_2\right)^{(1)}_{\mu-m}|\mathcal{E};J_2M_2\rangle\,.\end{aligned} \tag{3.38}$$

Because our DC field is polarized along z, only $(\hat{\mathbf{d}}_1)_0^{(1)}$ and $(\hat{\mathbf{d}}_2)_0^{(1)}$ matrix elements are nonzero, enforcing $\mu = 0$, $m = 0$. With this in mind, the interaction takes the particularly simple form

$$\begin{aligned}
&\langle \mathcal{E}; J_1' M_1', J_2' M_2' | \hat{H}_{dd} | \mathcal{E}; J_1 M_1, J_2 M_2 \rangle = -\tfrac{\sqrt{6}}{R^3} C_0^{(2)}\left(\mathbf{R}\right) \\
&\times \langle 1,0,1,0|20\rangle \langle \mathcal{E}; J_1' M_1 | \left(\hat{\mathbf{d}}_1\right)_0^{(1)} | \mathcal{E}; J_1 M_1 \rangle \langle \mathcal{E}; J_2' M_2 | \left(\hat{\mathbf{d}}_2\right)_0^{(1)} | \mathcal{E}; J_2 M_2 \rangle \qquad (3.39) \\
&= \langle \mathcal{E}; J_1' M_1 | \left(\hat{\mathbf{d}}_1\right)_0^{(1)} | \mathcal{E}; J_1 M_1 \rangle \langle \mathcal{E}; J_2' M_2 | \left(\hat{\mathbf{d}}_2\right)_0^{(1)} | \mathcal{E}; J_2 M_2 \rangle \left(\tfrac{1 - 3\cos^2\theta}{R^3}\right) . \qquad (3.40)
\end{aligned}$$

The intermolecular axis plays a crucial role in the sign of the interaction. Two molecules oriented along the intermolecular axis attract if their dipoles are parallel and repel if their dipoles are antiparallel. Two molecules oriented perpendicular to the intermolecular axis, on the other hand, repel if their dipoles are parallel and attract if their dipoles are antiparallel. The DC field that orients the molecules in our setup is polarized along z, perpendicular to the intermolecular quasi-1D axis x. This gives rise to repulsive interactions for positive dipole matrix elements. With this geometry the dipole potential becomes

$$\begin{aligned}
&\langle \mathcal{E}; J_1' M_1', J_2' M_2' | \hat{H}_{dd} | \mathcal{E}; J_1 M_1, J_2 M_2 \rangle \\
&= \frac{1}{R^3} \langle \mathcal{E}; J_1' M_1 | \left(\hat{\mathbf{d}}_1\right)_0^{(1)} | \mathcal{E}; J_1 M_1 \rangle \langle \mathcal{E}; J_2' M_2 | \left(\hat{\mathbf{d}}_2\right)_0^{(1)} | \mathcal{E}; J_2 M_2 \rangle \, , \qquad (3.41)
\end{aligned}$$

yielding

$$U_{dd}^{\substack{J_1,\, J_1',\, J_2,\, J_2' \\ M,\, M'}} = \frac{8}{\lambda^3} \langle \mathcal{E}; J_1' M_1 | \left(\hat{\mathbf{d}}_1\right)_0^{(1)} | \mathcal{E}; J_1 M_1 \rangle \langle \mathcal{E}; J_2' M_2 | \left(\hat{\mathbf{d}}_2\right)_0^{(1)} | \mathcal{E}; J_2 M_2 \rangle \, , \qquad (3.42)$$

where λ is the wavelength of the optical lattice.

3.2.5 Energy Scales

We proceed to clarify the energy scales associated with each term in Eq. (3.1). Between previous discussion in Sec. 3.2 and that of appendix 3.6, all terms in Eq. (3.1) are now clearly defined. The energy scales of the dressed basis are B, the rotational constant, which is roughly $60\hbar$ GHz, and $d\mathcal{E}_{\mathrm{DC}}$, which is of order $1-10B$. The DC term has no length scale associated with it because the field is uniform, and the length scale of the rotational term is the internuclear separation, on the order of angstroms. The relative contribution of the DC electric field and rotational terms in Eq. (3.1) are expressed through the dimensionless parameter

$$\beta_{\mathrm{DC}} \equiv d\mathcal{E}_{\mathrm{DC}}/B, \tag{3.43}$$

the ratio of the DC field energy to the rotational level splitting.

The energy scales of the AC term are $\hbar\omega$, where ω is the angular frequency of the driving field, and $d\mathcal{E}_{\mathrm{AC}}$. The scale $\hbar\omega$ is of order $2B$ for small $\beta_{\mathrm{DC}} \ll 1$, and of order $B\sqrt{\beta_{\mathrm{DC}}}$ for large $\beta_{\mathrm{DC}} \gg 1$. The AC field energy $d\mathcal{E}_{\mathrm{AC}}$ is of order $0.5\hbar\omega$. The single-molecule timescale associated with $d\mathcal{E}_{\mathrm{AC}}$ is the Rabi period, the time it takes for the population of a two-level system to cycle once, as seen in Figure Figure 3.10(a). In real time, this is on the order of 10ps for the parameters in the preceding paragraph. The timescale associated with ω is the timescale on which the small oscillations in Figure Figure 3.10(a) occur, of order 0.5ps. The length scale of the AC field is on the order of centimeters, and so we can neglect this in light of the micron length scale of the trap.

The tunneling term has several scales. The optical lattice near the point of confinement has a length scale given by the harmonic oscillator length $l_{\mathrm{ho,x}}^{(00)}$ ~100nm and an energy scale of $E_R \approx$1.4$\hbar$ kHz. The energy scales of the tunneling operator proper are given by the $\{t_{JJ'M}\}$ which are of order 10^{-1}-$10^{-2}E_R$ ~100$\hbar$ Hz for the given recoil energy.

There are also many scales for the dipole term. For the B and d specified in the first paragraph of this section and $\beta_{\mathrm{DC}}=1.9$, the characteristic length scale where the dipole-dipole energy becomes comparable to the rotational energy is

$$r_B \equiv \left(\left|\langle\mathcal{E};00|\hat{\mathbf{d}}|\mathcal{E};00\rangle\right|^2/B\right)^{\frac{1}{3}}, \tag{3.44}$$

approximately 348 Bohr radii (18.4nm). Outside this region the Born-Oppenheimer adiabatic approximation is easily fulfilled [6]. Since the length scale of our optical lattice is of order μm, we are justified in working within the Born-Oppenheimer framework. For the same parameters, the length scale where the off-resonant van der Waals potential $C_6/r^6 \approx -d^4/(6Br^6)$ becomes comparable to the dipole-dipole interaction is

$$r_{\mathrm{vdW}} \equiv (2\,|C_6|\,/\left|\langle\mathcal{E};00|\hat{\mathbf{d}}|\mathcal{E};00\rangle\right|^2)^{\frac{1}{3}}. \tag{3.45}$$

This length is very small, on the order of tens to hundreds of Bohr radii. Outside of this region the resonant dipole potential dominates and the intermolecular force is repulsive. This repulsion enforces the hard-core limit. The energy scale of the dipole-dipole force is $\left|\langle\mathcal{E};00|\hat{\mathbf{d}}|\mathcal{E};00\rangle\right|^2/\lambda^3 \sim 1.2\hbar$ kHz, with higher J being an order of magnitude or so lower for small β_{DC}, and of the same order for large β_{DC} (see Figure 3.8(a)).

To summarize, the scales of the problem are shown in Table 3.2.

3.2.6 Novel Features of the Molecular Hubbard Hamiltonian

The MHH, Eq. (3.1), has a number of novel features which distinguish it from typical Hubbard and extended Hubbard models [2, 30]. First, the tunneling energies $\{t_{J,J'M}\}$ not only depend on the rotational level J, M but even change rotational states from J to J'. This is due both to the polarizability tensor's dependence on rotational level, and to the dressed basis. This differs from other Hubbard models which consider spin degrees of freedom, as tunneling does not occur between spin states – hopping does not cause spin transitions. If we consider populating a single mode (e.g. $J=0$,

Table 3.2: Comparison of energy and length scales for the Molecular Hubbard Hamiltonian of Eq. (3.1).

Term	Length scale	Energy scale
Rotation	internuclear distance ~ 1 Å	$B \sim 60\hbar$ GHz $\approx 2\text{cm}^{-1}$
DC field	N/A, uniform	$d\mathcal{E}_{\text{DC}} \sim 120\hbar$ GHz $\approx 4\text{cm}^{-1}$
AC field	$2\pi c/\omega \sim 1\text{cm}$	$\hbar\omega \sim 30\hbar$ GHz $\approx 1\text{cm}^{-1}$
Kinetic	$l^{(00)}_{\text{ho,x}} \sim 100\text{nm}$	$E_R \sim 1.46\hbar$ kHz
Tunneling	Lattice spacing$\sim 1\mu$m	$\{t_{J'JM}\} \sim 100\hbar$ Hz
Resonant Dipole-Dipole	energy comparable to B at $r_B \simeq 348$ Bohr radii	$\left\lvert\langle\mathcal{E};00\lvert\hat{\mathbf{d}}\rvert\mathcal{E};00\rangle\right\rvert^2/(1\mu\text{m})^3 \sim 1.2\hbar$ kHz for nearest neighbors

$M = 0$) in the $\Omega \rightarrow 0$ limit, then Eq. (3.1) becomes the extended Bose-Hubbard Hamiltonian, and the phase diagram is known [10, 31]. This gives ideas of how to characterize the static phases of the MHH. However, because the tunneling energy depends on J, the borders of the phase diagram will depend on the rotational state of the system. We will discuss this property and provide an application in Sec. 3.4.

Second, the Hamiltonian is fundamentally time-dependent because it is a driven system. This allows for the study of dynamic quantum phases, requiring the concept of a quantum phase diagram to be generalized to an inherently time-dependent picture. In a case study for hard core bosonic molecules at half filling presented in Sec. 3.4, we show that the MHH has emergent timescales.

3.3 Methods

3.3.1 Time-Evolving Block Decimation

The Time-evolving Block Decimation algorithm (TEBD) is a new method [13, 14] designed to study the dynamics of entangled quantum systems. The essential idea of TEBD is to provide a moving "spotlight" in Hilbert space which tracks a dynamical system. The portion of the Hilbert space so illuminated is an exponentially small fraction of the full Hilbert space; this is justified by the fact that real, physical quantum many body systems, especially in real materials, typically explore only a small, lowly-entangled part of the total Hilbert space.

In fact, TEBD moves the full quantum many-body problem from the NP-complete complexity class to the P class through an exponential reduction in the number of parameters needed to represent the many body state. We can understand the possibility of this reduction through an analogy to image compression. Present digital cameras are capable of producing a roughly 3000 × 3000 array of pixels. Downloading the images from such a camera, one notices that there are far less than 10 Megapixels worth of data per image. Image compression algorithms such as JPEG produce images of remarkable quality with only a small fraction of the raw data. The reason that these algorithms are so effective is that a physical image, as opposed to a random 2D pixel array, is not the "most common" or most probable image; it contains a great deal of structure and regularity. In the same way, physical states in Hilbert space tend to be lowly entangled (by some entanglement measure), even though a general state in Hilbert space has a much larger probability of being highly entangled. There is no general proof of this fact, just as there is no guarantee that an image will come out perfectly crisp after JPEG compression; it is simply a trend observed in many-body quantum systems.

To be slightly more specific, TEBD performs a partial trace over a particular bipartite splitting of the lattice, and then keeps the χ largest eigenvalues of the resulting reduced density matrix. The cut-off parameter χ is based on the Schmidt measure [15], and so it also serves as a measure of the degree of spatial entanglement. This idea is not unique to TEBD. In fact, the density matrix renormalization group (DMRG) method first proposed by White [32] did something analogous years before. The innovation of TEBD is that at each time step it re-optimizes the truncated basis (thus the "moving spotlight"). The Schmidt number is just the number of nonzero eigenvalues in the reduced density matrix, and so is an entanglement measure natural to quantum many body systems. The parameter χ is the number of non-zero eigenvalues in the reduced density matrix that TEBD retains. It is the principal convergence parameter of the algorithm, both in entanglement and in time. Although the time-propagation method we use is Trotter-Suzuki [33], it turns out that, due to a normalization drift, χ controls convergence at long times.

With χ interpreted as an entanglement measure, we can say that TEBD treats the system not as a wavefunction in a d^L-dimensional Hilbert space (L is the number of lattice sites), but as a collection of wavefunctions in d^2-dimensional two-site spaces that are weakly entangled with the environment created by the rest of the system. To facilitate this viewpoint, we replace the d^L coefficients of the full many-body wavefunction with L sets of $(d\chi^2 + \chi)$ coefficients corresponding to the wavefunctions of each bipartite splitting. The most computationally expensive portion of the TEBD algorithm is typically the diagonalization of these local coefficient matrices at a cost of $\mathcal{O}\left(d^3\chi^3\right)$. Looping over all $L-1$ bipartite splittings and evolving the system for a total time t_f in time steps of length δt, one obtains an asymptotic scaling of $\mathcal{O}\left(L\frac{t_f}{\delta t}d^3\chi^3\right)$.

This scaling can be greatly improved by the presence of conserved quantities. When a conserved quantity exists in the system we are able to diagonalize reduced density matrices corresponding to distinct values of this conserved quantity independently, which can result in significantly smaller reduced density matrices to diagonalize. Implementing this idea, scalings of $\mathcal{O}\left(\chi^2\right)$ have been reported for fixed d [34]. In addition, conserved quantities in the presence of selection rules can reduce the local dimension. For example, in the case of the MHH, z-polarized electric fields disallow transitions from a particular M to any other. If we begin with all molecules in a particular M state, this allows us to restrict our attention only to states with this M. In our numerics we conserve both the projection M, and the total number N. Furthermore, to match our hard core requirement, we allow only zero or one molecules per site, so that the local dimension is $d \leq J_{\max} + 1$, $J_{\max}$ being the magnitude of the greatest angular momentum that we consider (note that the local dimension d, mentioned only here in Sec. 3.3.1, bears no relation to the permanent electric dipole moment d used throughout the rest of our treatment).

A more detailed description of TEBD can be found in Ref. [23]. We also recommend Ref. [35], besides Vidal's original papers [13, 14].

3.3.2 Quantum Measures

We use a suite of quantum measures to characterize the reduced MHH, Eq. (3.50) below. The few-body measures we use are $\langle \hat{n}_i^J \rangle$, the number in the J^{th} rotational state on the i^{th} site, $E \equiv \langle \hat{H} \rangle$, the expectation of the energy, and $\frac{1}{L}\langle \hat{n}^J \rangle$, the average number in the J^{th} rotational state per site (L is the number of lattice sites). The latter is a J-dependent filling factor. The many body measures we use include the density-density correlation between rotational modes J_1 and J_2 evaluated at the middle site

$$g_2^{(J_1 J_2)}\left(\lfloor \frac{L}{2} \rfloor, i\right) \equiv \langle \hat{n}_{\lfloor \frac{L}{2} \rfloor}^{(J_1)} \hat{n}_i^{(J_2)} \rangle - \langle \hat{n}_{\lfloor \frac{L}{2} \rfloor}^{(J_1)} \rangle \langle \hat{n}_i^{(J_2)} \rangle, \tag{3.46}$$

where $\lfloor q \rfloor$ is the floor function, defined as the greatest integer less than or equal to q. As an entanglement measure we use the Meyer Q-measure [36–38]

$$Q \equiv \tfrac{d}{d-1}\left[1 - \tfrac{1}{L}\textstyle\sum_{k=1}^{L} \mathrm{Tr}\left(\hat{\rho}^{(k)}\right)^2\right], \tag{3.47}$$

where $\hat{\rho}^{(k)}$ is the single-site density matrix obtained by tracing over all but the k^{th} lattice site, and the factor outside of the bracket is a normalization factor (d is the on-site dimension). This gives an average measure of the entanglement of a single site with the rest of the system. The Q-measure can also be interpreted as the average local impurity (recall that the $\mathrm{Tr}(\hat{\rho}^2) = 1$ if and only if $\hat{\rho}$ is a pure state).

To determine what measures we can use to ascertain the static phases of our model we reason by analogy with the extended Bose-Hubbard Hamiltonian where we know that the possible static phases are charge density wave, superfluid, supersolid, and Bose metal [10]. The charge density wave is an insulating phase appearing at half integer fillings which has a wavelength of two sites. Like the Mott insulating phase, it has an excitation gap and is incompressible. While the extended Bose-Hubbard Hamiltonian has only one charge density wave phase due to the presence of only one species, the MHH has the possibility of admitting several charge density wave phases due to the presence of multiple rotational states. As such, we define the structure factor

$$S_{\pi}^{(J_1 J_2)} = \tfrac{1}{N} \sum_{ij} (-1)^{|i-j|} \langle \hat{n}_i^{(J_1)} \hat{n}_j^{(J_2)} \rangle , \tag{3.48}$$

where N is the total number of molecules. We recognize this object as the spatial Fourier transform of the equal-time density-density correlation function between rotational states J_1 and J_2, evaluated at the edge of the Brillouin zone. This measure is of experimental interest because it is proportional to the intensity in many scattering experiments, e.g. neutron scattering [39]. Crystalline order between rotational states J_1 and J_2 is characterized by a nonzero structure factor $S_{\pi}^{(J_1 J_2)}$. The charge density wave is the phase with crystalline order but no off-diagonal long-range order as quantified by the superfluid stiffness of rotational state J

$$\rho_s^{(J)} = \lim_{\phi \to 0} L \frac{\partial^2 E^{(J)}(\phi, L)}{\partial \phi^2} \tag{3.49}$$

(note that ρ_s bears no relation to the density matrix $\hat{\rho}$). If both the structure factor and the superfluid stiffness are nonzero, the phase is called supersolid. If both the structure factor and the superfluid stiffness are zero, the phase is called Bose Metal. Finally, if the structure factor is zero and the superfluid stiffness is nonzero, the phase is superfluid. In 1D systems with short-range interactions the structure factor is zero in the thermodynamic limit and the entire superfluid phase is critical, thus there are formally no order parameters [10]. Superfluidity is instead signaled by a diverging correlation length and solid order by slow power law decay of the density-density correlator.

3.4 Case Study: Hard Core Bosonic Molecules at Half Filling

In the following, we consider a particular case of Eq. (3.1) for dynamical study. We choose the hard core case, which can occur naturally due to strong on-site dipole-dipole interactions, and half filling, which is an interesting point in a number of models, including the repulsive Fermi-Hubbard Hamiltonian and the extended Bose-Hubbard Hamiltonian discussed in Sec. 3.3.2. For example, in the latter case, the charge-density-wave phase requires a minimum of half-filling [10].

If we assume that our system begins in its ground state ($J = 0$, $M = 0$) we need only include states which have a dipole coupling to this state. For z-polarized DC and AC fields, this means we only consider $M = 0$ states, yielding the reduced Hamiltonian

$$\begin{aligned}\hat{H} = &-\sum_{JJ'} t_{JJ'} \sum_{\langle i,i'\rangle} \left(\hat{a}^{\dagger}_{i',J'}\hat{a}_{iJ} + \text{h.c.}\right) + \sum_{J} E_J \sum_{i} \hat{n}_{iJ} \\ &-\pi \sin\left(\omega t\right) \sum_{J} \Omega_J \sum_{i} \left(\hat{a}^{\dagger}_{iJ}\hat{a}_{iJ+1} + \text{h.c.}\right) \\ &+\frac{1}{2} \sum_{J_1,J_1',J_2,J_2'} U_{dd}^{J_1,J_1',J_2,J_2'} \sum_{\langle i,i'\rangle} \hat{a}^{\dagger}_{iJ_1}\hat{a}_{iJ_1'}\hat{a}^{\dagger}_{i'J_2}\hat{a}_{i'J_2'}\,. \end{aligned} \tag{3.50}$$

This is the specific case of the MHH that we study using TEBD.

A matter of practical concern, as apparent in Table 3.2, is the large disparity between the timescales of the first three (Rotational, DC, and AC) and the last three (kinetic, tunneling, and Dipole-Dipole) terms. The accumulation of error resulting from truncating the Hilbert space at each TEBD timestep causes the algorithm to eventually fail after a certain "runaway time," making studies over long times intractable [40]. This invites a multiscale approach in the future [41, 42]. In our current numerics we artificially increase the recoil energy and dipole-dipole potential to be of the order of the rotational constant in order to study Eq. (3.50) using TEBD. In particular, we take

$$U_{dd}^{J_1,J_1',J_2,J_2'} = \tfrac{10B}{d^2}\langle \mathcal{E};J_1'|\hat{\mathbf{d}}|\mathcal{E};J_1\rangle\langle \mathcal{E};J_2'|\hat{\mathbf{d}}|\mathcal{E};J_2\rangle\,, \tag{3.51}$$

$$\begin{aligned} t_J &= 10B\left[\eta\left(1+2\tfrac{\Delta\alpha}{\bar{\alpha}}\tfrac{J(J+1)}{(2J+1)(2J+3)}\right)\right]^{1.051} \\ &\quad\times\exp\left[-2.121\sqrt{\eta\left(1+2\tfrac{\Delta\alpha}{\bar{\alpha}}\tfrac{J(J+1)}{(2J+1)(2J+3)}\right)}\right]\,, \end{aligned} \tag{3.52}$$

where the dimensionless variable η becomes an ersatz "lattice height." To see the scaling more explicitly, we compare the above with the actual expressions for the MHH parameters

$$U_{dd}^{J_1,J_1',J_2,J_2'} = \tfrac{8}{\lambda^3}\langle \mathcal{E};J_1'|\hat{\mathbf{d}}|\mathcal{E};J_1\rangle\langle \mathcal{E};J_2'|\hat{\mathbf{d}}|\mathcal{E};J_2\rangle \tag{3.53}$$

$$= \left(\tfrac{2mE_Rd^{4/3}}{\hbar^2\pi^2}\right)^{\frac{3}{2}}\langle \mathcal{E};J_1'|\hat{\mathbf{d}}|\mathcal{E};J_1\rangle\langle \mathcal{E};J_2'|\hat{\mathbf{d}}|\mathcal{E};J_2\rangle/d^2\,, \tag{3.54}$$

$$t_{JM} \approx 1.397E_R\left(\tfrac{|\mathbf{E}_{\mathrm{opt}}|^2\bar{\alpha}}{3E_R}\left[1+2\tfrac{\Delta\alpha}{\bar{\alpha}}\tfrac{J(J+1)-3M^2}{(2J-1)(2J+3)}\right]\right)^{1.051} \tag{3.55}$$

$$\times\exp\left(-2.121\sqrt{\tfrac{|\mathbf{E}_{\mathrm{opt}}|^2\bar{\alpha}}{3E_R}\left[1+2\tfrac{\Delta\alpha}{\bar{\alpha}}\tfrac{J(J+1)-3M^2}{(2J-1)(2J+3)}\right]}\right)\,. \tag{3.56}$$

If we now scale E_R to be $10B/1.397$ and set d such that $\left[2mE_Rd^{4/3}/\left(\hbar^2\pi^2\right)\right]^{\frac{3}{2}} = 10B$ for this E_R, we recover Eqs. (3.51) and (3.52) provided we make the definition

$$\eta \equiv -\left|\mathbf{E}_{\mathrm{opt}}\left(\mathbf{x}\right)\right|^2\bar{\alpha}/\left(3E_R\right) = V_x^{(JM)}\bar{\alpha}/\left(3E_R\alpha_{JM}^{(t)}\right)\,. \tag{3.57}$$

Since this dimensionless parameter plays the same role as the quasi-1D lattice height scaled to the recoil energy did in the actual MHH, we refer to it as the lattice height. For the polarizability tensor, we choose $\Delta\alpha/\bar{\alpha} = 165.8/237$, corresponding to LiCs [28]. This rescaling does not change the qualitative static and dynamical features of Eq. (3.50); it only makes Eq. (3.50) treatable directly by TEBD, without multiscale methods. In the future, we plan to apply multiscale methods to determine the emergent timescales for experimentally relevant parameters.

First, we point out that if we consider populating a single rotational state (e.g. $J = 0$, $M = 0$) in the $\Omega \to 0$ limit, then Eq. (3.50) becomes the extended Bose-Hubbard Hamiltonian, and the phase diagram is known [10, 31]. Because the tunneling energy is different for different rotational states (see Eq. (3.31)) and this difference depends only on the properties of the polarizability tensor, we can relate the borders of the phase diagram for different rotational states to properties of the polarizability tensor. The MHH thus gives a means to measure the polarizability tensor, a standing issue in experiments [43]. Our calculations in Sec. 3.2 can be used to compare directly to the phase diagram from the literature [10, 31]. In fact, this aspect of our work, unlike the simulations below, is not restricted to 1D.

However, our main focus at present is on the dynamics of the MHH. In the following numerical study, we explore dynamics as a function of the physical characteristics of the lattice, namely, number of sites L and effective lattice height η. Specifically, we study $L = 9$, 10, and 21 lattice sites with N=4, 5, and 10 molecules, respectively, and η ranging from 1 to 10. We fix the dipole-dipole term as in Eq. (3.51), and fix the DC field parameter to be $\beta_{\mathrm{DC}} = 1.9$. While $\beta_{\mathrm{DC}} = 1.9$ may not correspond to a physically realizable situation, its exploration provides insight into the MHH.

The Rabi oscillations between the $J = 0$ and the $J = 1$ states damp out exponentially in the rotational time $t_r \equiv Bt/\hbar$ as

$$\langle \hat{n}_0 \rangle = a_0 - b_0 \, e^{-t_r/\tau} \cos\left(c_0 t_r\right) , \tag{3.58}$$

$$\langle \hat{n}_1 \rangle = a_1 - b_1 \, e^{-t_r/\tau} \cos\left(c_1 t_r\right) , \tag{3.59}$$

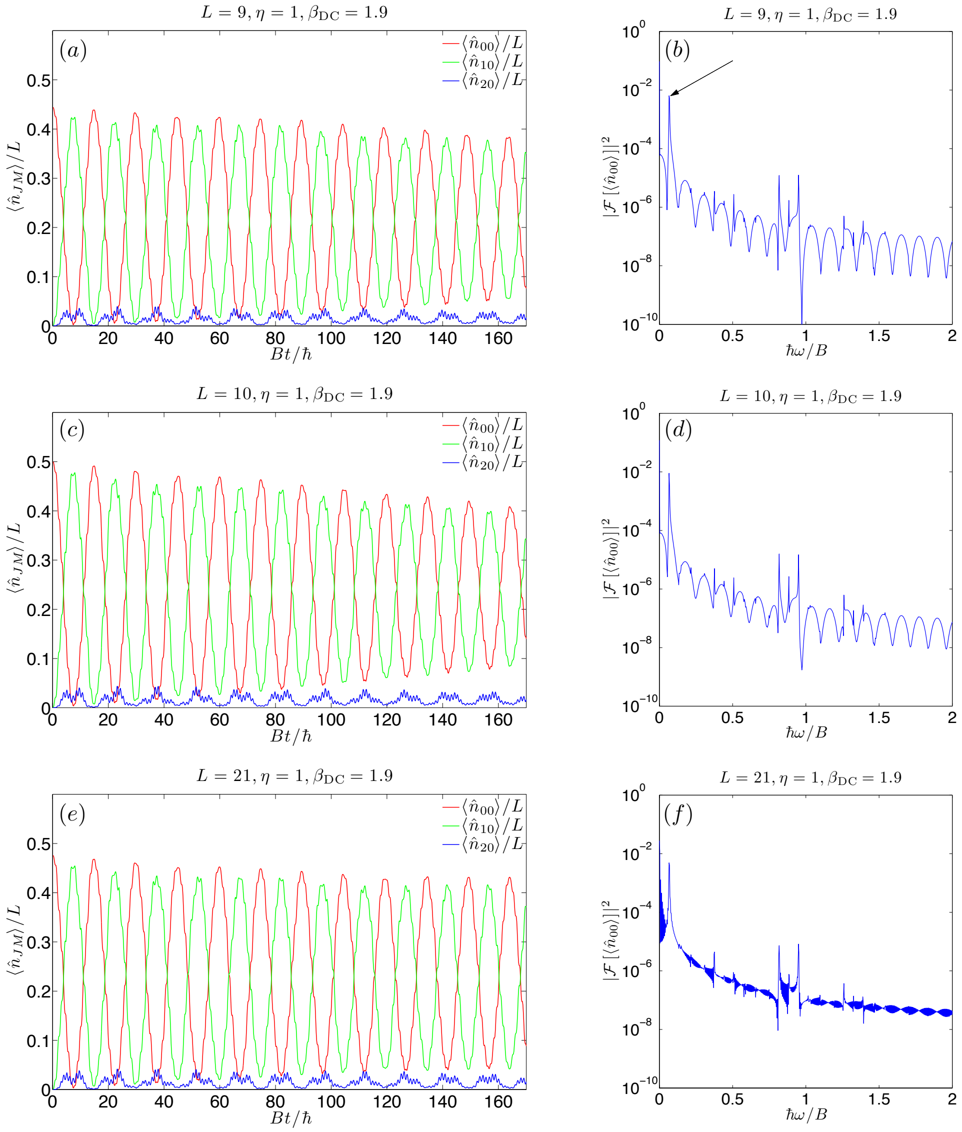


Figure 3.2: Dependence of site-averaged number on lattice size L. For this set of parameters, the site-averaged $J = 0$ and $J = 1$ populations appear to asymptotically approach quarter filling. The $J = 2$ mode is populated slightly by off resonant AC couplings. The peak near the left side of the Fourier transform plots is the Rabi frequency Ω_{00}, denoted by an arrow. (a) Site-averaged population vs. rotational time for 9 sites. Note the general theme; a gradual decrease (increase) of the maxima (minima) of oscillations. (b) Squared modulus of Fourier transform of site-averaged $J = 0$ population vs. rotationally scaled frequency for $L = 9$ sites. The arrow denotes the Rabi frequency Ω_{00}. (c) Site-averaged population vs. rotational time for 10 sites. Note that there is no significant difference between an odd and even number of sites. (d) Squared modulus of Fourier transform of site-averaged $J = 0$ population vs. rotationally scaled frequency for $L = 10$ sites. (e) Site-averaged population vs. rotational time for 21 sites. Note that there is no significant difference between this and the smaller system sizes. (f) Squared modulus of Fourier transform of site-averaged $J = 0$ population vs. rotationally scaled frequency for $L = 21$ sites.

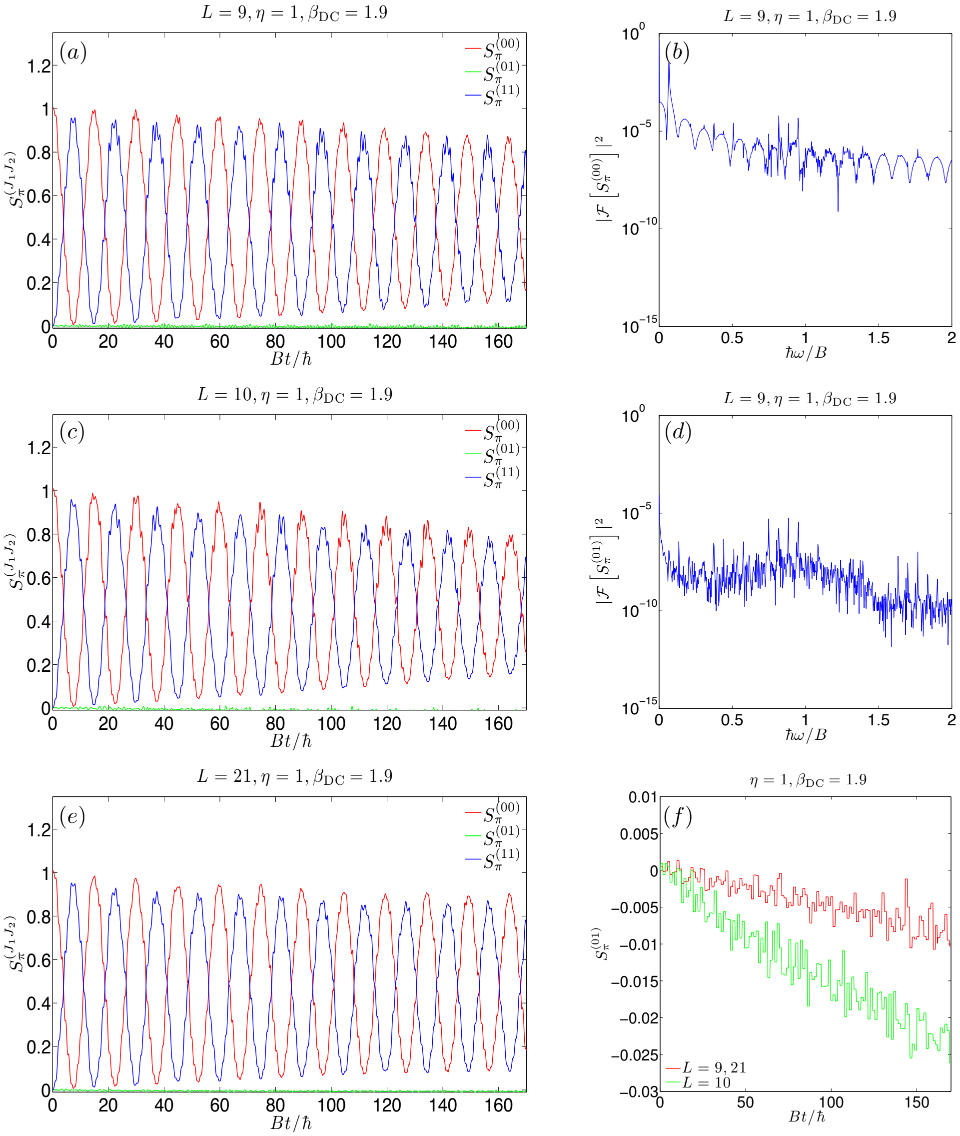


Figure 3.3: Dependence of structure factors within and between rotational states J on the number of lattice sites. We do not consider the off-resonant $J = 2$ and higher rotational states because they have a very small occupation; $J = 2$ is shown explicitly in Figure 3.2. (a) Structure factors vs. rotational time for 9 sites. Note the similar asymptotic behavior to the populations in Figure 3.2(a). (b) Squared modulus of Fourier transform of $S_\pi^{(00)}$ vs. rotationally scaled frequency for $L = 9$ sites. Note the similarity with Figure 3.2(b) above. (c) Structure factors vs. rotational time for 10 sites. There is no significant difference in the $S_\pi^{(00)}$ and $S_\pi^{(11)}$ between even and odd L. For the difference in $S_\pi^{(01)}$, see Figure 3.3(f). (d) Squared modulus of Fourier transform of $S_\pi^{(10)}$ vs. rotationally scaled frequency for $L = 9$ sites. Note the absence of the Rabi frequency. (e) Structure factors vs. rotational time for 21 sites. Note the lack of significant difference with the smaller odd system size. (f) Comparison of the $S_\pi^{(01)}$ correlation structure factor for odd and even numbers of sites. Note that the even site (exactly half filling) structure factor grows faster and larger than the odd site (slightly less than half filling) structure factor.

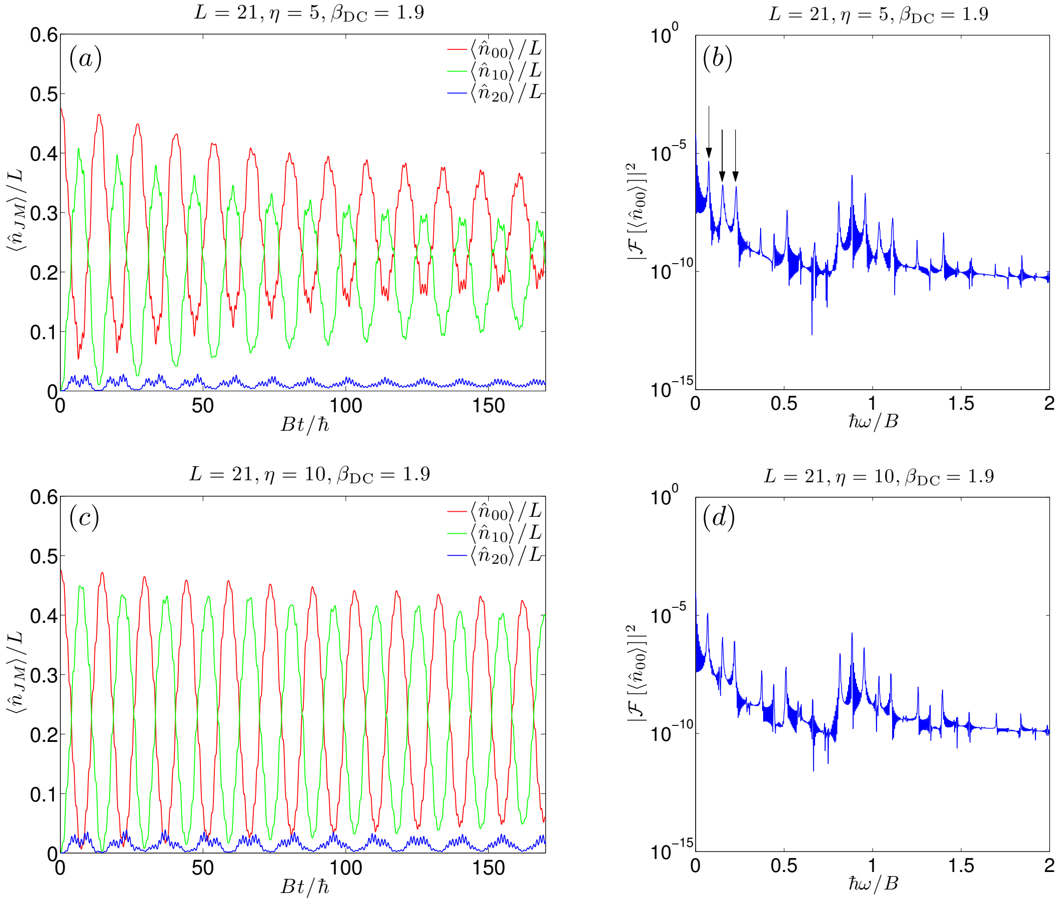


Figure 3.4: Dependence of the asymptotic behavior of rotational state populations on the lattice height η. (a) Site-averaged population vs. rotational time for 21 sites with $\eta = 5$. Note that the $J = 0$ and $J = 1$ states now appear to converge to different fillings. (b) Squared modulus of Fourier transform of $\langle \hat{n}_{00} \rangle$ vs. rotationally scaled frequency for $L = 21$ sites and $\eta = 5$. Note the presence of several new frequencies not observed in the $\eta = 1$ case (Figure 3.2(f)). In particular, Ω_{00}, $2\Omega_{00}$, and $3\Omega_{00}$, are denoted by arrows. (c) Site-averaged population vs. rotational time for 21 sites with $\eta = 10$. Note the similarity to the $\eta = 1$ case (Figure 3.2(e)) and the difference from the $\eta = 5$ case (Figure 3.4(a))–the asymptotic behavior is *not* a monotonic function of the lattice height. (d) Squared modulus of Fourier transform of $\langle \hat{n}_{00} \rangle$ vs. rotationally scaled frequency for $L = 21$ sites and $\eta = 10$. Note that the frequencies that emerged during $\eta = 5$ have persisted.

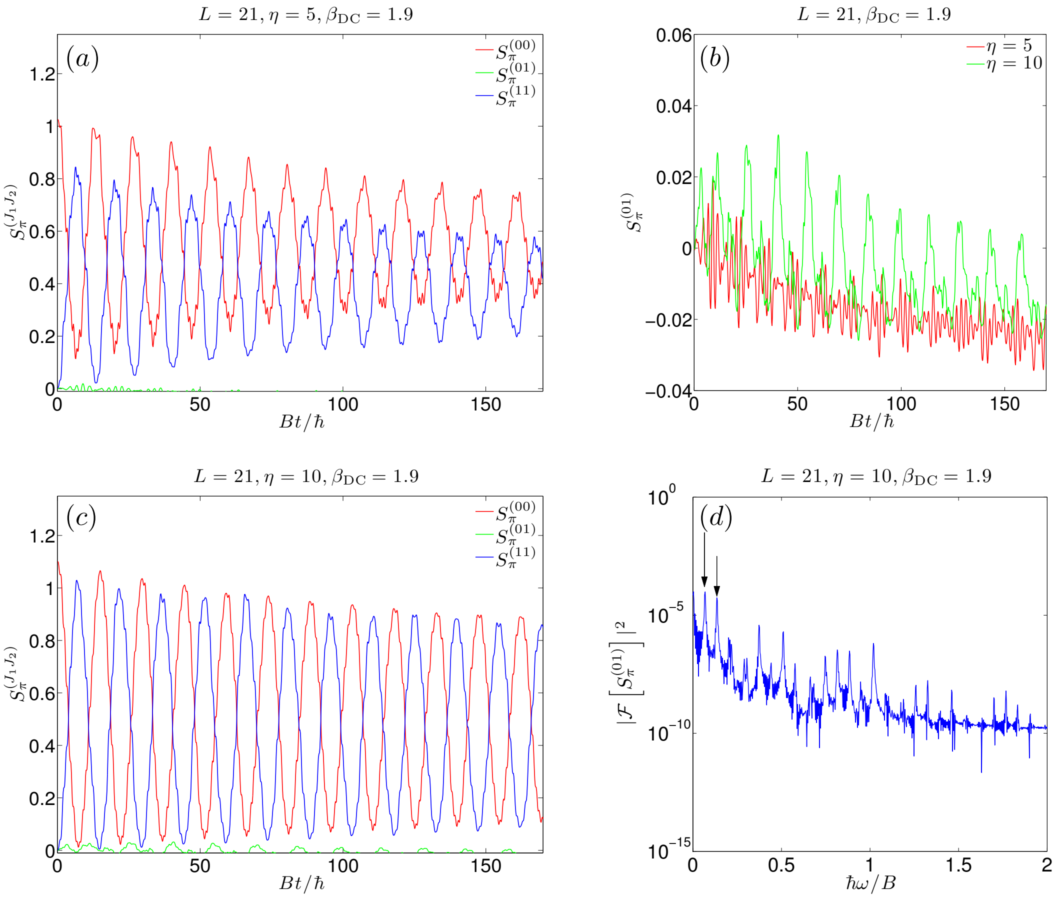


Figure 3.5: Dependence of the asymptotic behavior of structure factors on the lattice height η. (a) Structure factors vs. rotational time for 21 sites with $\eta = 5$. (b) Correlation structure factor $S_\pi^{(01)}$ vs. rotational time for 21 sites with $\eta = 5, 10$. (c) Structure factors vs. rotational time for 21 sites with $\eta = 10$. Note the similarity of $S_\pi^{(00)}$ and $S_\pi^{(11)}$ to the $\eta = 1$ case (Figure 3.3(e)). Note also that $S_\pi^{(01)}$ is now nonzero, and is periodic with the Rabi frequency Ω_{00} at short times and twice the Rabi frequency at long times (see also Figure 3.5(d) and Figure 3.5(b)). (d) Squared modulus of Fourier transform of $S_\pi^{(10)}$ vs. rotationally scaled frequency for $L = 21$ sites and $\eta = 10$. Many new frequencies appear, in particular the Rabi frequency and double the Rabi frequency, denoted with arrows.

Table 3.3: Emergent timescales τ and τ_Q and their fit asymptotic standard errors for various lattice heights and system sizes.

L	η	$\tau B/\hbar$	Asymp. S.E. (%)	$\tau_Q B/\hbar$	Asymp. S.E. (%)
9	1	414.04	0.72	398.4	0.51
9	2	224.32	1.79	149.9	1.36
9	3	117.5	1.86	126.7	1.03
9	10	613.00	1.07	1079.66	14.09
10	1	259.96	0.76	240	0.6454
10	4	140.70	1.19	72.04	0.60
10	10	526.21	0.88	396.46	1.018
21	1	756.18	3.13	110.68	0.96
21	5	177.53	1.62	75.18	0.902
21	10	716.21	2.96	244.09	2.82

with some characteristic timescale τ, as seen in Figure 3.2. We note that an exponential fit has a lower reduced chi-squared than a power-law, or algebraic fit. We also tried fit functions where the oscillations do not decay to zero, but rather persist with some asymptotic nonzero amplitude. We find that the fit functions Eqs. (3.58) and (3.59) above fit the data better as quantified by the convergence properties of the algorithms used, as discussed in appendix 3.7.

The timescale τ also describes the decay of physically measurable quantities, for example the structure factors as defined in Eq. (3.48) and illustrated in Figure 3.3. We show the emergent timescale τ for various lattice heights and systems sizes in Table 3.3.

Examining Figure 3.2, one observes that the driven system approaches a dynamical equilibrium that is a mixture of rotational levels. The timescale with which the system relaxes to this equilibrium, τ, cannot be determined from the single-molecule physics, and so we refer to τ as an *emergent timescale*. For the low lattice height $\eta = 1$, the populations of the first two rotational states appear to oscillate around and asymptotically converge to roughly quarter filling, with $J = 1$ being lower due to contributing to population of $J = 2$ via an off-resonant AC coupling (Figure 3.2(a)).

For $\eta = 5$, the asymptotic equilibrium is an uneven mixture of rotational states that favors occupation of the $J = 0$ state (Figure 3.4(a)), and the emergent timescale for reaching this equilibrium is shorter than it was for $\eta = 1$ by roughly a factor of four. As the lattice height is then increased to $\eta = 10$, the populations return to the trend of $\eta = 1$, again converging to quarter filling with a timescale comparable to that of $\eta = 1$ (Figure 3.4(c)). This illustrates the fact that the emergent timescale τ is not, in general, a monotonic function of the parameters of the lattice.

While the dynamics of the site-averaged rotational state populations are superficially similar for $\eta = 1$ and $\eta = 10$, the underlying physics is not identical, as can be seen by comparing Figure 3.2(f), Figure 3.4(b), and Figure 3.4(d). These figures display the squared modulus of the Fourier transform of the site-averaged number in the $J = 0$ state. The only significant frequency observed for $\eta = 1$ is the Rabi frequency $\Omega \sim 0.064B/\hbar$. In contrast, the $\eta = 5$ case has numerous other characteristic frequencies. As we raise the lattice height to $\eta = 10$, the frequencies that arose for $\eta = 5$ remain, even though the overall visual trend of the site-averaged number reflects that of the single-frequency $\eta = 1$ behavior. While we do not explicitly see the new frequencies in the site-averaged number, we do see them in the structure factors. An example is Figure 3.5(b), which clearly displays the 2Ω frequency behavior of the correlation structure factor $S_{\pi}^{(01)}$ for $\eta = 10$. This frequency, which we easily pick out in the site-averaged number's Fourier transform, can also be seen in the Fourier transform of $S_{\pi}^{(01)}$, see Figure 3.5(d).

We find that the emergent timescale τ does not depend strongly on the size of the system L, even though the distribution of molecules on the lattice is, in general, quite different for different numbers of sites, as can be seen by comparing Figure 3.2(a) and Figure 3.2(e). Examining Figure 3.2(c) and Table 3.3, the $L = 10$ case has a smaller τ than either of the odd L cases. We think this has to do with the filling being exactly 1/2 and not, strictly speaking, with the number of lattice sites, as the

$L = 9$ and $L = 21$ cases have fillings less than $1/2$. We see this clearly by comparing Figure 3.6(a) with Figure 3.2(a), Figure 3.2(c), and Figure 3.2(e). Figure 3.6(a) displays $\langle \hat{n}_{00} \rangle / N$, a quantity which is independent of filling but dependent, in general, on the number of lattice sites. There is a weak dependence on the number of lattice sites. On the other hand, Figure 3.2(a), Figure 3.2(c), and Figure 3.2(e) display $\langle \hat{n}_{00} \rangle / L$, a quantity which is independent of the number of lattice sites but dependent, in general, on the filling. There is a marked difference between $L = 10$, which has filling of $5/10 = 1/2$ and the others, which have fillings$< 1/2$, but there is not a significant difference between $L = 9$ and $L = 21$, which have fillings of $4/9$ and $10/21$, respectively.

The dependence of τ on the filling is also evidenced by the correlation structure factor $S_{\pi}^{(01)}$ in Figure 3.3(f), which shows that there is a stronger correlation between the $J = 0$ and $J = 1$ states for exactly half filling than for fillings less than half, regardless of the system size. Half filling is known to be important in the extended Bose Hubbard model, where it marks the introduction of the charge density wave phase. We thus interpret this greater correlation structure factor as the appearance of a dynamic charge density wave phase *between* rotational states at half filling.

This is in contrast to the usual behavior, where the structure factors $S_{\pi}^{(00)}$ and $S_{\pi}^{(11)}$ are nonzero whenever there is nonzero occupation of the particular rotational state and the structure factor $S_{\pi}^{(01)}$ is much smaller–essentially zero, see Figure 3.3(a) and Figure 3.3(e). These results for the structure factors means that the $J = 0$ and $J = 1$ states tend to lie on top of one another, and not to "checkerboard" with a different rotational state occupying alternating sites. This is due to the fact that the Rabi flopping timescale is much shorter than the dipole-dipole timescale, meaning that the population cycles before there is sufficient time for the molecules to rearrange to a configuration which is energetically favorable with respect to the dipole-dipole term. However, because the population in each rotational level asymptotically reaches some nonzero value, we do see a small amount of rearrangement after many Rabi periods for any filling, corresponding to a nonzero $S_{\pi}^{(01)}$. Note that this rearrangement does not affect the site-averaged numbers, but rather the distribution of rotational states among the lattice sites. This asymptotic distribution emerges on timescales longer than we have considered, and is more prone to finite size effects than the site-averaged quantities, so we do not make a conjecture about it here.

We find that the Q-measure saturates as

$$Q = Q_{\max} - \Delta Q e^{-t_r/\tau_Q}, \tag{3.60}$$

with a different timescale τ_Q, see Figure 3.7(a) and Table 3.3. We also find that the saturation timescale of the Q-measure is not, in general, a monotonic function of the lattice height η, as shown in Figure 3.7(a).

This timescale is different from the timescale τ at which the populations approach an asymptotic equilibrium, though both timescales respond similarly to changes in the Hamiltonian parameter, see Table 3.3. For example, if τ_Q gets larger as a parameter is changed then τ also gets larger, as illustrated in Figure 3.7(a) and Figure 3.7(b). The timescale τ_Q displays a stronger dependence on the number of lattice sites L than τ, as can be seen in Figure 3.6(b) and Figure 3.6(a). This is because τ describes a quantity that has been averaged over sites, while τ_Q does not.

3.5 Conclusions

We have presented and derived a novel lattice Hamiltonian, the MHH. The MHH is a natural Hamiltonian for connecting theoretical studies of the dynamics of quantum phase transitions to near-term experimental setups using ultracold molecular gases. We presented a case study of this new Hamiltonian for hard core bosonic molecules at half filling. Starting from an initial condition of half filling in the $J = 0$, $M = 0$ state, we found that initial large oscillations in the system self-damp to an asymptotic

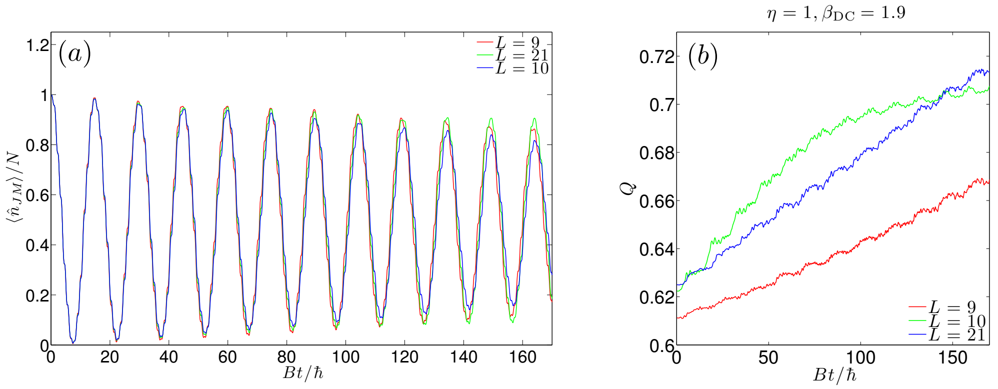


Figure 3.6: Dependence of emergent time scales on number of lattice sites. (a)Dependence of the population damping timescale τ on the number of lattice sites. When we remove the dependence on the filling by dividing through by the total number, we see that there is little difference in the timescales with which systems of different size approach dynamic equilibrium. Contrast Figure 3.2(a), Figure 3.2(c), and Figure 3.2(e), which display a profound dependence on filling when the dependence on lattice sites has been removed. (b) Dependence of spatial entanglement on number of lattice sites. We see that systems of different size have different spatial entanglement in their static ground state. The timescale of the Q-measure saturation, τ_Q, is shorter for $L = 10$ than it is for the odd L cases. This follows the general trend of τ and τ_Q responding correspondingly to changes in the Hamiltonian parameters, and so we associate this shorter timescale partially with the filling, not entirely with the system size.

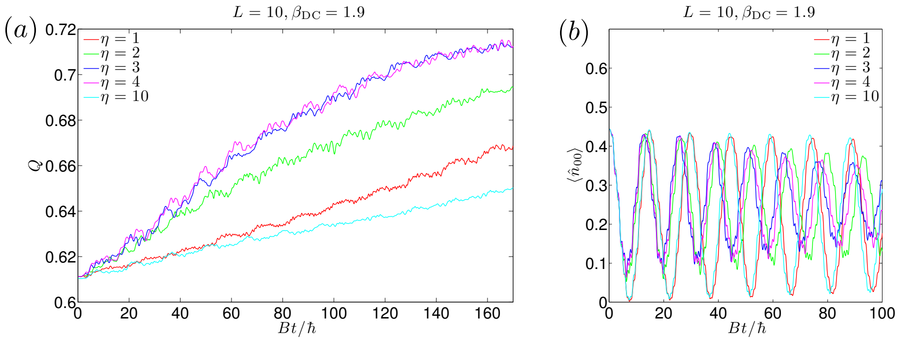


Figure 3.7: Dependence of emergent timescales on lattice height. (a) Dependence of spatial entanglement on lattice height. Note that the spatial entanglement and its associated timescale are *not* monotonic functions of the lattice height. Note also that the entanglement of the static ground state appears to be largely insensitive to the lattice height. (b) Dependence of the site-averaged number on the lattice height. Note that the emergent timescale τ is *not* a monotonic function of the lattice height. Note also that τ responds in the same way that τ_Q does to changes in the lattice height.

equilibrium which consists of a lattice height and filling-dependent spatially entangled superposition of dressed states. This occurs on an emergent timescale τ which can not be predicted from the single molecule theory. We showed that τ depends non-monotonically on lattice height, weakly on lattice size, and strongly on filling (as apparent in simulations with odd and even numbers of sites). We also discovered a separate emergent timescale τ_Q which describes how quickly the many body spatial entanglement saturates. We demonstrated that τ_Q and τ respond similarly to changes in the Hamiltonian parameters and that τ_Q depends on the filling, the lattice size, and, non-monotonically, on the lattice height. In addition to these emergent timescales, we studied the time-dependent structure factors and their frequency-domain Fourier transforms.

In future studies we will consider different filling factors, DC field strength to rotation ratios β_{DC}, and initial conditions, as well as polarized and unpolarized spin-1/2 fermionic molecules. In addition, we will use multiscale methods to study how the emergent timescale demonstrated above compares to experimental timescales for physical systems, and thereby make quantitative predictions for experiments.

We acknowledge useful discussions with Deborah Jin, Heather Lewandowski, and Jun Ye. This work was supported by the National Science Foundation under Grant PHY-0547845 as part of the NSF CAREER program.

3.6 Single Molecule Physics

3.6.1 Relationship Between Operators in Space-Fixed and Molecule-Fixed Coordinate Systems

It is well known that the representation of the angular momentum operators in a molecule-fixed coordinate frame lead to the *anomalous* commutation relations $[J_i, J_j] = -i\hbar\epsilon_{ijk}J_k$ [44]. The simplest way to avoid this trouble is to transform all expressions into the space-fixed frame where the angular momentum operators satisfy the normal commutation relations $[J_i, J_k] = i\hbar\epsilon_{ijk}J_k$ [45]. If the molecule-fixed axes are obtained by rotation of the space-fixed axes through the Euler angles $\{\phi, \theta, \chi\}$ [26] (which we collectively abbreviate as $(\mathbf{R})$), then the component of a k^{th}-rank spherical tensor T that has projection p along the space-fixed z axis, denoted $(T)_p^{(k)}$, can be expressed in terms of the molecule fixed components as

$$(T)_p^{(k)} = \sum_q \mathcal{D}_{pq}^{(k)}(\mathbf{R})^\star (T)_q^{(k)} \,, \tag{3.61}$$

where $\mathcal{D}_{pq}^{(k)}(\mathbf{R})^\star$ is the complex conjugate of the pq element of the k^{th}-rank rotation matrix (Wigner D-matrix). To avoid confusion, we will label all space-fixed components with the letter p and all molecule-fixed components with q. From the orthogonality of the rotation matrices we have the inverse relationship

$$\begin{aligned}(T)_q^{(k)} &= \sum_p \mathcal{D}_{pq}^{(k)}(\mathbf{R})\,(T)_p^{(k)} && (3.62)\\ &= \sum_p (-1)^{p-q}\,\mathcal{D}_{-p,-q}^{(k)}(\mathbf{R})^\star (T)_p^k \,. && (3.63)\end{aligned}$$

3.6.2 Rotational Hamiltonian

In the rigid rotor approximation the rotational Hamiltonian is simply

$$\hat{H}_{\mathrm{rot}} = B\hat{\mathbf{J}}^2 \,, \tag{3.64}$$

where we have defined the rotational constant $B \equiv 1/2\mu r_e^2$, with μ the molecule's reduced mass and r_e its equilibrium internuclear separation. Typical values of B are $\sim 60\hbar$ GHz [46]. This Hamiltonian has eigenvalues $BJ(J+1)$ and eigenstates $|JM\rangle$, with J the total angular momentum and M its projection along the internuclear axis.

3.6.3 DC Field Term

The dipole moment of a polar molecule in a rotational eigenstate is zero in an average sense due to the spherical symmetry of the rotational Hamiltonian. We break this symmetry by introducing a DC electric field along the space-fixed z axis, with Hamiltonian

$$\hat{H}_{\mathrm{DC}} \quad = -\hat{\mathbf{d}} \cdot \mathcal{E}_{\mathrm{DC}}\,, \tag{3.65}$$

where $\mathcal{E}_{\mathrm{DC}}$ is the electric field amplitude. The field defines the spherical space-fixed axis $p = 0$, and the molecule-fixed internuclear axis defines $q = 0$. We transform between them using a first-rank rotation matrix as outlined above:

$$\hat{H}_{\mathrm{DC}} \quad = -\left(\hat{\mathbf{d}}\right)_0^{(1)} \mathcal{E}_{\mathrm{DC}}. \tag{3.66}$$

The matrix elements of the DC Hamiltonian in the basis which diagonalize the rotational Hamiltonian Eq. (3.65) are

$$\begin{aligned}\langle J', M'|\hat{H}_{\mathrm{DC}}|J, M\rangle \quad &= -d\mathcal{E}_{\mathrm{DC}}\sqrt{(2J+1)\,(2J'+1)}\,(-1)^M \\ &\times \begin{pmatrix} J & 1 & J' \\ -M & 0 & M' \end{pmatrix}\begin{pmatrix} J & 1 & J' \\ 0 & 0 & 0 \end{pmatrix}\end{aligned} \tag{3.67}$$

where we use the notation $(\dots)$ for the Wigner 3-j symbol [26]. Note that the symbol d refers to the permanent dipole moment of a molecule, and is not to be confused with the dipole operator denoted by $\hat{\mathbf{d}}$. We refer to the basis which simultaneously diagonalizes the Rotational and DC Hamiltonians as the "dressed basis," and we denote the kets that span this basis by $|\mathcal{E}; JM\rangle$, where the labels J and M are the zero field values of the corresponding quantum number and the symbol $\mathcal{E}$ is a reminder that these kets are superpositions of field free rotational states and DC field.

The effects of the DC field can be clearly seen by considering the dressed state wavefunctions, energies, and dipole moments to lowest order in perturbation theory in the dimensionless parameter $\beta_{\mathrm{DC}} \equiv d\mathcal{E}_{\mathrm{DC}}/B$, the ratio of the field energy to the rotational level splitting:

$$|\mathcal{E}; J, M\rangle = |J, M\rangle - \frac{\beta_{\mathrm{DC}}}{2J}\sqrt{\frac{J^2 - M^2}{4J^2 - 1}}|J-1, M\rangle + \frac{\beta_{\mathrm{DC}}}{2\,(J+1)}\sqrt{\frac{(J+1)^2 - M^2}{4\,(J+1)^2 - 1}}|J+1, M\rangle\,, \tag{3.68}$$

$$\Delta E_{JM}^{(2)} = \frac{d^2\mathcal{E}_{\mathrm{DC}}^2}{2B}\left[\frac{J\left(J+1\right)-3M^2}{J\left(J+1\right)\left(2J-1\right)\left(2J+3\right)}\right], \tag{3.69}$$

$$\langle\mathcal{E};JM|\hat{\mathbf{d}}|\mathcal{E};JM\rangle/d = -\frac{\partial E_{JM}}{\partial\beta_{\mathrm{DC}}} = \beta_{\mathrm{DC}}\frac{3M^2/J\left(J+1\right)-1}{\left(2J-1\right)\left(2J+3\right)}, \tag{3.70}$$

where $\Delta E_{JM}^{(2)}$ is the lowest non-zero shift in the energy.

The DC field mixes states of different J, breaking the $(2J+1)$-fold degeneracy of the rotational Hamiltonian, and so J is no longer a good quantum number. In the case of a z-polarized field, M remains a good quantum number, and a degeneracy persists for all states with the same $|M|$. This mixing aligns the molecule with the field, inducing a nonzero dipole moment. This means of orienting polar molecules, known as “brute force” orientation, works well for molecules that both have a large dipole moment and can be efficiently rotationally cooled [47]. While more effective means of orienting molecules using intense laser fields are known [48], they complicate the theoretical discussion and the experimental setup, and so we do not consider them here.

In larger fields the rotational levels become deeply mixed, which allows states that are weak-field seeking in low fields to become high-field seeking in high fields [49]. The actual mixing of rotational levels vs. β_{DC} is depicted in Figure 3.9 for the lowest three dressed levels. We note that there always exists a field $\mathcal{E}_R$ such that the lowest R dressed states' dipole moments are all positive, as this is important to ensure the stability of a collection of dipoles. The universal curve of the induced dipole moments (in units of d) vs. β_{DC} of the first two dressed rotational manifolds are shown in Figure Figure 3.8(a). The universal curve of the dressed state energies energies (in units of B) vs. β_{DC} is shown in Figure Figure 3.8(b). For reference, $\beta_{\mathrm{DC}}=1$ corresponds to a field of roughly $1.93\frac{\mathrm{kV}}{\mathrm{cm}}$ for $B\sim 60\hbar$ GHz and $d\sim 9$ D.

Expanding the field operators in Eq. (3.1) in a Wannier basis of dressed states centered at a particular discrete position $\mathbf{r}_i$ as described in Eq. (3.4), we find

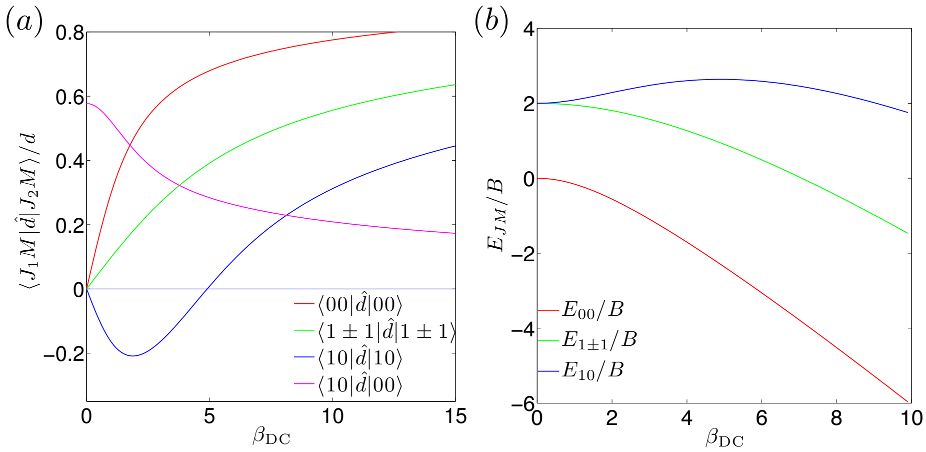


Figure 3.8: Dressed state dipole moments and energies. Note that the $J = 1, M = 0$ resonant dipole moment changes from weak-field seeking to high-field seeking at $\beta_{\mathrm{DC}} \approx 5$. All rotational states have a field where this transition occurs, and the dipole tends monotonically towards unity after this field. The $\langle 10|\hat{\mathbf{d}}|00\rangle$ dipole moment (and all transition dipole moments, generically) tends towards zero monotonically as β_{DC} increases. Note also that the energetic differences between rotational levels are smallest at zero field and grow monotonically thereafter. (a) Scaled induced dipole moments vs. scaled DC field energy. (b) Scaled dressed energies vs. scaled DC field energy.

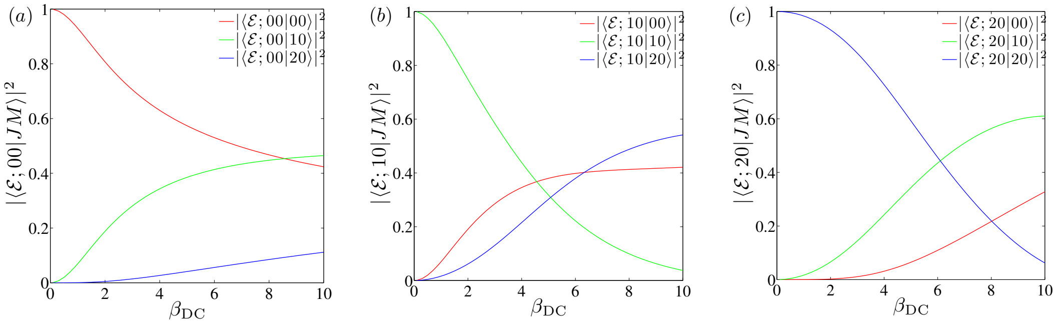


Figure 3.9: Compositions of dressed states vs. scaled rotational energy. The states become deeply mixed in large fields, and that the dressed state $|\mathcal{E}; JM\rangle$ whose zero field value is $|JM\rangle$ does not always have the greatest overlap with $|JM\rangle$ for all β_{DC}. The field strength where the first dressed state changes from weak-field to high-field seeking, $\beta_{\mathrm{DC}} = 5$, is also roughly the place where its overlap with the $|00\rangle$ field-free level is greater than the overlaps with all other field-free levels. (a) Composition of 1^{st} dressed state. (b) Composition of 2^{nd} dressed state. (c) Composition of 3^{rd} dressed state.

$$\hat{H}_{\text{rot}} + \hat{H}_{\text{DC}} \quad = \sum_J \sum_{M=-J}^{J} E_{J,M} \hat{n}_{\mathcal{E},JM} \,, \tag{3.71}$$

where E_{JM} is the energy of the $|\mathcal{E}; J, M\rangle$ dressed state (see Figure 3.9(a)) and $\hat{n}_{\mathcal{E},JM}$ is the number operator associated with this same state.

If the DC field were aligned at a small angle θ_a to the z field of the trap (say, in the xz plane), then small dipole moments mixing $M' = M \pm 1$ states would arise and the $M' = M$ dipoles would decrease slightly (we can view them as being in an effective field of $\mathcal{E}_{\text{eff}} = \cos\theta_a \mathcal{E}_{\text{DC}}$). Treating the new contribution perturbatively in the small parameter $\sin\theta_a \beta_{\text{DC}}$, we find the lowest order couplings to the ground state

$$\langle \mathcal{E}; 00 | \hat{H}_{\text{DC}} | \mathcal{E}; 1 \pm 1 \rangle \simeq \frac{\sin\theta_a d\mathcal{E}}{\sqrt{6}} \left(1 - \frac{49 \sin^2\theta_a}{1440} \beta_{\text{DC}}^2 \right) \,, \tag{3.72}$$

and associated timescale τ_{θ_a} for occupation of $M \neq 0$ states from the ground state,

$$\tau_{\theta_a} = \frac{\sqrt{6}\hbar}{\sin\theta_a d\mathcal{E} \left(1 - \frac{49 \sin^2\theta_a}{1440} \beta_{\text{DC}}^2 \right)} \sim \frac{\sqrt{6}}{\beta_{\text{DC}} \sin\theta_a} \frac{\hbar}{B} \,. \tag{3.73}$$

3.6.4 AC Field Term

An AC microwave field of frequency ω resonantly drives transitions between two DC dressed states $|\mathcal{E}; J'M'\rangle$ and $|\mathcal{E}; JM\rangle$ with energy difference $\left(E_{J'M'} - E_{JM}\right)/\hbar \approx \omega$ provided the induced dipole moment $\langle \mathcal{E}; J'M' | \hat{\mathbf{d}} | \mathcal{E}; JM \rangle$ is nonzero. Two states separated by an energy difference ΔE that is off-resonant from the driving field (i.e. $\Delta E \gg \omega$) will also be coupled, albeit much more weakly. In our system we resonantly couple the lowest two dressed rotational levels, $|\mathcal{E}; 10\rangle$ and $|\mathcal{E}; 00\rangle$. We consider the case of z polarization, in which the effective Hamiltonian in the dressed Wannier basis is

$$\hat{H}_{AC}\left(t\right) \quad = -\pi \sin\left(\omega t\right) \sum_{JM} \Omega_{JM} \left(\hat{a}^{\dagger}_{\mathcal{E};J,M} \hat{a}_{\mathcal{E};J+1,M} + \text{h.c} \right) \,, \tag{3.74}$$

where

$$\Omega_{JM} \equiv \mathcal{E}_{\mathrm{AC}}\langle\mathcal{E};J,M|\hat{\mathbf{d}}|\mathcal{E};J+1,M\rangle/\hbar\,. \tag{3.75}$$

is the Rabi frequency. This is the frequency with which the populations of a two-level system cycles. In experiments, the AC field has spatial curvature on the order of cm which is negligible on the μm system size scale.

In the absence of couplings between sites, the physics of the system is determined by the on-site, single-molecule physics. The percentage population of each component in both the $|\mathcal{E};J,M\rangle$ dressed and $|JM\rangle$ field-free bases are shown below for one Rabi period. In these plots only the $|\mathcal{E};10\rangle$ and $|\mathcal{E};00\rangle$ dressed states are considered, which is close to the actual behavior when all other states are far off-resonant. Each site undergoes Rabi flopping independently of the others. Figure 3.10(a) and Figure 3.10(b) show this behavior for $\beta_{\mathrm{DC}} = 1.900$ and $\beta_{\mathrm{AC}} \equiv d\mathcal{E}_{\mathrm{AC}}/B = 0.200$, giving a Rabi period of $2\pi/\Omega_{00} = 36.5\hbar/B$.

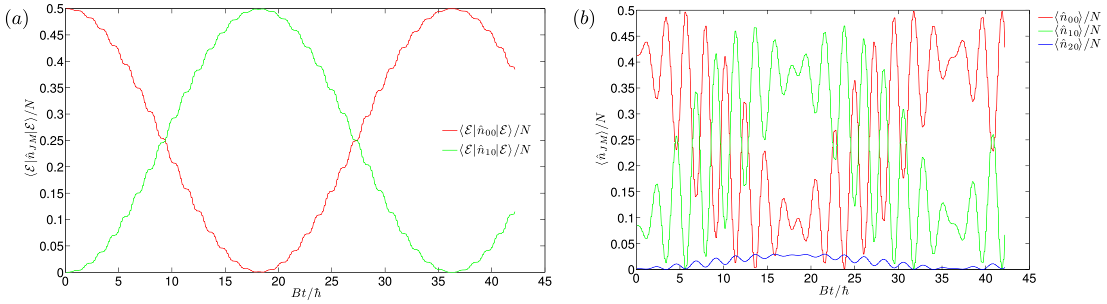


Figure 3.10: Resonant AC field induced population cycling in the dressed and field-free bases. (a) Populations of the dressed states vs. rotational time. The small amplitude rapid oscillations occur on the timescale $1/\omega$, and are often averaged away via the rotating wave approximation. The large amplitude oscillations occuring on the timescale $1/\Omega_{00}$ that periodically transfer the population between $|\mathcal{E};00\rangle$ and $|\mathcal{E};10\rangle$ are the characteristic "Rabi oscillations" of a driven two-level system. (b) Populations of the field-free states vs. rotational time. The $|20\rangle$ state is occupied because both $|\mathcal{E};00\rangle$ and $|\mathcal{E};10\rangle$ have a nonzero projection with this state due to the mixing from the DC field, see Figure 3.9. It is apparent from comparison with Figure 3.10(a) that the dressed basis greatly simplifies the AC term in the Hamiltonian.

3.7 Convergence

3.7.1 Single Molecule Considerations

Each dressed state $|\mathcal{E}; J, M\rangle$ is, in principle, an infinite linear combination of field free states

$$|\mathcal{E}; J, M\rangle \quad = \sum_{J'=0}^{\infty} c_{J'} |J', M\rangle. \tag{3.76}$$

Numerically, we must have a finite upper bound to the sum in Eq. (3.76), which we call J_{cut}. This does not cause difficulty in practice, as the overlap of a dressed state $|\mathcal{E}; JM\rangle$ with a field-free state $|J'M\rangle$ diminishes rapidly as J' differs more greatly from J. We find the coefficients in Eq. (3.76), as well as the dressed state energies and dipole moments by simultaneously diagonalizing the rotational and DC field Hamiltonians in a basis consisting of the first J_{cut} rotational levels. Because TEBD scales poorly with the on-site dimension, we form as small an on-site basis as possible by keeping the eigenvectors corresponding to the R lowest dressed levels. To form a proper basis, we must renormalize these eigenvectors (which, for z-polarized field, does not change their orthogonality). We now demonstrate the convergence of these two procedures

To show convergence of the first procedure, we plot the difference between the energy of the J^{th} rotational state calculated for a particular value of $J_{\text{cut}} = i$ and one higher value, $\Delta E_J(i)$ as a function of i. The results for various field strengths are shown in Figure 3.11(a)-Figure 3.11(b). We see very fast convergence for the low fields (e.g. $\beta_{\text{DC}} = 1.9$) of interest. In our numerics we use $J_{\text{cut}} = 25$, which ensures convergence for any of the β_{DC} considered.

To determine convergence with respect to the second procedure, examine Figure 3.9(a)-Figure 3.9(b), which show

$$P_J^{(J_{\max})} \quad \equiv 1 - \sum_{i=0}^{J_{\max}-1} |\langle \mathcal{E}; J0|i0\rangle|^2 \ , \tag{3.77}$$

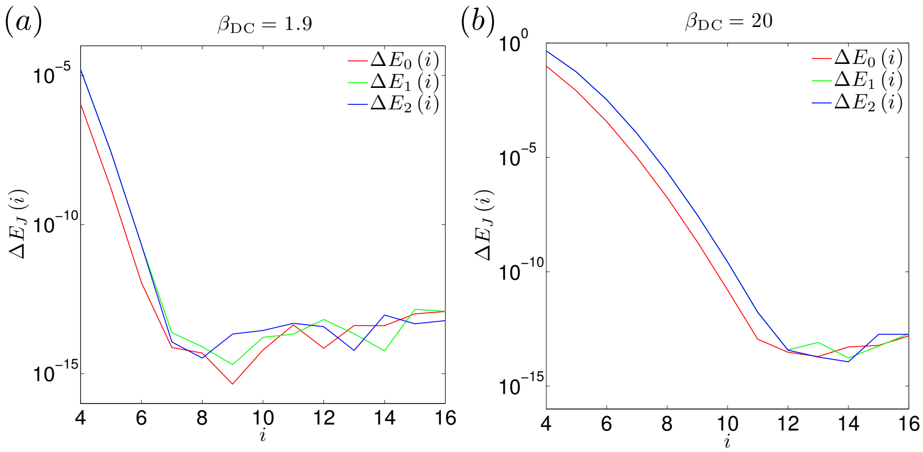


Figure 3.11: Convergence with respect to DC dressing rotational state cutoff. As few as 7 field-free levels are needed for the weak field $\beta_{\mathrm{DC}} = 1.9$ to have the dressed state energies of interest converge to machine precision (left panel), and even a large DC field $\beta_{\mathrm{DC}} = 20$ requires only 12 field-free levels for the energy to converge (right panel).

the amount of the total dressed wave function norm $|\langle \mathcal{E}; J0|\mathcal{E}; J0\rangle|^2$ that lies outside of the first $J_{\max}$ field-free rotational levels for $J_{\max} = 3$ and 4, respectively. For $J_{\max} = 4$ the renormalization of the first three rotational levels is a very small effect for the β_{DC} we consider, and the fourth level is not populated to any appreciable extent during time evolution for any β_{DC} (see Fig.Figure 3.10(a)), so we expect that keeping the $J_{\max} = 4$ lowest levels will give sufficient accuracy. By direct simulation, we find six digit accuracy in the suite of quantum measures defined in Sec. 3.3.2; specifically, we compare $J_{\max} = 3$ and 4.

3.7.2 Many Body Considerations

There are also convergence issues that are inherent to the TEBD algorithm. The first, called the *Schmidt error*, is the error that arises from truncating the Hilbert space at each time step. We can parameterize the error per step in terms of the entanglement cutoff parameter χ as

$$\tau_l^S \;= 1 - \sum_{\alpha_l=1}^{\chi} \left(\lambda_{\alpha_l}^{[l]}\right)^2 \tag{3.78}$$

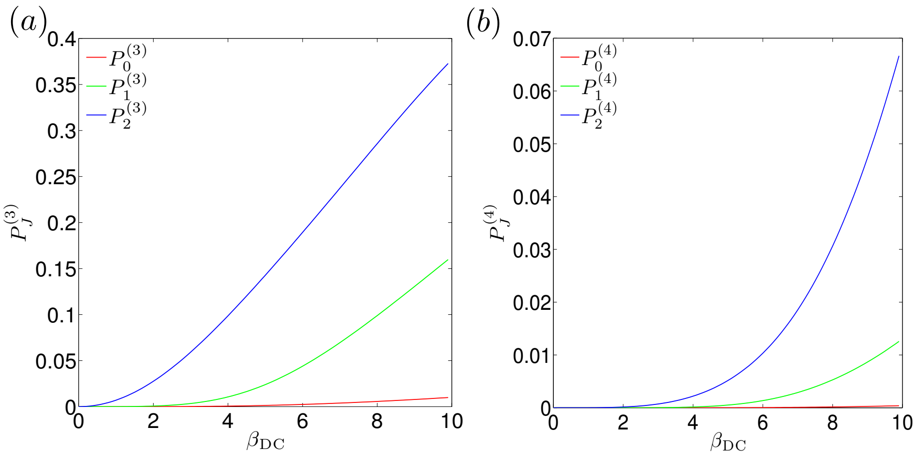


Figure 3.12: Convergence with respect to local dimension cutoff. Dressed states with greater J lose more of their norm in truncation, as mixing occurs most strongly with adjacent J. Also, as the field is increased, the states become more deeply mixed, and so all states lose more of their norm. Truncating the local basis at the $J = 3$ dressed level incurs at most a 1% loss of norm for any of the states that are appreciably populated during time evolution (right panel).

where $\lambda^{[l]}$ is a vector containing the eigenvalues of the reduced density matrix obtained by tracing over all sites but l, and α_l is the local index that entangles the site l with the rest of the system, with smaller α_l states having greater weight. We find that, among the measures we use, the one that is the most sensitive to χ is the Q-measure, which we plot for four values of χ in Figure 3.13. Increasing χ improves the accuracy over longer times, but there is always a time after which the measure begins to deviate. This is the normalization drift alluded to in Sec. 3.3.1. The χ-dependent time after which the Schmidt error dominates is referred to as the runaway time [40]. In the case study of Sec. 3.4, we used $\chi = 50$ for all simulations, which gives the Q-measure accurately to within four decimal places over the timescales considered.

The second intrinsic source of error in TEBD is due to the Trotter-Suzuki expansion of the propagator [33]. We parameterize this error in terms of δt, the time step. When we halve the time step from that used in the simulations above ($= 2\pi/(133\omega)$), we find no change in the measures to the ninth digit. It is clear that the Schmidt error discussed above is the chief source of error in our simulations.

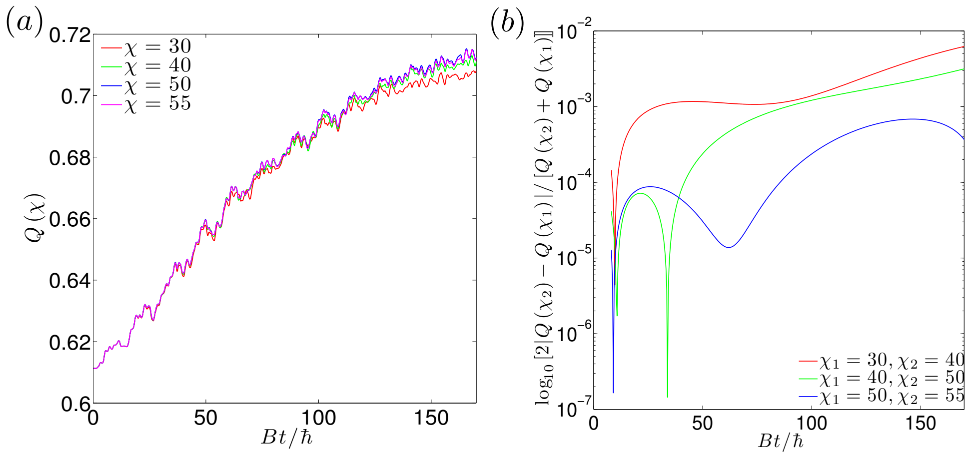


Figure 3.13: Convergence with respect to entanglement cutoff parameter. The left figure shows the spatial entanglement measure Q for various values of the TEBD entanglement cutoff parameter χ. As χ is increased, Q remains close to its true value for longer. In the right figure we plot the log of the absolute difference in Q for two values of χ divided by its arithmetic mean. We see at least four-digit accuracy for the largest values of χ we consider. Note also that even small values of χ are accurate for short times.

To extract the emergent timescales defined in Eqs. (3.58) and (3.60), we used two different methods. The first is the nonlinear curve fitting routine "fit" in gnuplot. The second is the "NonlinearRegression" package in Mathematica 6.0. Both methods use nonlinear regression, which fits the data to a specified nonlinear function of the model parameters. The goodness of the fit is quantified by the asymptotic standard errors of the model parameters, which gives the standard deviation of each parameter. A low percent asymptotic error means that the model parameters cannot be adjusted very far without noticeably changing the goodness-of-fit. Both gnuplot and Mathematica returned the same values for the emergent timescales to within the stated asymptotic standard error.

3.8 References Cited

[1] R. P. Feynman. Simulating Physics with Computers. *Int. J. Theor. Phys.*, 21: 467, 1982.

[2] M. Sanpera A. Lewenstein, V. Ahufinger, B. Damski, A. Sen De, and U. Sen. Ultracold atomic gases in optical lattices: Mimicking condensed matter physics and beyond. *Adv. Phys.*, 56:243–379, 2007.

[3] K.-K. Ni, S. Ospelkaus, M. H. G. de Miranda, A. Peér, B. Neyenhuis, J. J. Zirbel, S. Kotochigova, P. S. Julienne, D. S. Jin, and Jun Ye. A High Phase-Space-Density Gas of Polar Molecules. *Science*, 322:231–235, 2008.

[4] F. Lang, K. Winkler, C. Strauss, R. Grimm, and J. Hecker Denschlag. Ultracold Triplet Molecules in the Rovibrational Ground State. *Phys. Rev. Lett.*, 101: 133005, Sep 2008. doi: 10.1103/PhysRevLett.101.133005. URL `http://link.aps.org/doi/10.1103/PhysRevLett.101.133005`.

[5] Gavin K Brennen, Andrea Micheli, and Peter Zoller. Designing spin-1 lattice models using polar molecules. *New J. Phys.*, 9:138, 2007.

[6] H. P. Büchler, E. Demler, M. Lukin, A. Micheli, N. Prokof'ev, G. Pupillo, and P. Zoller. Strongly Correlated 2D Quantum Phases with Cold Polar Molecules: Controlling the Shape of the Interaction Potential. *Phys. Rev. Lett.*, 98:060404, Feb 2007. doi: 10.1103/PhysRevLett.98.060404. URL `http://link.aps.org/doi/10.1103/PhysRevLett.98.060404`.

[7] D. DeMille. Quantum Computation with Trapped Polar Molecules. *Phys. Rev. Lett.*, 88:067901, 2002.

[8] K. Goral, L. Santos, and M. Lewenstein. Quantum Phases of Dipolar Bosons in Optical Lattices. *Phys. Rev. Lett.*, 88:170406–1–4, 2002.

[9] G. Pupillo, A. Micheli, H. P. Buchler, and P. Zoller. *Cold molecules: Creation and applications*, chapter 12. Taylor & Francis, 2009.

[10] Till D. Kühner, Steven R. White, and H. Monien. One-dimensional Bose-Hubbard model with nearest-neighbor interaction. *Phys. Rev. B*, 61(18):12474–12489, May 2000. doi: 10.1103/PhysRevB.61.12474.

[11] D. B. M. Dickerscheid, Y. Kawaguchi, and M. Ueda. Heteronuclear fermionic superfluids with spin degrees of freedom. *Phys. Rev. A*, 77:053605, 2008.

[12] J. M. Higbie, L. E. Sadler, S. Inouye, A. P. Chikkatur, S. R. Leslie, K. L. Moore, V. Savalli, and D. M. Stamper-Kurn. Direct Nondestructive Imaging of Magnetization in a Spin-1 Bose-Einstein Gas. *Phys. Rev. Lett.*, 95:050401, 2005.

[13] Guifré Vidal. Efficient Classical Simulation of Slightly Entangled Quantum Computations. *Phys. Rev. Lett.*, 91(14):147902, Oct 2003. doi: 10.1103/PhysRevLett.91.147902.

[14] Guifré Vidal. Efficient Simulation of One-Dimensional Quantum Many-Body Systems. *Phys. Rev. Lett.*, 93(4):040502, Jul 2004. doi: 10.1103/PhysRevLett.93.040502.

[15] M. Nielsen and I. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, 2000.

[16] T. Wenger T. Kinoshita and D. S. Weiss. A quantum Newton's cradle. *Nature*, 440:900–903, 2006.

[17] A. Polkovnikov, S. Sachdev, and S. M. Girvin. Nonequilibrium Gross-Pitaevskii dynamics of boson lattice models. *Phys. Rev. A*, 66:053607–1–7, 2002.

[18] E. Altman, A. Polkovnikov, E. Demler, B. Halperin, and M. D. Lukin. Superfluid-insulator transition in a moving system of interacting bosons. *Phys. Rev. Lett.*, 95:020402–1–4, 2005.

[19] S. R. Manmana, S. Wessel, R. M. Noack, and A. Muramatsu. Strongly Correlated Fermions after a Quantum Quench. *Phys. Rev. Lett.*, 98:210405, 2007.

[20] Corinna Kollath, Andreas M. Läuchli, and Ehud Altman. Quench Dynamics and Nonequilibrium Phase Diagram of the Bose-Hubbard Model. *Phys. Rev. Lett.*, 98:180601, 2007.

[21] Andreas M Läuchli and Corinna Kollath. Spreading of correlations and entanglement after a quench in the one-dimensional Bose-Hubbard model. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(05):P05018, 2008. URL `http://stacks.iop.org/1742-5468/2008/i=05/a=P05018`.

[22] Claudio Verdozzi. Time-Dependent Density-Functional Theory and Strongly Correlated Systems: Insight from Numerical Studies. *Phys. Rev. Lett.*, 101:166401, 2008.

[23] R. V. Mishmash and L. D. Carr. Ultracold Atoms in 1D Optical Lattices: Mean Field, Quantum Field, Computation, and Soliton Formation. *Journal of Mathematics and Computers in Simulation*, 80:732, 2008.

[24] John Brown and Alan Carrington. *Rotational Spectroscopy of Diatomic Molecules*. Cambridge University Press, Cambridge, 2003.

[25] A. Micheli, G. Pupillo, H. P. Büchler, and P. Zoller. Cold polar molecules in two-dimensional traps: Tailoring interactions with external fields for novel quantum phases. *Phys. Rev. A*, 76:043604, 2007.

[26] We use the conventions of Zare, op. cit.

[27] Ana Maria Rey. *Ultracold Bosonic Atoms in Optical Lattices*. PhD thesis, University of Maryland, 2004.

[28] J. Deiglmayr, M. Aymar, R. Wester, M. Weidemüller, and O. Dulieu. Calculations of static dipole polarizabilities of alkali dimers: Prospects for alignment of ultracold molecules. *J. Chem. Phys.*, 129:064309, 2008.

[29] D. Jaksch, C. Bruder, J. I. Cirac, C. W. Gardiner, and P. Zoller. Cold Bosonic Atoms in Optical Lattices. *Phys. Rev. Lett.*, 81(15):3108–3111, Oct 1998. doi: 10.1103/PhysRevLett.81.3108.

[30] S. Sachdev. *Quantum Phase Transitions*. Cambridge University Press, New York, 1999.

[31] M. P. A. Fisher, P. B. Weichman, G. Grinstein, and D. S. Fisher. Boson localization and the superfluid-insulator transition. *Phys. Rev. B*, 40:546–570, 1989.

[32] Steven R. White. Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.*, pages 2863–2866, 1992.

[33] M. Suzuki. Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations. *Physics Letters A*, 146: 319–323, June 1990. doi: 10.1016/0375-9601(90)90962-N.

[34] Andrew John Daley. *Manipulation and Simulation of Cold Atoms in Optical Lattices*. PhD thesis, PhD thesis, Leopold-Franzens-Universität Innsbruck, 2005.

[35] A. J. Daley, C. Kollath, U. Schollwöck, and G. Vidal. Time-dependent density-matrix renormalization-group using adaptive effective Hilbert spaces. *Journal of Statistical Mechanics: Theory and Experiment*, 2004(04):P04005, 2004. URL http://stacks.iop.org/1742-5468/2004/i=04/a=P04005.

[36] G. K. Brennen. An observable measure of entanglement for pure states of multi-qubit systems. *Quant. Inf. Comp.*, 3:619–626, 2003.

[37] Howard Barnum, Emanuel Knill, Gerardo Ortiz, and Lorenza Viola. Generalizations of entanglement based on coherent states and convex sets. *Phys. Rev. A*, 68:032308, 2003.

[38] Howard Barnum, Emanuel Knill, Gerardo Ortiz, Rolando Somma, and Lorenza Viola. A Subsystem-Independent Generalization of Entanglement. *Phys. Rev. Lett.*, 92:107902, 2004.

[39] Neil Ashcroft and David Mermin. *Solid State Physics*. Saunders College Publishing, Orlando, 1976.

[40] Dominique Gobert, Corinna Kollath, Ulrich Schollwöck, and Gunter Schütz. Real-time dynamics in spin-(1/2) chains with adaptive time-dependent density matrix renormalization group. *Phys. Rev. E*, 71:036102, 2005.

[41] J.K. Kevorkian and J.D. Cole. *Multiple Scale and Singular Perturbation Methods*. Springer, New York, 1^{st} edition edition, 1996.

[42] Ali H. Nayfeh. *Perturbation Methods*, volume 1^{st} edition. Wiley-Interscience, July 12, 2000.

[43] Deborah Jin, 2008. JILA, NIST and Univ. of Colorado, private communication.

[44] J. H. Van Vleck. The coupling of angular momentum vectors in molecules. *Rev. Mod. Phys.*, 23:213–227, 1951.

[45] Richard Zare. *Angular Momentum: Understanding Spatial Aspects in Chemistry and Physics*. Wiley, New York, 1988.

[46] http://physics.nist.gov/physrefdata/molspec/diatomic/, 2008. nist diatomic spectral database.

[47] Henrik Stapelfeldt and Tamar Seideman. *Colloquium* : Aligning molecules with strong laser pulses. *Rev. Mod. Phys.*, 75:543–557, Apr 2003. doi: 10.1103/RevModPhys.75.543. URL `http://link.aps.org/doi/10.1103/RevModPhys.75.543`.

[48] Bretislav Friedrich and Dudley Herschbach. Alignment and Trapping of Molecules in Intense Laser Fields. *Phys. Rev. Lett.*, 74:4623–4626, Jun 1995. doi: 10.1103/PhysRevLett.74.4623. URL `http://link.aps.org/doi/10.1103/PhysRevLett.74.4623`.

[49] Alexander V. Avdeenkov, Masatoshi Kajita, and John L. Bohn. Suppression of inelastic collisions of polar $^1\sigma$ state molecules in an electrostatic field. *Phys. Rev. A*, 73:022707, Feb 2006. doi: 10.1103/PhysRevA.73.022707. URL `http://link.aps.org/doi/10.1103/PhysRevA.73.022707`.

CHAPTER 4

HYPERFINE MOLECULAR HUBBARD HAMILTONIAN

Abstract:[24] An ultracold gas of heteronuclear alkali dimer molecules with hyperfine structure loaded into a one-dimensional optical lattice is investigated. The hyperfine molecular Hubbard Hamiltonian (HMHH), an effective low-energy lattice Hamiltonian, is derived from first principles. The large permanent electric dipole moment of these molecules gives rise to long range dipole-dipole forces in a dc electric field and allows for transitions between rotational states in an ac microwave field. Additionally, a strong magnetic field can be used to control the hyperfine degrees of freedom independently of the rotational degrees of freedom. By tuning the angle between the dc electric and magnetic fields and the strength of the ac field it is possible to control the number of internal states involved in the dynamics as well as the degree of correlation between the spatial and internal degrees of freedom. The HMHH's unique features have direct experimental consequences such as quantum dephasing, tunable complexity, and the dependence of the phase diagram on the molecular state.

4.1 Introduction

Ultracold molecular gases are of interest in many subfields of science ranging from precision science to quantum simulation of many-body Hamiltonians [1]. Recent success using the stimulated Raman adiabatic passage (STIRAP) method has allowed experimentalists to produce a gas of KRb molecules close to Fermi degeneracy, in the ground rovibrational state, and in a specific hyperfine level [2, 3]. Rovibonic ground state molecules have also been formed for polar LiCs[4] as well as nonpolar Cs_2[5] and Rb_2[6], with studies on other species currently underway[7, 8]. To reach the quantum

---

[24] Published previously as *Hyperfine molecular Hubbard Hamiltonian*, M. L. Wall and L. D. Carr, Physical Review A **82**, 013611 (2010).

degenerate regime one must have all molecules in the same quantum state, a task which is complicated by the rich hyperfine structure of alkali dimer molecules. Thus, a number of recent works [9–11] have investigated the single-molecule microwave spectra to find a route by which all molecules are transferred to the lowest hyperfine state, yielding a gas of *absolute* ground state molecules.

From the condensed matter perspective, ultracold gases are enticing in their capacity to act as *quantum simulators* [12, 13]. Such specialized quantum computers allow for the study of complex many-body Hamiltonians in a setting where many parameters are amenable to experimental control. From this point of view, it is natural to ask how the various degrees of freedom in the quantum simulator may be controlled and used as resources. Theoretical proposals for many-body physics using ultracold molecules have so far focused only on the rotational degrees of freedom in $^1\Sigma$ molecules with external fields [14, 15] or on the hyperfine degree of freedom in $^2\Sigma$ molecules without external fields [16]. In this work we study $^1\Sigma$ molecules in strong fields including the effects of hyperfine structure and discuss how the hyperfine degrees of freedom may be controllably accessed and manipulated as a resource for generating complex quantum dynamics.

For $^1\Sigma$ molecules it has been shown that the interaction of the rotational degrees of freedom with external electric fields allows for the tuning of the strength and range of the two-molecule interaction potential [14]. Many of these results also hold for molecules with hyperfine structure, as the rotational and nuclear spin degrees are only weakly coupled in strong fields. In particular, the application of a dc field and an optical trapping potential gives rise to a purely repulsive dipole-dipole interaction between molecules in reduced geometries. Also, it has been shown that the combination of a strong uniform magnetic field and a suitably chosen microwave field allows for transitions between particular hyperfine single-molecule states, and that this may be used to transfer a collection of molecules that have been cooled to the rovibrational ground state but an excited hyperfine state to their hyperfine ground state [9–11]. This idea also works in reverse: one can select the states which are involved in many-body dynamics with the ground state by judicious choice of the field strengths and geometries. The hyperfine molecular Hubbard Hamiltonian (HMHH) reflects this fact; not only the parameters of the Hamiltonian but also the dimensionality and character of the basis are suited to experimental control.

This article is organized as follows. In Sec. 4.2 we introduce the HMHH, define its parameters, and discuss its interesting experimental consequences. This section contains the main results of the paper. In Sec. 4.2.3 we derive the HMHH from first principles and state the key assumptions underlying its derivation. Finally, in Sec. 4.4, we conclude. Some details concerning the single molecule physics are provided in the appendices in the interest of completeness.

4.2 Statement of the Hamiltonian and Experimental Consequences

The hyperfine molecular Hubbard Hamiltonian is

$$\begin{aligned}\hat{H} = &\sum_{\sigma} \Delta_{\sigma} \sum_{i} \hat{n}_{i\sigma} - \sum_{\sigma} t_{\sigma} \sum_{\langle i,j \rangle} \left[\hat{a}^{\dagger}_{i\sigma} \hat{a}_{j\sigma} + \text{h.c.} \right] \\ &+ \frac{1}{2} \sum_{\sigma,\sigma'} U_{\sigma\sigma'} \sum_{\langle i,j \rangle} \hat{n}_{i\sigma} \hat{n}_{j\sigma'} \\ &- \frac{1}{2} \sum_{\sigma\sigma'} d_{\sigma\sigma'} E_{\text{ac}} \sum_{i} \left[\hat{a}^{\dagger}_{i\sigma} \hat{a}_{i\sigma'} + \text{h.c.} \right] , \end{aligned} \tag{4.1}$$

where $\hat{a}_{i\sigma}$ destroys a bosonic or fermionic molecule in state $|\sigma\rangle$ on the i^{th} lattice site, and the bracket notation $\langle \dots \rangle$ denotes that the sum is taken over nearest neighbors. The single-molecule basis $\{|\sigma\rangle\}$ takes into account the hyperfine interactions (Appendix 4.5) and static fields (Appendix 4.6) and the quantum number σ is a composite index referring to both rotational and nuclear spin degrees of freedom. The properties and dimensionality of this basis can be modified by the geometry and strength of the external fields, as will be discussed in more detail below.

Table 4.1: Table of energy scales of the hyperfine molecular Hubbard Hamiltonian. From top to bottom: energy Δ_σ of internal state $|\sigma\rangle$, relative to the ground state; tunneling t_σ; dipole-dipole interaction $U_{\sigma\sigma'}$; transition dipole moment $d_{\sigma\sigma'}$ due to the ac electric drive E_{ac}.

Term	Energy scale
Δ_σ	$\sim 1 - 100$ kHz (depends on static field strengths)
t_σ	~ 1 kHz
$U_{\sigma\sigma'}$	$\sim$250 Hz
$d_{\sigma\sigma'}E_{\mathrm{ac}}$	$\sim 1 - 50$ kHz

The first term in the HMHH represents the energy offset of a molecule in state $|\sigma\rangle$ from a reference ground state. The second term describes the tunneling of molecules between lattice sites and depends on the rotational state. The third term describes resonant dipole-dipole interactions between molecules on neighboring sites. The final term corresponds to transitions driven between states $|\sigma\rangle$ and $|\sigma'\rangle$ by an ac microwave field. Here the transition dipole moment between two states $|\sigma\rangle$, $|\sigma'\rangle$ is $d_{\sigma\sigma'} \equiv \langle\sigma|\hat{d}_1|\sigma'\rangle$, where $\hat{d}_1 \equiv \hat{\mathbf{d}} \cdot \mathbf{e}_1$ is the projection of the dipole operator along the space-fixed spherical basis direction $\mathbf{e}_1 = -\left(\mathbf{e}_x + i\mathbf{e}_y\right)/\sqrt{2}$.

For ^{40}K^{87}Rb, which is the most experimentally relevant species, the energy scales of the various terms are summarized in Table 4.1. The detunings Δ_σ are determined chiefly by the linear Zeeman effect, and so are tunable by the dc magnetic field, and will be similar for other molecular species. The tunneling energy scale t_σ is set by the recoil energy, and so will be similar for other alkali dimers. The dipole-dipole energy scale $U_{\sigma\sigma'}$ is fixed by the permanent dipole moment, and so will change with the molecular species. For example, LiCs has a dipole moment roughly 10 times larger than that of KRb, and so $U_{\sigma\sigma'}$ will be of order 25 kHz. The scale of the ac term is determined by the power of the microwave field E_{ac}, which is readily tunable. The range of energies we have quoted represents the most interesting regime where the basic assumptions of our derivation hold.

In the following sections we will justify the HMHH and list the essential assumptions underlying its derivation, but we first pause to note some of its unusual properties.

4.2.1 Quantum Dephasing

The first property, which we call *quantum dephasing*, was investigated previously for a molecular Hubbard Hamiltonian involving only rotational degrees of freedom [15]. The effect, which is purely many-body in nature, may be summarized in this context as the destruction of coherent Rabi flopping due to the population of many spatial degrees of freedom in a many-body system driven at a single-molecule resonance. This effect is also of interest in the more general context of oscillations in a many-body system that are damped by some intrinsic mechanism following a quench [17, 18].

Dephasing is strongest when the Rabi frequency is on the order of the tunneling energies and the difference in tunneling energies for the two internal modes is also comparable to these two scales. For a system with two single-particle levels 0 and 1 and tunneling energies t_0 and t_1, respectively, this gives the condition $\Omega \sim t_0 \sim t_1 \sim |t_0 - t_1|$, which can be achieved with the HMHH for reasonable parameter values. The Rabi oscillations between the two internal states connected by the single molecule resonance damp out exponentially in time with an emergent time scale τ which can be measured experimentally, see Figure 4.1. Dephasing can be observed in the structure factors

$$S_{\pi}^{\sigma\sigma'} = \frac{1}{L} \sum_{i,j=1}^{L} (-1)^{i-j} \left\langle \hat{n}_{i\sigma} \hat{n}_{j\sigma'} \right\rangle , \tag{4.2}$$

where L is the number of lattice sites; $S_{\pi}^{\sigma\sigma'}$ can be measured in scattering experiments [19].

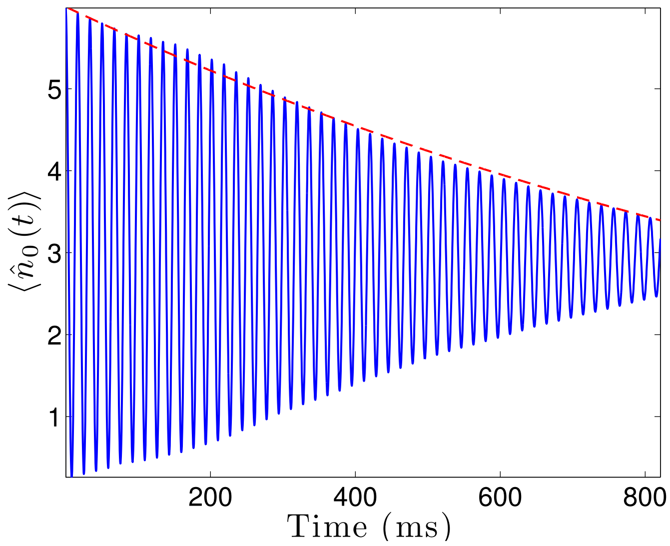


Figure 4.1: (Color online) *Quantum dephasing in the HMHH.* The plot shows the behavior of the total number in state 0: $\langle \hat{n}_0 \rangle \equiv \langle \sum_i \hat{n}_{i0} \rangle$ when the system evolves under the Hamiltonian (4.1). Quantum dephasing produces an emergent exponential envelope on the Rabi oscillation pattern between states 0 and 1. Only the number of state 0 is shown for clarity. The dashed red curve is an exponential envelope fit to $N \exp(-t/\tau)$ with $\tau = 1441.17$ms. The nonexponential behavior near $t = 200$ is due to the finite size of the lattice.

4.2.2 Internal State Dependence of Phase Diagram

The dependence of the tunneling energy t_σ on the internal state σ makes the borders of the phase diagram shift strongly (e.g. by a factor of 2). This dependence is shown explicitly in Figure 4.2. Thus, by preparing a collection of molecules in multiple internal states one can study interactions of many-body systems in different quantum phases and possibly far from equilibrium. Possibilities for quantum statics include studying the properties of phase equilibria as a function of population imbalance and effective mass (as determined by the tunneling energy) [21]. Also, as the difference in tunneling energy between different modes depends only on the elements of the molecular polarizability tensor, measuring the borders of the static phase diagram for different internal states also provides a means to measure this tensor. Possibilities for quantum dynamics include the study of quench phenomena for interacting many-body

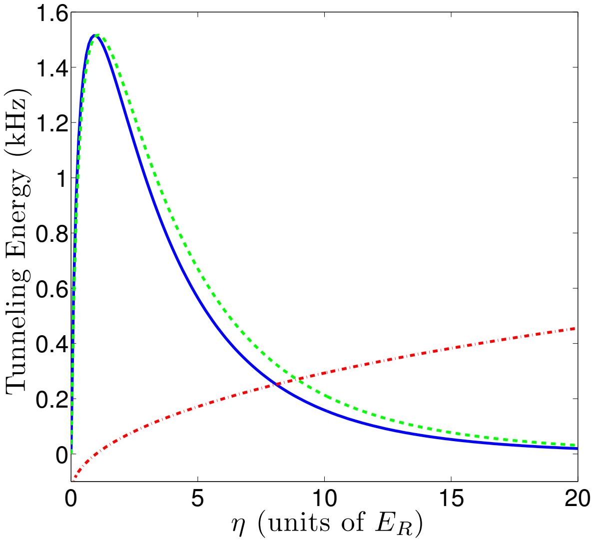


Figure 4.2: (Color online) *Tunneling matrix elements in a dc electric field.* Tunneling energies (in kHz) of the $N = 0$ (solid blue line) and $N = 1$ (dashed green line) rotational states and their difference divided by their arithmetic mean, $2\left(t_1 - t_0\right)/\left(t_1 + t_0\right)$, (dash-dotted red line) for KRb in a field of 10 kV/cm as a function of the effective isotropic lattice height $\eta \equiv \bar{\alpha}\left|\mathbf{E}_{\mathrm{opt}}\right|^2$ (in recoil energy units). The values of the polarizability tensor are taken from Ref. [20].

systems in different quantum phases.

4.2.3 Tunable Complexity

A final noteworthy property which was not present in the molecular Hubbard Hamiltonians previously studied is the possibility of *tunable complexity*. By complexity we mean that the system is comprised of many interacting degrees of freedom and displays emergent behavior such as the dephasing discussed above. Tunability refers to the fact that we may alter the *number* of internal degrees of freedom that are accessed dynamically as well as the *timescale* of their relative interactions. The key point for tunability is that the electric and magnetic fields affect different degrees of freedom: the electric dipole moment and nuclear spins, respectively. We illustrate this concept, and the corresponding geometries and polarizations needed for experiments, in Figure 4.3.

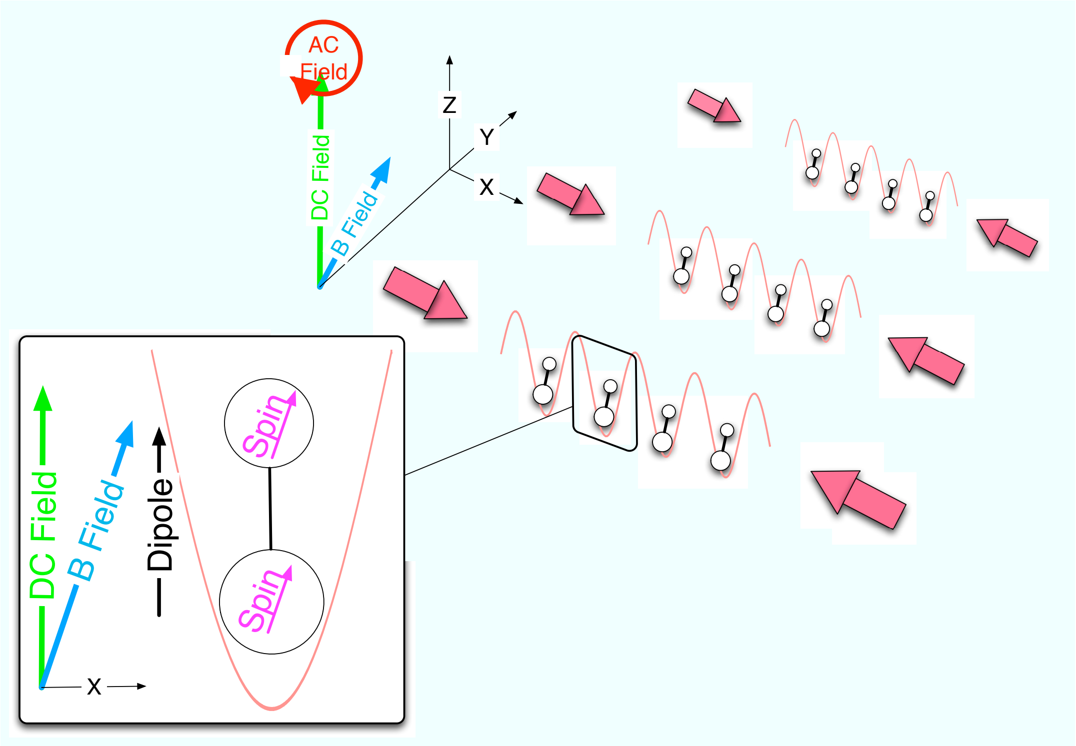


Figure 4.3: (Color online) *Geometry of the HMHH.* Counter-propagating laser beams along the y and z directions create an array of 1D tubes, and an additional pair of laser beams along x creates a lattice potential. A strong dc field orients the dipoles along the direction perpendicular to motion, and a magnetic field orients the nuclear spins. An ac field of circular space-fixed polarization drives transitions between internal levels.

In slightly more detail, tunability is achieved as follows. In the presence of an electric field aligned along the z direction, dipole moments are induced between states having the same nuclear spin projection along the field. The introduction of a strong magnetic field defines an effective axis of quantization for the nuclear spins while leaving the rotational structure unchanged because of the strong nuclear Zeeman effect, the weak rotational Zeeman effect, and the presence of only weak (quadrupole) coupling between the rotational and nuclear spin degrees of freedom. In the presence of a strong magnetic field that is not collinear with the electric field it is therefore possible to induce dipole moments between states with different hyperfine quantum numbers.

Thus, by changing the relative angle between the electric and magnetic fields one can control the number of states accessible from a particular state. The power of the applied ac field determines the interaction scale and the Rabi frequency of these dipole couplings, and the strength of the magnetic field determines the energetic splittings between states, in turn determining the relative rates of internal state population. The HMHH may therefore be used as a quantum simulator of a *quantum complex system* where the number and timescale of the internal components may be dynamically altered. Precise measures of complexity and simulations displaying characteristic behavior in various regimes will be discussed in future work [22].

4.3 Derivation of the Hyperfine Molecular Hubbard Hamiltonian

We consider the experimental setup shown schematically in Figure 4.3. Counter-propagating laser beams along the y and z directions create a series of 1D optical lattice "tubes." The intensity of these beams is such that the tubes are isolated from one another, and the lattice spacing is chosen (e.g. by crossed beams) such that the dipole-dipole interaction along y and z is negligible on experimental timescales. An additional pair of beams creates a lattice potential along the x-direction. The experimental techniques required to create this setup have been well established for ultracold atoms [23–25]. In addition to the lattice potential there is a uniform dc electric field along the z direction, a uniform magnetic field which lies in the xz plane, and an ac microwave field propagating in z which is assumed to have circular polarization $q = 1$ in the space-fixed spherical basis.

In the lattice is an ultracold quantum degenerate gas of $^1\Sigma$ heteronuclear molecules characterized by permanent electric dipole moment d, rotational constant B_N, rotational angular momentum $\mathbf{N}$ [25], and nuclear spins $\mathbf{I}_1$ and $\mathbf{I}_2$. Both nuclear spins are taken to be greater than one-half, so that both nuclei have nonzero electric quadrupole moments. In second quantization the full low-energy Hamiltonian for this setup is

---

[25] We reserve $\mathbf{J}$ for future studies involving nonzero orbital or electronic spin angular momentum.

$$\begin{aligned}\hat{H} \;=\; & \int d\mathbf{r}\,\hat{\psi}^{\dagger}(\mathbf{r})\left[\hat{H}_{\text{in}}+\hat{H}_{\text{F}}+\hat{H}_{\text{ac}}+\hat{H}_{\text{kin}}+\hat{H}_{\text{opt}}\right]\hat{\psi}(\mathbf{r}) \\ & +\frac{1}{2}\int d\mathbf{r}d\mathbf{r}'\,\hat{\psi}^{\dagger}(\mathbf{r})\,\hat{\psi}^{\dagger}(\mathbf{r}')\,\hat{H}_{\text{DD}}(|\mathbf{r}-\mathbf{r}'|)\,\hat{\psi}(\mathbf{r}')\,\hat{\psi}(\mathbf{r}),\end{aligned} \tag{4.3}$$

where

$$\begin{aligned}\hat{H}_{\text{in}} =& \hat{H}_{\text{rot}}+\hat{H}_{\text{scal}}+\hat{H}_{\text{tens}}+\hat{H}_{\text{r-s}}+\hat{H}_{\text{quad}} \\ =& B_N\hat{\mathbf{N}}^2+c_4\hat{\mathbf{I}}_1\cdot\hat{\mathbf{I}}_2+c_3\hat{\mathbf{I}}_1\cdot\tilde{T}\cdot\hat{\mathbf{I}}_2+\sum_{i=1}^{2}c_i\hat{\mathbf{N}}\cdot\hat{\mathbf{I}}_i \\ & +\sum_{i=1}^{2}\hat{\hat{\mathbf{V}}}_i\cdot\hat{\hat{\mathbf{Q}}}_i\,,\end{aligned} \tag{4.4}$$

$$\begin{aligned}\hat{H}_{\text{F}} =& -g_r\mu_N\hat{\mathbf{N}}\cdot\mathbf{B}-\sum_{i=1}^{2}g_i\mu_N\left(1-\sigma_i\right)\hat{\mathbf{I}}_i\cdot\mathbf{B} \\ & -\mathbf{E}_{\text{dc}}\cdot\hat{\mathbf{d}}\,,\end{aligned} \tag{4.5}$$

$$\hat{H}_{\text{ac}} = -\mathbf{E}_{\text{ac}}\cdot\hat{\mathbf{d}}\,, \tag{4.6}$$

$$\hat{H}_{\text{kin}} = \frac{\hat{\mathbf{p}}^2}{2m}\,, \tag{4.7}$$

$$\hat{H}_{\text{opt}} = -\mathbf{E}^{\star}_{\text{opt}}\cdot\hat{\hat{\alpha}}\left(\omega_{\text{opt}}\right)\cdot\mathbf{E}_{\text{opt}}\,, \tag{4.8}$$

$$\hat{H}_{\text{DD}}\left(R\right) = \frac{\hat{\mathbf{d}}_1\cdot\hat{\mathbf{d}}_2-3\left(\hat{\mathbf{d}}_1\cdot\mathbf{e}_R\right)\left(\mathbf{e}_R\cdot\hat{\mathbf{d}}_2\right)}{R^3}\,. \tag{4.9}$$

The first line of Eq. (4.3) is comprised of single-molecule terms. In order, these are $\hat{H}_{\text{in}}$, the Hamiltonian governing the internal rotational and nuclear spin degrees of freedom; $\hat{H}_{\text{F}}$, the interaction of the molecule with externally applied dc electric and magnetic fields; $\hat{H}_{\text{ac}}$, the interaction of the molecule with an ac microwave field; $\hat{H}_{\text{kin}}$, the kinetic energy of the molecule; and $\hat{H}_{\text{opt}}$, the interaction of the molecule with the optical lattice potential. The second line of Eq. (4.3) is the two-molecule resonant dipole-dipole force. The main assumptions underlying this Hamiltonian and our subsequent analysis are the following.

First, the individual molecules are assumed to be in their electronic and vibrational ground states, and it is assumed that none of these degrees of freedom can be excited at the large intermolecular separations and low temperatures/relative energies that we consider.

Second, the characteristic trapping potential length is chosen large enough compared to the internuclear axis to assume spherical symmetry, i.e. a locally constant potential.

Third, we consider only the lowest two rotational levels. All ac fields will be sufficiently weak to allow this assumption. We also work in the rotating wave approximation, which requires that the detuning be small compared to the driving frequency.

Fourth, we consider all molecules to be in the lowest Bloch band. The ac Rabi frequencies are chosen to be small (∼1-50 kHz) in comparison with the lattice bandwidth ($\sim 10 E_R \sim$100 kHz) to ensure this assumption.

Fifth, we work in the "hard-core" limit where at most one molecule is allowed per site. This is enforced by strongly repulsive dipole-dipole interactions on-site, caused by our z-alignment of the electric field, as sketched in Figure 4.3. We consider the lattice spacing large enough to include only nearest-neighbor dipole-dipole interactions. We neglect the effects of chemical reactions or hyperfine changing collisions which occur at very short range.

Sixth, we neglect dipole-dipole interactions between molecules in different 1D "tubes." For a consistent level of approximation this requires the tubes to be separated by twice the lattice spacing. This can be achieved in principle using crossed beams to create larger lattice spacings.

Seventh, we consider only pairwise interactions of the molecules, neglecting three and higher-body interactions. This is valid for KRb because the permanent dipole moment $d = 0.566$D is rather small. For molecules such as LiCs with larger permanent dipole moments, the three-body interaction can play a significant role [26].

To derive a Hamiltonian of Hubbard type from Eq. (4.3) we follow the standard prescription [13] of expanding the field operators of our second-quantized Hamiltonian in a Wannier basis of single-molecule states centered at a particular discrete position $\mathbf{r}_i$:

$$\hat{\psi} \;= \sum_i \sum_\sigma \hat{a}_{i\sigma} w_\sigma \left(\mathbf{r}-\mathbf{r}_i\right) \,, \tag{4.10}$$

where i is a site index and σ an index denoting the internal state of the molecule. The Wannier basis we use is the basis which diagonalizes the internal plus static field Hamiltonians $\hat{H}_{\mathrm{in}}+\hat{H}_{\mathrm{F}}$ and in which all states with $N=1$ rotate with frequency ω, where ω is the frequency of the applied ac electric field. With the field operator written in this manner, we find the Hubbard parameters

$$t_\sigma \equiv -\int d\mathbf{r}\, w^\star_\sigma \left(\mathbf{r}-\mathbf{r}_i\right) \left[\hat{H}_{\mathrm{kin}}+\hat{H}_{\mathrm{opt}}\right] w_\sigma \left(\mathbf{r}-\mathbf{r}_{i+1}\right) \,, \tag{4.11}$$

$$\Delta_\sigma \equiv \int d\mathbf{r}\, w^\star_\sigma \left(\mathbf{r}-\mathbf{r}_i\right) \left[\hat{H}_{\mathrm{in}}+\hat{H}_{\mathrm{F}}\right] w_\sigma \left(\mathbf{r}-\mathbf{r}_i\right) \,, \tag{4.12}$$

and

$$-d_{\sigma\sigma'}E_{\mathrm{ac}} \equiv \int d\mathbf{r}\, w^\star_\sigma \left(\mathbf{r}-\mathbf{r}_i\right) \hat{H}_{\mathrm{ac}} w_{\sigma'} \left(\mathbf{r}-\mathbf{r}_i\right) \,, \tag{4.13}$$

$$\begin{aligned} U_{\sigma\sigma'} \equiv & \int d\mathbf{r}d\mathbf{r}'\, w^\star_\sigma \left(\mathbf{r}-\mathbf{r}_i\right) w^\star_{\sigma'} \left(\mathbf{r}'-\mathbf{r}_{i+1}\right) \\ & \times H_{\mathbf{DD}} \left(\mathbf{r}-\mathbf{r}'\right) w_\sigma \left(\mathbf{r}-\mathbf{r}_i\right) w_{\sigma'} \left(\mathbf{r}'-\mathbf{r}_{i+1}\right) \,. \end{aligned} \tag{4.14}$$

The detunings Δ_σ are determined by the single-molecule spectra, which are well-known [9, 27]. In the interest of the present article's completeness, we have included appendices reviewing the basic results and explaining them in the context of the present problem. In the following sections we discuss the remaining Hubbard parameters.

4.3.1 Tunneling Energies

A key component of the realization of many-body Hamiltonians using ultracold molecules is the presence of a far off-resonant optical lattice which confines the molecules in a reduced geometry. The Hamiltonian of this interaction is

$$\hat{H}_{\mathrm{opt}} = -\mathbf{E}^{\star}_{\mathrm{opt}}\left(\mathbf{r}, \omega_{\mathrm{opt}}\right) \cdot \hat{\tilde{\alpha}}\left(\omega_{\mathrm{opt}}\right) \cdot \mathbf{E}_{\mathrm{opt}}\left(\mathbf{r}, \omega_{\mathrm{opt}}\right) , \tag{4.15}$$

where $\mathbf{E}_{\mathrm{opt}}\left(\mathbf{r}, \omega_{\mathrm{opt}}\right)$ is the optical lattice field and $\hat{\tilde{\alpha}}\left(\omega_{\mathrm{opt}}\right)$ is the polarizability tensor operator of the molecule, evaluated at the optical lattice frequency ω_{opt}. In our notation, the circumflex accent (the 'hat') denotes an operator, the tilde denotes a rank 2 tensor, and boldface denotes a rank 1 tensor, or vector. This optical potential couples to the electronic degrees of freedom and is detuned from resonance by an amount several orders of magnitude larger than any hyperfine splittings. Thus dependence on the hyperfine quantum numbers in negligible. For tight optical traps, the optical trap potential at each well is close to that of a harmonic trap plus a small state-dependent tensor shift of the trap frequency affecting levels with $N > 0$ due to the polarizability anisotropy [15].

When the optical potential is combined with the kinetic portion of the Hamiltonian and evaluated in the Wannier basis one obtains the tunneling energies. As the tunneling energies are independent of the hyperfine quantum numbers, we can use results obtained in the case of only rotational degrees of freedom, derived in our earlier work [15]. Then the tunneling energies in the eigenbasis of $\hat{H}_{\mathrm{rot}}$, $|NM_N\rangle$, are given by

$$\frac{\tilde{t}_{NM_N}}{E_R} = A\left(\frac{V_{NM_N}}{E_R}\right)^{B} \exp\left(-C\sqrt{\frac{V_{NM_N}}{E_R}}\right) \tag{4.16}$$

where $A = 1.397$, $B = 1.051$, and $C = 2.121$ are fit parameters [28], E_R the recoil energy, and

$$V_{NM_N}=\left|\mathbf{E}_{\text{opt}}\right|^2\left[\bar{\alpha}+\frac{2\Delta\alpha}{3}\frac{N\left(N+1\right)-3M_N^2}{\left(2N-1\right)\left(2N+3\right)}\right] \tag{4.17}$$

is the effective lattice height for the $|NM_N\rangle$ level. Here $\bar{\alpha}$ is the average polarizability and $\Delta\alpha$ the polarizability anisotropy.

In the presence of a dc field the rotational levels become mixed, leading to new effective tunneling energies which we denote as t_{NM_N}, with N and M_N the corresponding zero field values. This hybridization of rotational levels in principle also allows tunneling events which change the rotational state of the molecule, but we can ignore such events because the rotational level separation is much larger than the tunneling energies. The effective tunneling for the $N=0$ and $N=1$, $M_N=\pm1$ levels is shown in Figure 4.2. The scale is set by the recoil energy, which is $2\pi\times 1.44\,\text{kHz}$ for KRb in a 1054-nm optical lattice.

4.3.2 Two-Molecule Interactions

Heteronuclear $^1\Sigma$ molecules posses permanent dipole moments, and thus interact via a dipole-dipole interaction

$$\hat{H}_{\text{DD}}\left(\mathbf{R}\right)=\frac{\hat{\mathbf{d}}_1\cdot\hat{\mathbf{d}}_2-3\left(\hat{\mathbf{d}}_1\cdot\mathbf{e}_R\right)\left(\mathbf{e}_R\cdot\hat{\mathbf{d}}_2\right)}{R^3}\,, \tag{4.18}$$

where $\mathbf{R}\equiv\mathbf{r}_2-\mathbf{r}_1$, $\mathbf{e}_R$ is a unit vector in the direction of $\mathbf{R}$, and $\hat{\mathbf{d}}_i$ is the vector dipole operator of the i^{th} molecule. In the absence of external fields, this interaction is off-resonant, leading to a van der Waals interaction $\hat{H}_{\text{DD}}\left(\mathbf{R}\right)\sim R^{-6}$, but in the presence of electric fields resonant dipoles are induced and the interaction displays a resonant R^{-3} behavior in addition to the R^{-6} behavior.

The anisotropic nature of the dipole-dipole force has been experimentally shown to dominate the rethermalization behavior of a molecular gas via inelastic collisions [29]. This is because a “head-to-tail” arrangement of molecules leads to an attractive potential, whereas "side-to-side" interactions are repulsive. To ensure the stability of an ultracold molecular ensemble and to prevent losses from inelastic collisions it is crucial therefore not only to orient the dipoles using a dc field, but also to confine the molecules in a reduced geometry. A thorough discussion of the nature of the two-molecule spectra for ${}^1\Sigma$ molecules without hyperfine structure and its implications for stability in two dimensions is presented in Ref. [14]. Diagonalization of the full two-molecule Hamiltonian is impractical when hyperfine structure is included due to the very large matrices that result. Instead, we argue based on comparisons of length and energy scales that the hyperfine structure is negligible during the collisional processes which occur in our proposed setup.

Our reduced geometry is imposed by the optical lattice described earlier. Namely, we consider the case where the molecules are confined to move only along the x direction and a dc field polarized along the z direction orients the dipoles such that all collisions are side-to-side and repulsive. The dipole-dipole interaction in this geometry reduces to

$$\begin{aligned}\hat{H}_{\mathrm{DD}} =&\frac{1}{R^3}\Big[\hat{d}_0\otimes\hat{d}_0+\frac{1}{2}\left(\hat{d}_{-1}\otimes\hat{d}_1+\hat{d}_1\otimes\hat{d}_{-1}\right)\\ &-3\left(\hat{d}_{-1}\otimes\hat{d}_{-1}+\hat{d}_1\otimes\hat{d}_1\right)\Big],\end{aligned} \tag{4.19}$$

where $\hat{d}_q \equiv \hat{\mathbf{d}}\cdot\mathbf{e}_q$ is the component of the dipole operator along the q direction in the space-fixed spherical basis. For z-polarized electric field, the only diagonal components are those involving d_0. The components of the interaction involving $d_{\pm1}$ couple states with $\Delta M_N = \pm1$ that are separated in energy by an amount of order the rotational constant for the dc fields we consider (see Figure 4.5). Contributions from these components are suppressed at distances greater than $r_B \equiv \left(d^2/B\right)^{1/3}$, of order a few nanometers. Thus, at the nearest-neighbor distance in a 1054-nm optical lattice we consider only the diagonal elements of the dipole-dipole interaction. This restriction gives rise to the two-body term

$$\hat{H}_{\mathrm{DD}} = \frac{1}{2}\sum_{\sigma\sigma'} U_{\sigma\sigma'} \sum_{\langle i,j\rangle} \hat{n}_{i\sigma}\hat{n}_{j\sigma'} \, , \tag{4.20}$$

where

$$U_{\sigma\sigma'} = \frac{d_\sigma d_{\sigma'}}{(\lambda/2)^3} \, . \tag{4.21}$$

In Eq. (4.21) d_σ is the resonant dipole moment of state $|\sigma\rangle$ and λ is the wavelength of the optical lattice. We assume that the long-range repulsive diagonal d_0 portion of the dipole-dipole interaction is strong enough to prevent both the occupation of any one lattice site by more than one molecule and access to the region where hyperfine-changing collisions involving the $d_{\pm 1}$ dipole moments occur.

4.3.3 Interactions with Static External Fields

The spectral properties of ${}^1\Sigma$ molecules in collinear dc electric and magnetic fields have been elucidated elsewhere in the literature [9, 10, 27], and the basic results of the analysis are given in Appendix 4.6 for the reader's convenience. In this section, we focus on the properties of such molecules in *noncollinear* fields, in particular on the dipole moments.

The behavior of the molecular dipole moments are controlled by an external dc electric field which mixes rotational levels of opposite parity and thus orients the molecule. However, a dc field does not couple to the nuclear spins. So for a z-polarized field the selection rules $\Delta M_1 = 0$, $\Delta M_2 = 0$ are enforced, where M_1 and M_2 are the nuclear spin projections along the field direction. In contrast, a magnetic field couples strongly to the nuclear spins but only weakly to the rotational angular momentum due to the relative sizes of the g factors [27]. The magnetic field Hamiltonian thus has eigenstates which are energetically distinct nuclear spin states with a quantization axis given by the field direction. It is in this sense that we say the magnetic field

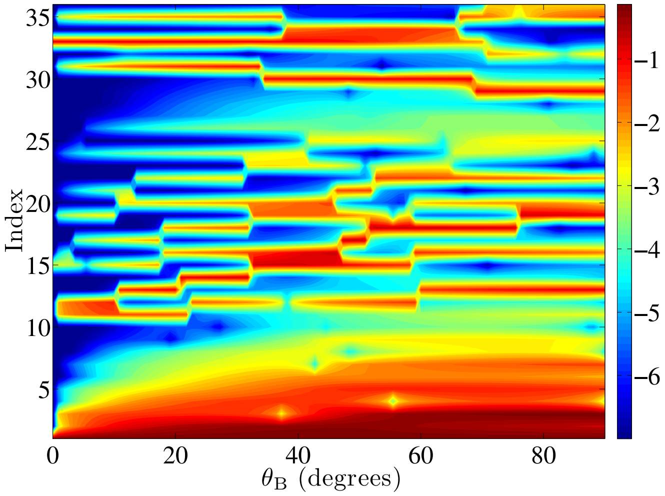


Figure 4.4: (Color online) *Distribution of dipolar character.* The colorbar shows the logarithm of the transition dipole moment with the ground state, $\log\langle \text{g.s.}|\hat{d}_1|i\rangle$, as a function of the angle between the magnetic and electric fields θ_B and the state index(ordered by energy). Changing the angle between the electric and magnetic fields breaks the nuclear spin projection selection rule and allows for transition dipole moments between many states. Only dipole moments greater than 10^{-7} are displayed.

defines an effective axis of quantization for the nuclear spins. Thus, in the absence of internal couplings of the rotational and hyperfine degrees of freedom they may be manipulated independently: the rotational angular momentum with an electric field and the nuclear spin angular momenta with a magnetic field.

The presence of nuclear quadrupole couplings in alkali dimer molecules couples states with the same total angular momentum projection M_F but different rotational and nuclear spin projections. For example, in KRb, the interaction couples $|\sigma\rangle = |N=1, M_N=0, M_{\mathrm{K}}, M_{\mathrm{Rb}} \pm 1\rangle$ to $|\sigma'\rangle = |N=1, M_N=\pm 1, M_{\mathrm{K}}, M_{\mathrm{Rb}}\rangle$ with the latter accounting for $\sim 10\%$ of the state in the absence of fields.[26] Clearly, since the $N=0$ state has only one projection $M_N=0$, the nuclear quadrupole interaction leaves this level unaffected. In a dc electric field where the rotational levels become mixed, the states correlating with the $N=0$ levels and $N=1$ levels both display quadrupole

---

[26] The interaction also couples the $M_{\mathrm{K}} \pm 1$ states, but the coupling constant $(eqQ)_{\mathrm{K}}$ is significantly smaller than $(eqQ)_{\mathrm{Rb}}$ and so the mixing is negligible in comparison [11].

effects, but these effects are still not identical. In strong fields the Zeeman effect dominates over the quadrupole coupling, allowing control over the nuclear spins that displays a weak dependence on the rotational level.

Thus, a strong magnetic field defines an effective axis of quantization for the nuclear spins, resulting in nuclear spin states which are superpositions of states in the basis with the axis of quantization along the electric field axis. This implies that by changing the angle of the magnetic field with respect to the electric field, it is possible to change the number of states which are coupled by transition dipole moments. This is illustrated in Figure 4.4, which shows the logarithm of the transition dipole moment with the ground state as a function of the angle between the dc and magnetic fields θ_B and a state index (ordered by energy). The lowest state index denotes the lowest energy state in the $N = 1$ manifold. When the fields are collinear, one state dominates the dipole spectrum. As the angle changes the dipolar character becomes spread over many states. These transition dipole moments allow the states to couple in an ac microwave field and generate complex dynamics.

4.3.4 Interaction with an ac Microwave Field

The introduction of an ac microwave field contributes to the Hamiltonian in a similar way to a dc field. In addition, the inherent time dependence allows for circular and linear polarization as well as the possibility of driving transitions between internal states. In the absence of hyperfine structure, an ac field of spherical polarization q couples the $|N = 0, M_N = 0\rangle$ and $|N = 1, M_N = q\rangle$ levels, leading to an effective two-level system in the Floquet picture [14]. In the presence of hyperfine structure, states with different total angular momentum projections M_F in the $N \geq 1$ manifolds become mixed due to the electric quadrupole interaction. Thus no rigorous selection rules can be established. This complicates the issue of addressing single hyperfine states using microwave fields, but it also allows the hyperfine state to be changed using microwave fields. Addressing a single hyperfine state can be achieved by the application of a strong magnetic field such as those used in the STIRAP procedure, which defines the projections sufficiently to suppress transitions to non-target hyperfine states [9]. In the presence of an electric field, this last comment holds only in the case where the two fields are collinear. When the fields are not collinear many states can be accessed from any one state via a microwave transition due to the behavior of the transition dipole moments in crossed fields, as was described in Sec. 4.3.3.

We choose the polarization of the ac field to be purely circular, $q_{\mathrm{ac}} = 1$. A component along $q = 0$ would lead to rapid oscillation of the eigenenergies because the d_0 moments induced by the electric field couple to the ac field, and this complicates the analysis. Furthermore, we consider Rabi frequencies which are much less than the bandwidth of the optical lattice so that our approximation of being in the lowest Bloch band remains valid and we are also justified in using a rotating wave approximation. The above considerations together with the single-molecule ac Hamiltonian

$$\hat{H}_{\mathrm{ac}} = -\hat{\mathbf{d}} \cdot \mathbf{E}_{\mathrm{ac}} = -\hat{d}_q E_{\mathrm{ac}} e^{-i\omega t} + \mathrm{h.c.} \tag{4.22}$$

lead directly to the second quantized Hamiltonian

$$\hat{H}_{\mathrm{ac}} = -\frac{1}{2} \sum_{\sigma\sigma'} d_{\sigma\sigma'} E_{\mathrm{ac}} \sum_i \left[\hat{a}^{\dagger}_{i\sigma} \hat{a}_{i\sigma'} e^{i\omega t} + \mathrm{h.c.} \right] . \tag{4.23}$$

In Eq. (4.23) the label σ refers to the eigenstate $|\sigma\rangle$ of the internal plus static field Hamiltonian $\hat{H}_{\mathrm{in}} + \hat{H}_{\mathrm{F}}$, $d_{\sigma\sigma'} \equiv \langle \sigma | d_1 | \sigma' \rangle$, and E_σ is the energy of state $|\sigma\rangle$.

Assembling all the many-body terms expressed in this basis, we obtain the time-dependent Hamiltonian

$$\begin{aligned}\hat{H} = &\sum_{\sigma} E_{\sigma} \sum_{i} \hat{n}_{i\sigma} - \sum_{\sigma} t_{\sigma} \sum_{\langle i,j \rangle} \left[\hat{a}^{\dagger}_{i\sigma} \hat{a}_{j\sigma} + \text{h.c.} \right] \\ &+ \frac{1}{2} \sum_{\sigma,\sigma'} U_{\sigma\sigma'} \sum_{\langle i,j \rangle} \hat{n}_{i\sigma} \hat{n}_{j\sigma'} \\ &- \frac{1}{2} \sum_{\sigma\sigma'} d_{\sigma\sigma'} E_{\text{ac}} \sum_{i} \left[\hat{a}^{\dagger}_{i\sigma} \hat{a}_{i\sigma'} e^{i\omega t} + \text{h.c.} \right] . \qquad (4.24)\end{aligned}$$

If we change to a basis where all single-molecule states with $N = 1$ rotate with frequency ω we have, finally:

$$\begin{aligned}\hat{H} = &\sum_{\sigma} \Delta_{\sigma} \sum_{i} \hat{n}_{i\sigma} - \sum_{\sigma} t_{\sigma} \sum_{\langle i,j \rangle} \left[\hat{a}^{\dagger}_{i\sigma} \hat{a}_{j\sigma} + \text{h.c.} \right] \\ &+ \frac{1}{2} \sum_{\sigma,\sigma'} U_{\sigma\sigma'} \sum_{\langle i,j \rangle} \hat{n}_{i\sigma} \hat{n}_{j\sigma'} \\ &- \frac{1}{2} \sum_{\sigma\sigma'} d_{\sigma\sigma'} E_{\text{ac}} \sum_{i} \left[\hat{a}^{\dagger}_{i\sigma} \hat{a}_{i\sigma'} + \text{h.c.} \right] , \qquad (4.25)\end{aligned}$$

where $\Delta_{\sigma} = E_{\sigma}$ for states with $N = 0$ and $E_{\sigma} - \omega$ for states with $N = 1$.

4.4 Conclusions

We have presented and derived the hyperfine molecular Hubbard Hamiltonian (HMHH). The HMHH is a lattice Hamiltonian describing the effective low-energy physics of an ultracold gas of heteronuclear alkali dimer molecules with hyperfine structure loaded into a 1D optical lattice and interacting with external dc electric, ac microwave, and static magnetic fields. By tuning the angle between the electric and magnetic fields and the strength of the magnetic and ac fields it is possible to change the number and timescale of internal states contributing to many-body dynamics. The Hamiltonian also displays emergent quantum dephasing, and has a phase diagram which depends strongly on the initial state. These features make the HMHH an ideal candidate for a model quantum complex system.

Future work will involve time-evolving block decimation simulations of the HMHH similar to past studies of molecular Hubbard Hamiltonians [15]. In particular, we will discuss measures of complexity and how they relate to experimentally measurable quantities. Future work on the Hamiltonian itself will include realistic models of molecule loss due to inelastic and chemical processes. Such dissipative processes are key to dissipative quantum phase transitions, which is a major area of interest in quantum many-body theory [30–32].

We acknowledge useful discussions with Immanuel Bloch, John Bohn, Silke Ospelkaus, Luis Santos, and Peter Zoller. This work was supported by the National Science Foundation under Grant PHY-0903457.

4.5 The Internal Hamiltonian

A $^1\Sigma$ molecule in its electronic and vibrational ground states has three angular momentum degrees of freedom: the rotational angular momentum $\mathbf{N}$ and the nuclear spins $\mathbf{I}_1$ and $\mathbf{I}_2$. In this work we shall use the coupling schemes $|\,(I_1I_2)\,INFM_F\rangle$ and $|I_1M_1I_2M_2NM_N\rangle$, which we refer to as the coupled and uncoupled bases, respectively. Explicit expressions for all single-molecule matrix elements in both bases are provided in Appendix 4.7. The relevant Hamiltonian for the internal degrees of freedom $\hat{H}_{\mathrm{in}}$ may be written as a sum of rotational and hyperfine terms as

$$\hat{H}_{\mathrm{in}} = \hat{H}_{\mathrm{rot}} + \hat{H}_{\mathrm{hf}} \tag{4.26}$$

where

$$\hat{H}_{\mathrm{rot}} = B_N\mathbf{N}^2\,, \tag{4.27}$$

$$\hat{H}_{\mathrm{hf}} = \sum_{i=1}^{2} c_i\mathbf{N}\cdot\mathbf{I}_i + c_3\mathbf{I}_1\cdot\tilde{T}\cdot\mathbf{I}_2 + c_4\mathbf{I}_1\cdot\mathbf{I}_2 + \sum_{i=1}^{2}\mathbf{V}_i\cdot\mathbf{Q}_i\,. \tag{4.28}$$

The rotational term Eq. (4.27) corresponds to the Hamiltonian of a rigid spherical rotor with $(2N+1)$-fold degenerate eigenstates $|NM_N\rangle$, M_N being the projection of **N** on a space-fixed quantization axis [33]. The eigenenergies are given by $E_{NM_N} = B_N N(N+1)$, where B_N is the rotational constant of the molecule (we use the notation B_N instead of the more common B to avoid confusion with the magnetic field magnitude B). In the case of ^{40}K^{87}Rb, B_N=1.114 GHz [27]. The rotational level splitting defines the largest intrinsic energy scale for $^1\Sigma$ molecules.

The first term of the hyperfine Hamiltonian, $\sum_{i=1}^{2} c_i \mathbf{N}\cdot\mathbf{I}_i$ represents the interaction of the nuclear spins with the magnetic field created by the rotation of the molecule, and is governed by two coupling constants c_{K} and c_{Rb} related to the nuclear shielding tensor. For ^{40}K^{87}Rb, these have been determined from density functional calculations to be ∼20 Hz and ∼100 Hz, respectively [27]. Because of the smallness of these constants and the fact that this term does not couple states with different N, this term plays a very small role in the spectra.

The two nuclear spins have nuclear magnetic moments which interact via a resonant dipole-dipole interaction

$$\hat{H}_{\mathrm{hf-dd}} = g_H^2 \mu_N^2 \left(\mu_0/4\pi\right) \left[\frac{\mathbf{I}_1\cdot\mathbf{I}_2}{R^3} - \frac{3\left(\mathbf{I}_1\cdot\mathbf{R}\right)\left(\mathbf{R}\cdot\mathbf{I}_2\right)}{R^5}\right], \tag{4.29}$$

where g_H is the proton g factor and $\mathbf{R}$ the vector joining the two nuclei [33]. This may be written as the contraction of two rank-2 spherical tensors as

$$\hat{H}_{\mathrm{hf-dd}} = -g_H^2 \mu_N^2 \left(\mu_0/4\pi\right) \langle R^{-3}\rangle \sqrt{6} \left(\mathbf{C}\right)^{(2)} \cdot \left(\mathbf{T}\left(\mathbf{I}_1, \mathbf{I}_2\right)\right)^{(2)} \tag{4.30}$$

where $\left(\mathbf{C}\right)^{(2)}$ is an unnormalized spherical harmonic in the relative degrees of freedom. The nuclear spins can also interact indirectly through the electron spins, and do so even for $^1\Sigma$ configurations [33]. This indirect interaction is represented by a tensor $\tilde{J}$ which may be decomposed into its isotropic part J_{iso} and its anisotropy $\Delta J = J_{\parallel} - J_{\perp}$. The combination of direct and indirect nuclear spin-nuclear spin interaction may thus be written as the sum of a scalar interaction and a tensor interaction as

$$\hat{H}_{\mathrm{hf-dd}} + \hat{H}_{\mathrm{indirect}} = c_4 \mathbf{I}_1 \cdot \mathbf{I}_2 + c_3 \mathbf{I}_1 \cdot \tilde{T} \cdot \mathbf{I}_2 \tag{4.31}$$

where $c_4 \equiv J_{\mathrm{iso}}$, $c_3 \equiv g_H^2 \mu_N^2 \left(\mu_0/4\pi\right) \langle R^{-3} \rangle - \Delta J/3$, and the tensor $\tilde{T}$ contains the angular dependence of the tensor interaction. c_3 is of order 10 Hz for the various isotopes of KRb, and so plays a very small role in the spectra. c_4 splits the various levels according to their total nuclear spin I as

$$\begin{aligned} &\langle (I_1 I_2)\, INFM_F | c_4 \mathbf{I}_1 \cdot \mathbf{I}_2 |\, (I_1 I_2)\, I' N' F' M_F' \rangle \\ &= \delta_{I,I'} \delta_{N,N'} \delta_{F,F'} \delta_{M_F,M_F'} \\ &\times \frac{c_4}{2} \left[I\left(I+1\right) - I_1\left(I_1+1\right) - I_2\left(I_2+1\right) \right] . \end{aligned} \tag{4.32}$$

c_4 is of order 100 Hz-10 kHz for isotopes of KRb, and so is the dominant hyperfine contribution for $N = 0$ in the absence of external fields, see Figure 4.7. Note that c_4 may be either positive or negative. For ^{40}K^{87}Rb, c_4=-20.304 kHz [27], and so the lowest energy states for $N = 0$ in zero field are the highest nuclear spin states $I = 11/2$.

The final term in the hyperfine Hamiltonian is the interaction of the quadrupole moment of the nuclei with the gradient of the electric field produced by the electrons. We may represent this interaction by the sum $\sum_{i=1}^{2} \mathbf{V}_i \cdot \mathbf{Q}_i$ where $\mathbf{V}_i$ is a second rank spherical tensor describing the electric field gradient at the i^{th} nucleus and $\mathbf{Q}_i$ is a second rank spherical tensor describing the nuclear quadrupole of the i^{th} nucleus. The pertinent coupling constants $(eqQ)_i$ which arise in the matrix elements of this Hamiltonian are of order 100-1000 kHz, making it the largest term in the hyperfine Hamiltonian. The quadrupole term doesn't affect the $N = 0$ level, however, and so the scalar spin-spin coupling dominates there. In a strong dc field the rotational levels become deeply mixed and the nuclear quadrupole thus becomes the dominant hyperfine contribution for all states.

4.6 Interactions with Static External Fields

Polar molecules such as heteronuclear dimers can couple to external fields either through their permanent electric dipole moment, through magnetic moments generated from their rotation or nuclear spin, or through their polarizability tensor. The Hamiltonian representing interaction of the molecule with a static dc electric field $\mathbf{E}_{\mathrm{dc}}$ and a static magnetic field $\mathbf{B}$ may be written

$$\hat{H}_{\mathrm{F}} = -\,\mathbf{d}\cdot\mathbf{E}_{\mathrm{dc}} - g_r\mu_N\mathbf{N}\cdot\mathbf{B} - \sum_{i=1}^{2} g_i\mu_N\mathbf{I}_i\cdot\mathbf{B}\left(1-\sigma_i\right) . \tag{4.33}$$

For $^1\Sigma$ molecules the permanent dipole moment $\mathbf{d}$ lies along the internuclear axis which defines the $p = 0$ axis in a spherical coordinate system rotating with the molecule. Because this basis leads to anomalous commutation relations $[J_i, J_k] = -i\hbar\epsilon_{ijk}J_k$ [34] we find it convenient to transform to the space-fixed frame where the angular momentum operators satisfy the normal commutation relations $[J_i, J_k] = i\hbar\epsilon_{ijk}J_k$, giving $\mathbf{d}\cdot\mathbf{e}_q \equiv d_q = dC_q^{(1)}(\theta,\phi)$, where $\mathbf{e}_q$ is a unit vector along the space fixed spherical q direction and $C_q^{(1)}(\theta,\phi)$ is an unnormalized spherical harmonic whose arguments θ and ϕ are the polar and azimuthal angles of the internuclear axis in the space fixed frame. Taking matrix elements of d_q in our two basis sets yields

$$\begin{aligned}&\langle I_1M_1I_2M_2NM_N|d_q|I_1M_1'I_2M_2'N'M_N'\rangle\\ &= \delta_{M_1,M_1'}\delta_{M_2,M_2'}d\sqrt{(2N+1)(2N'+1)}\,(-1)^{M_N}\\ &\times\begin{pmatrix} N & 1 & N' \\ 0 & 0 & 0\end{pmatrix}\begin{pmatrix} N & 1 & N' \\ -M_N & q & M_N'\end{pmatrix} ,\end{aligned} \tag{4.34}$$

$$\begin{aligned}&\langle (I_1I_2)\,INFM_F|d_q|\,(I_1I_2)\,I'N'F'M_F'\rangle\\ &= \delta_{I,I'}d\,(-1)^{2F-M_F+I+N'+N+1}\begin{pmatrix} N & 1 & N' \\ 0 & 0 & 0\end{pmatrix}\\ &\times\sqrt{(2N+1)(2N'+1)(2F+1)(2F'+1)}\\ &\times\begin{pmatrix} N & 1 & N' \\ -M_N & q & M_N'\end{pmatrix}\begin{Bmatrix} N & F & I \\ F' & N' & 1\end{Bmatrix} ,\end{aligned} \tag{4.35}$$

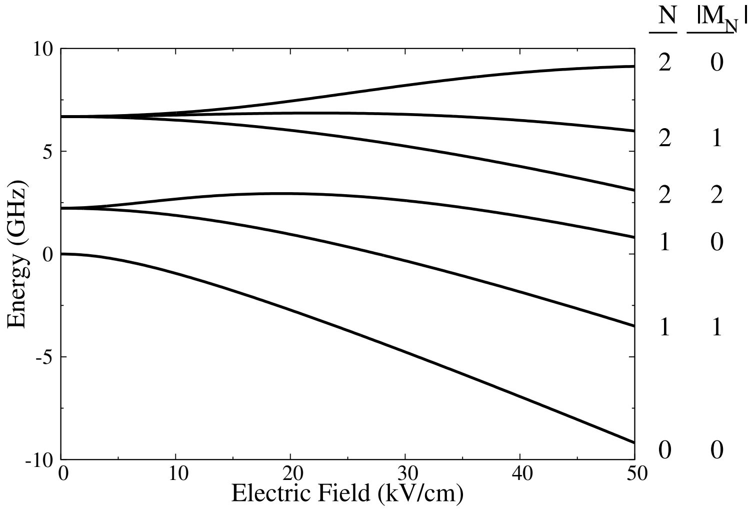


Figure 4.5: *GHz scale view of the Stark effect for KRb.* Introduction of a dc field breaks the degeneracy between all states with the same N but different $|M_N|$. The large electric dipole moment causes GHz scale energy shifts which completely obscure the hyperfine splittings on the scale of this plot. Because the dipole moment is the same for any isotope of KRb, the Stark effect on this scale is the same for all isotopes.

where $\begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix}$ is a Wigner 3-j coefficient and $\begin{Bmatrix} j_1 & j_2 & j_3 \\ j_4 & j_5 & j_6 \end{Bmatrix}$ is a Wigner 6-j coefficient [35]. We see that the rotational eigenstates have no net dipole moment, but that the dipole operator couples the state $|N, F, M_F\rangle$ with the states $|N \pm 1, F \pm 1, M_F + q\rangle$. The introduction of a dc electric field $\mathbf{E}_{\mathrm{dc}}$ with Hamiltonian $-\mathbf{d} \cdot \mathbf{E}_{\mathrm{dc}}$ couples these levels and induces dipole moments, breaking the rotational symmetry and removing the $(2N + 1)$-fold degeneracy. Typical molecular dipole moments are measured in Debye (D), where 1D=503.4 MHz/(kV/cm), and so the dc field becomes the dominant contribution to the Hamiltonian for modest fields of a few kV/cm. The permanent dipole moment of KRb has been experimentally determined to be 0.566D [3].

On the scale of the rotational constant, the effect of a dc field on the single-molecule energy spectrum is as in Figure 4.5. It is quadratic for field energies small compared to the rotational energy but becomes linear in stronger fields because the

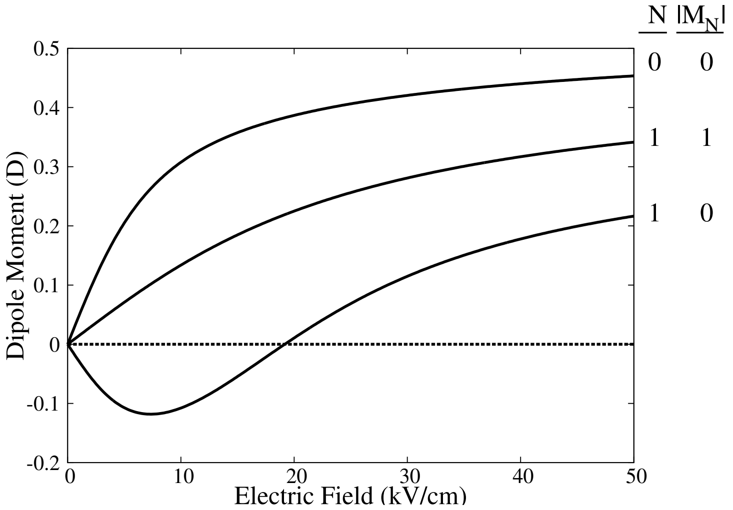


Figure 4.6: *Induced dipoles for KRb in an electric field.* The $N = 0$ and $N = 1$, $M_N = \pm 1$ levels orient along the field, giving rise to positive dipole moments. The $N = 1$, $M_N = 0$ state antialigns with the field for small fields, but aligns in stronger fields. All resonant dipole moments approach the "permanent" value 0.566D as the field strength increases.

field strongly mixes states of opposite parity [36]. We consider the quantization axis to lie along the field direction, and so states with the same value of $|M_N|$ remain degenerate. A universal plot for all $^1\Sigma$ molecules results on this scale if the energy and field strength dE_{dc} are both scaled to the rotational constant.

The average orientation of the molecule with the electric field can be obtained with the Feynman-Hellman theorem as

$$\langle \cos\theta \rangle = -\frac{\partial E}{\partial \left(dE_{\mathrm{dc}} \right)}\,, \tag{4.36}$$

where E is the energy eigenvalue. The energy eigenvalue is dominated by the GHz scale structure, thus the degree of alignment with the field is essentially independent of the hyperfine structure. From the degree of orientation we can also determine the effective space-fixed dipole moment as $d\langle \cos\theta \rangle$. Figure 4.6 shows the behavior of the induced dipoles as the field strength is increased. For all field strengths the $N = 0$ and $N = 1$, $M_N = \pm 1$ states align with the field and so have a positive induced dipole

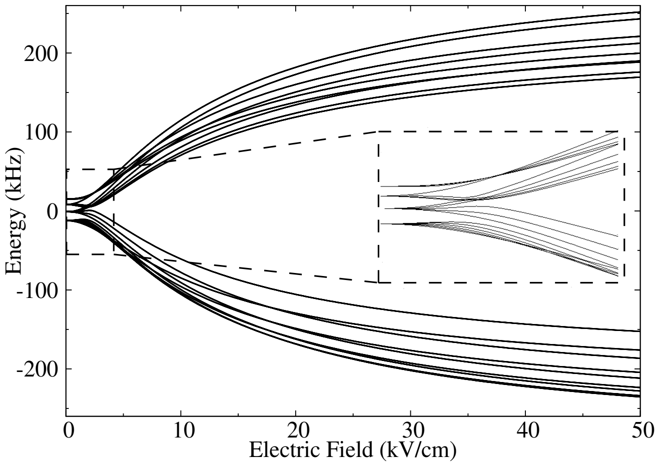


Figure 4.7: *kHz scale view of the Stark effect for* $^{40}K^{87}Rb$, *N=0.* All energies are shown relative to the GHz scale field-dependent average energy for $N = 0$, see Figure 4.5. The inset shows the weak field region where the scalar spin-spin interaction has split the levels according to I (equivalently, F), with larger I having lower energy. As the field is increased the nuclear quadrupole couplings split according to M_I, and in large fields M_{Rb} and M_{K} also become well defined. See text for details.

moment. In contrast, the $N = 1$, $M_N = 0$ state antialigns with the field for weak fields and aligns with the field for stronger fields.

The magnitude of the field energy completely obscures the hyperfine splittings, and so to see the effects of hyperfine structure we subtract from each state with a given N the field-dependent average energy of all hyperfine states with the same N. For $N = 0$ the results are shown in Figure 4.7. For low fields the hyperfine splittings are dominated by the scalar spin-spin coupling and are of order c_4, a few kHz. As the field is increased the various hyperfine states split according to $|M_I|$. For large fields M_1 and M_2 also become well defined, which occurs because the energetic differences between states with $\Delta M_N = \pm 1$ become larger than the quadrupole coupling constants (see Eq. (4.45)). Pairs of M_1 and M_2 which have the same $|M_1 + M_2|$ are degenerate, and the state with $|M_1 + M_2| = 0$ is degenerate due to reflection symmetry in the plane of the electric field vector.

Because of the signs of the quadrupole couplings for ^{40}K^{87}Rb, the lowest energy states are those with M_{Rb} the largest and M_{K} the smallest. Because the kHz scale Stark effect depends on several molecular parameters it cannot be put into a universal form for all $^1\Sigma$ molecules like the GHz scale Stark effect. However, the qualitative structure will be similar for all $^1\Sigma$ molecules with nuclear quadrupole couplings; key differences being the number of nondegenerate levels and the energetic ordering of the magnetic quantum numbers [27]. The hyperfine Stark effect for $N = 1$ and other molecular species as well as the effects of electric fields on microwave spectra may be found in Ref. [10].

Magnetic fields couple to the magnetic moments generated by the rotation of the molecule and by the nuclear spins. The former interaction is given by $-g_r \mu_N \mathbf{N} \cdot \mathbf{B}$, where g_r is the rotational g factor of the molecule and μ_N is the nuclear magneton $e\hbar/2m_p$=762.259 Hz/G [37]. The latter interaction is given by $-\sum_{i=1}^{2} g_i \mu_N \mathbf{I}_i \cdot \mathbf{B}\,(1-\sigma_i)$, where g_1 and g_2 are the g-factors of nucleus 1 and 2, respectively, and σ_i is the isotropic part of the nuclear shielding tensor for nucleus i. The rotational contribution is typically much smaller than the contributions from the nuclei, due to smaller g-factors and the fact that the isotropic parts of the nuclear shielding tensors are typically only a few parts per thousand. For example, in ^{40}K^{87}Rb $g_r = 0.0140$, $g_{\text{K}} = -0.324$, $g_{\text{Rb}} = 1.834$, $\sigma_{\text{K}} = 1321$ ppm, and $\sigma_{\text{Rb}} = 3469$ ppm [27]. We neglect diamagnetic contributions to the Zeeman effect, as these contributions are small for the fields we consider.

Typical experimental magnetic fields are ~ 550G because of the Feshbach association stage of the STIRAP procedure [2]. In Figure 4.8 we show the Zeeman effect for the $N = 0$ level of ^{40}K^{87}Rb for fields up to this range. We see that the magnetic field splits the spectrum according to the nuclear spin projections M_{K} and M_{Rb}, with larger (smaller) M_{K} (M_{Rb}) having lower energy due to the signs of the g factors for ^{40}K^{87}Rb. Because the nuclear quadrupole interaction doesn't affect the $N = 0$

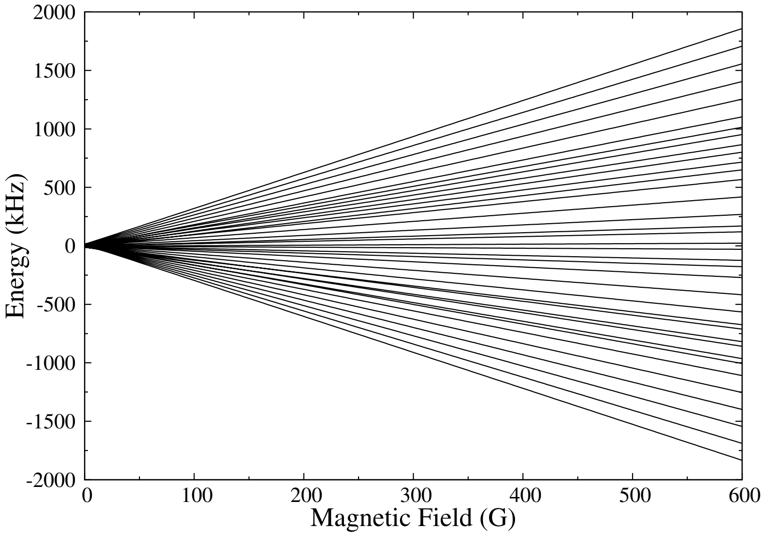


Figure 4.8: *Zeeman effect for* ^{40}*K*^{87}*Rb, N=0.* The magnetic field splits the hyperfine levels according to their projections M_{K} and M_{Rb} with splittings between adjacent levels of order kHz for the experimentally relevant range $B \sim 550$ G. The lowest (highest) energy state corresponds to $m_F = -4 + 3/2 = -5/2$ $(5/2)$. The zero field splitting is set by c_4 and is not visible on the scale of this plot.

level, the zero field splittings are determined by the small scalar spin-spin coupling parameter c_4. The Zeeman term dominates over the scalar spin-spin coupling at very low fields and so the effects of the scalar spin-spin coupling are not discernible on the scale of this plot. Additionally, the Zeeman contribution at these fields is larger than the hyperfine Stark splittings from the largest electric fields accessible in current experiments, see Figure 4.7.

The spectrum for the $N = 1$ level of ^{40}K^{87}Rb is shown in Figure 4.8. It is greatly complicated by the fact that there are three times as many states as the $N = 0$ case (corresponding to the allowed M_N). Also, the nuclear quadrupole interaction affects the $N = 1$ level, causing the large zero field splittings. These larger zero field splittings delay the separation of the levels into well defined M_1 and M_2, and also causes a complicated series of avoided crossings between states with the same M_F.

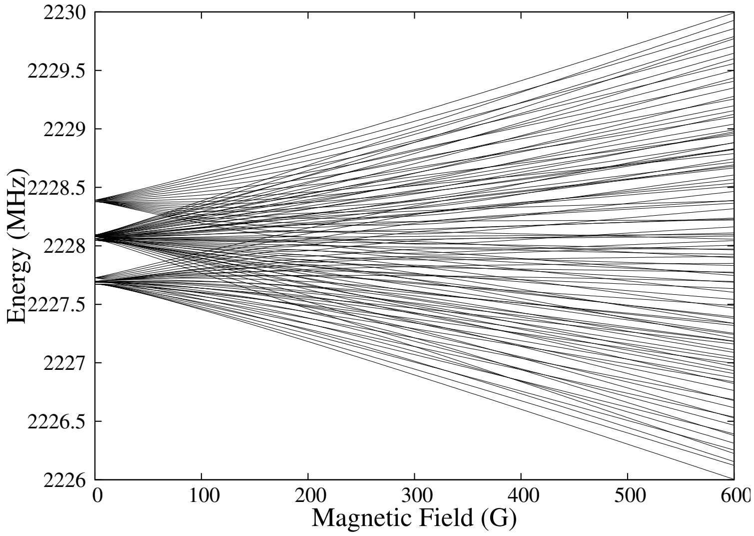


Figure 4.9: *Zeeman effect for* $^{40}K^{87}Rb$, $N=1$. The zero field splitting is caused mainly by the nuclear quadrupole intraction and separates the levels into groups of well defined F. The much larger zero field splitting causes avoided crossings between states with the same M_F to occur at much higher fields than in the $N = 0$ case.

4.7 Explicit Values for the Single-Molecule Matrix Elements

Here we present the matrix elements of the single-molecule terms of the Hamiltonian (4.26) in the coupled and uncoupled basis sets. We adopt the conventions of Zare [35].

The matrix elements of the rotational Hamiltonian are given by

$$\begin{aligned}&\langle I_1 M_1 I_2 M_2 N M_N | B_N \mathbf{N}^2 | I_1 M_1' I_2 M_2' N' M_N' \rangle \\ &= \delta_{M_1,M_1'} \delta_{M_2,M_2'} \delta_{N,N'} \delta_{M_N,M_N'} B_N N \left(N+1\right) ,\end{aligned} \tag{4.37}$$

$$\begin{aligned}&\langle \left(I_1 I_2\right) I N F M_F | B_N \mathbf{N}^2 | \left(I_1 I_2\right) I' N' F' M_F' \rangle \\ &= \delta_{I,I'} \delta_{N,N'} \delta_{F,F'} \delta_{M_F,M_F'} B_N N \left(N+1\right) .\end{aligned} \tag{4.38}$$

The matrix elements of the rotation-spin Hamiltonian are given by

$$
\begin{aligned}
&\langle I_1 M_1 I_2 M_2 N M_N | \sum_{i=1}^{2} c_i \mathbf{N}\cdot\mathbf{I}_i | I_1 M_1' I_2 M_2' N' M_N' \rangle \\
&= \delta_{N,N'} \sum_q (-1)^{q+N-M_N} \begin{pmatrix} N & 1 & N \\ -M_N & q & M_N' \end{pmatrix} \\
&\times \sum_i c_i (-1)^{I_i - M_i} \begin{pmatrix} I_i & 1 & I_i \\ -M_i & -q & M_i' \end{pmatrix} \\
&\times \sqrt{N(2N+1)(N+1) I_i (2I_i+1)(I_i+1)}\,,
\end{aligned}
\tag{4.39}
$$

$$
\begin{aligned}
&\langle (I_1 I_2)\, I N F M_F | \sum_{i=1}^{2} c_i \mathbf{N}\cdot\mathbf{I}_i |\, (I_1 I_2)\, I' N' F' M_F' \rangle \\
&= \delta_{N,N'} (-1)^{I+N+F+I_1+I_2+1} \begin{Bmatrix} I & N & F \\ N & I' & 1 \end{Bmatrix} \\
&\times \sqrt{N(2N+1)(N+1)(I+1)(2I+1)(2I'+1)} \\
&\times \Bigg[\delta_{M_2,M_2'} (-1)^{I'} c_1 \begin{Bmatrix} I_1 & I & I_2 \\ I' & I_1 & 1 \end{Bmatrix} \sqrt{I_1(2I_1+1)(I_1+1)} \\
&+ \delta_{M_1,M_1'} (-1)^{I} c_2 \begin{Bmatrix} I_2 & I & I_1 \\ I' & I_2 & 1 \end{Bmatrix} \sqrt{I_2(2I_2+1)(I_2+1)} \Bigg].
\end{aligned}
\tag{4.40}
$$

The matrix elements of the scalar spin-spin coupling are

$$
\begin{aligned}
&\langle I_1 M_1 I_2 M_2 N M_N | c_4 \mathbf{I}_1\cdot\mathbf{I}_2 | I_1 M_1' I_2 M_2' N' M_N' \rangle \\
&= \delta_{N,N'} \delta_{F,F'} \delta_{M_F,M_F'} c_4 (-1)^{I_1 - M_1 + I_2 - M_2} \\
&\times \sqrt{(2I_1+1) I_1 (I_1+1)(2I_2+1) I_2 (I_2+1)} \\
&\times \sum_q (-1)^q \begin{pmatrix} I_1 & 1 & I_1 \\ -M_1 & q & M_1' \end{pmatrix} \begin{pmatrix} I_2 & 1 & I_2 \\ -M_2 & -q & M_2' \end{pmatrix},
\end{aligned}
\tag{4.41}
$$

$$
\begin{aligned}
&\langle (I_1 I_2)\, I N F M_F | c_4 \mathbf{I}_1\cdot\mathbf{I}_2 |\, (I_1 I_2)\, I' N' F' M_F' \rangle \\
&= \delta_{I,I'} \delta_{N,N'} \delta_{F,F'} \delta_{M_F,M_F'} \\
&\times \frac{c_4}{2} \left[I(I+1) - I_1(I_1+1) - I_2(I_2+1) \right].
\end{aligned}
\tag{4.42}
$$

The matrix elements of the tensor spin-spin coupling are

$$\begin{aligned}
&\langle I_1 M_1 I_2 M_2 N M_N | c_3 \mathbf{I}_1 \cdot \tilde{T} \cdot \mathbf{I}_2 | I_1 M_1' I_2 M_2' N' M_N' \rangle \\
&= -c_3 \sqrt{6} \begin{pmatrix} N & 2 & N' \\ 0 & 0 & 0 \end{pmatrix} \sqrt{(2N+1)(2N'+1)} \\
&\times \sqrt{I_1 I_2 (2I_1+1)(2I_2+1)(I_1+1)(I_2+1)} \\
&\times \sum_q (-1)^{q-M_N+I_1-M_1+I_2-M_2} \begin{pmatrix} N & 2 & N' \\ -M_N & q & M_N' \end{pmatrix} \\
&\times \sum_m \langle 1, m; 1, -q-m | 2, -q \rangle \begin{pmatrix} I_1 & 1 & I_1 \\ -M_1 & m & M_1' \end{pmatrix} \\
&\times \begin{pmatrix} I_2 & 1 & I_2 \\ -M_2 & -q-m & M_2' \end{pmatrix},
\end{aligned} \tag{4.43}$$

$$\begin{aligned}
&\langle (I_1 I_2) I N F M_F | c_3 \mathbf{I}_1 \cdot \tilde{T} \cdot \mathbf{I}_2 | (I_1 I_2) I' N' F' M_F' \rangle \\
&= -c_3 \delta_{F,F'} \delta_{M_F,M_F'} (-1)^{I'+F} \begin{Bmatrix} I & N & F \\ N' & I' & 2 \end{Bmatrix} \\
&\times \sqrt{(2N+1)(2N'+1)} \begin{pmatrix} N & 2 & N' \\ 0 & 0 & 0 \end{pmatrix} \begin{Bmatrix} I_1 & I_1 & 1 \\ I_2 & I_2 & 1 \\ I & I' & 2 \end{Bmatrix} \\
&\times \sqrt{30 (2I+1)(2I'+1) I_1 I_2} \\
&\times \sqrt{(I_1+1)(I_2+1)(2I_1+1)(2I_2+1)}.
\end{aligned} \tag{4.44}$$

The matrix elements of the nuclear quadrupole Hamiltonian are given by

$$\begin{aligned}
&\langle I_1 M_1 I_2 M_2 N M_N | \sum_{i=1}^{2} \mathbf{V}_i \cdot \mathbf{Q}_i | I_1 M_1' I_2 M_2' N' M_N' \rangle \\
&= \sum_{i=1}^{2} \frac{(eqQ)_i}{4} \sum_q (-1)^{q-M_N+I_i-M_i} \sqrt{(2N+1)(2N'+1)} \\
&\times \begin{pmatrix} N & 2 & N' \\ -M_N & q & M_N' \end{pmatrix} \begin{pmatrix} I_i & 2 & I_i \\ -M_i & -q & M_i' \end{pmatrix} \\
&\times \begin{pmatrix} N & 2 & N' \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} I_i & 2 & I_i \\ -I_i & 0 & I_i \end{pmatrix}^{-1},
\end{aligned} \tag{4.45}$$

$$
\begin{aligned}
&\langle (I_1 I_2)\, I N F M_F | \sum_{i=1}^{2} \mathbf{V}_i \cdot \mathbf{Q}_i | \, (I_1 I_2)\, I' N' F' M_F' \rangle \\
&= \delta_{F,F'} \delta_{M_F,M_F'} \frac{1}{4} \left(-1\right)^{I'+F+I_1+I_2} \\
&\times \sqrt{(2N+1)\,(2N'+1)\,(2I'+1)\,(2I+1)} \\
&\times \begin{pmatrix} N & 2 & N' \\ 0 & 0 & 0 \end{pmatrix} \begin{Bmatrix} I & N & F \\ N' & I' & 2 \end{Bmatrix} \\
&\times \Bigg[\delta_{I_2,I_2'} \left(eqQ\right)_1 \left(-1\right)^{I'} \begin{Bmatrix} I_1 & I & I_2 \\ I' & I_1 & 2 \end{Bmatrix} \begin{pmatrix} I_1 & 2 & I_1 \\ -I_1 & 0 & I_1 \end{pmatrix}^{-1} \\
&+ \delta_{I_1,I_1'} \left(eqQ\right)_2 \left(-1\right)^{I} \begin{Bmatrix} I_2 & I & I_1 \\ I' & I_2 & 2 \end{Bmatrix} \begin{pmatrix} I_2 & 2 & I_2 \\ -I_2 & 0 & I_2 \end{pmatrix}^{-1} \Bigg] .
\end{aligned}
\tag{4.46}
$$

4.8 References Cited

[1] L. D. Carr, D. Demille, R. V. Krems, and Jun Ye. Cold and Ultracold Molecules: Science, Technology, and Applications. *New J. Phys.*, 11:055049, 2009.

[2] K.-K. Ni, S. Ospelkaus, M. H. G. de Miranda, A. Peér, B. Neyenhuis, J. J. Zirbel, S. Kotochigova, P. S. Julienne, D. S. Jin, and Jun Ye. A High Phase-Space-Density Gas of Polar Molecules. *Science*, 322:231–235, 2008.

[3] S. Ospelkaus, K. K. Ni, M. H. G. de Miranda, A. Pe'er, B. Nyenhuis, D. Wang, S. Kotochigova, P. S. Julienne, D. S. Jin, , and Jun Ye. Ultracold polar molecules near quantum degeneracy. *Faraday Discuss.*, 142:351–359, 2009.

[4] J. Deiglmayr, A. Grochola, M. Repp, K. Mörtlbauer, C. Glück, J. Lange, O. Dulieu, R. Wester, and M. Weidemüller. Formation of Ultracold Polar Molecules in the Rovibrational Ground State. *Phys. Rev. Lett.*, 101(13):133004, Sep 2008. doi: 10.1103/PhysRevLett.101.133004.

[5] J. G. Danzl, M. J. Mark, E. Haller, M. Gustavsson, R. Hart, J. Aldegunde, J. M. Hutson, and H.-C Nägerl. An ultracold high-density sample of rovibronic ground-state molecules in an optical lattice. *Nature Phys.*, 6:265–270, 2010.

[6] K. Winkler, F. Lang, G. Thalhammer, P. v. d. Straten, R. Grimm, and J. Hecker Denschlag. Coherent Optical Transfer of Feshbach Molecules to a Lower Vibrational State. *Phys. Rev. Lett.*, 98(4):043201, Jan 2007. doi: 10.1103/PhysRevLett.98.043201.

[7] K. Pilch, A. D. Lange, A. Prantner, G. Kerner, F. Ferlaino, H.-C. Nägerl, and R. Grimm. Observation of interspecies Feshbach resonances in an ultracold Rb-Cs mixture. *Phys. Rev. A*, 79(4):042718, Apr 2009. doi: 10.1103/PhysRevA.79.042718.

[8] A.-C. Voigt, M. Taglieber, L. Costa, T. Aoki, W. Wieser, T. W. Hänsch, and K. Dieckmann. Ultracold Heteronuclear Fermi-Fermi Molecules. *Phys. Rev. Lett.*, 102(2):020405, Jan 2009. doi: 10.1103/PhysRevLett.102.020405.

[9] J. Aldegunde, Hong Ran, and Jeremy M. Hutson. Manipulating ultracold polar molecules with microwave radiation: the influence of hyperfine structure. *Phys. Rev. A*, 80:043410, 2009.

[10] Hong Ran, J Aldegunde, and Jeremy M Hutson. Hyperfine structure in the microwave spectra of ultracold polar molecules. *New Journal of Physics*, 12(4): 043015, 2010. URL `http://stacks.iop.org/1367-2630/12/i=4/a=043015`.

[11] S. Ospelkaus, K. K. Ni, G. Quemener, B. Neyenhuis, D. Wang, M. H. G. de Miranda, J. L. Bohn, J. Ye, and D. S. Jin. Controlling the hyperfine state of rovibronic ground-state polar molecules. *Phys. Rev. Lett.*, 104:030402, 2010.

[12] R. P. Feynman. Simulating Physics with Computers. *Int. J. Theor. Phys.*, 21: 467, 1982.

[13] M. Sanpera A. Lewenstein, V. Ahufinger, B. Damski, A. Sen De, and U. Sen. Ultracold atomic gases in optical lattices: Mimicking condensed matter physics and beyond. *Adv. Phys.*, 56:243–379, 2007.

[14] A. Micheli, G. Pupillo, H. P. Büchler, and P. Zoller. Cold polar molecules in two-dimensional traps: Tailoring interactions with external fields for novel quantum phases. *Phys. Rev. A*, 76:043604, 2007.

[15] M. L. Wall and L. D. Carr. Emergent timescales in entangled quantum dynamics of ultracold molecules in optical lattices. *New J. Phys.*, 11:055027, 2009.

[16] Gavin K Brennen, Andrea Micheli, and Peter Zoller. Designing spin-1 lattice models using polar molecules. *New J. Phys.*, 9:138, 2007.

[17] Sebastian D. Huber and Ehud Altman. Universal Dephasing of Many-Body Rabi Oscillations of Atoms in One-Dimensional Traps. *Phys. Rev. Lett.*, 103:160402, 2009.

[18] Peter Barmettler, Matthias Punk, Vladimir Gritsev, Eugene Demler, and Ehud Altman. Quantum quenches in the anisotropic spin-1/2 heisenberg chain: different approaches to many-body dynamics far from equilibrium. *New Journal of Physics*, 12(5):055017, 2010. URL `http://stacks.iop.org/1367-2630/12/i=5/a=055017`.

[19] Neil Ashcroft and David Mermin. *Solid State Physics.* Saunders College Publishing, Orlando, 1976.

[20] J. Deiglmayr, M. Aymar, R. Wester, M. Weidemüller, and O. Dulieu. Calculations of static dipole polarizabilities of alkali dimers: Prospects for alignment of ultracold molecules. *J. Chem. Phys.*, 129:064309, 2008.

[21] M. Iskin and C. A. R. Sa de Melo. Mixtures of ultracold fermions with unequal masses. *Phys. Rev. A.*, 76:013601, 2007.

[22] M. L. Wall and L. D. Carr, 2010. In preparation.

[23] B. Paredes, A. Widera, V. Murg, O. Mandel, S. Folling, I. Cirac, G. V. Shlyapnikov, and T. W. Hansch. Tonks-Girardeau gas of ultracold atoms in an optical lattice. *Nature*, 429:277–281, 2005.

[24] B. L. Tolra, K. M. O'Hara, J. H. Huckans, W. D. Phillips, S. L. Rolston, and J. V. Porto. Observation of reduced three-body recombination in a correlated 1D degenerate bose gas. *Phys. Rev. Lett.*, 92:190401–1–4, 2004.

[25] T. Wenger T. Kinoshita and D. S. Weiss. A quantum Newton's cradle. *Nature*, 440:900–903, 2006.

[26] H. P. Büchler, A Micheli, and P Zoller. Three-body interactions with cold polar molecules. *Nature Phys.*, 3:726–731, 2007.

[27] J. Aldegunde, Ben A. Rivington, Piotr S. Żuchowski, and Jeremy M. Hutson. Hyperfine energy levels of alkali-metal dimers: Ground-state polar molecules in electric and magnetic fields. *Phys. Rev. A*, 78:033434, 2008.

[28] Ana Maria Rey. *Ultracold Bosonic Atoms in Optical Lattices.* PhD thesis, University of Maryland, 2004.

[29] K. K. Ni, S. Ospelkaus, D. Wang, G. Quemener, B. Neyenhuis, M. H. G. de Miranda, J. L. Bohn, J. Ye, and D. S. Jin. Dipolar collisions of polar molecules in the quantum regime. *Nature (London)*, 464:1324, 2010.

[30] S. Chakravarty. Quantum phase transition, dissipation, and measurement. *e-print http://arxiv.org/abs/0909.2316*, 2010.

[31] K. Le Hur. Quantum phase transitions in spin-boson systems: Dissipation and light phenomena. *e-print http://arxiv.org/abs/0909.4822*, 2010.

[32] P. Werner and M. Troyer. Cluster monte carlo algorithms for dissipative quantum phase transitions. *Chapter of the book 'Understanding Quantum Phase Transitions', edited by Lincoln D. Carr (CRC Press / Taylor and Francis, 2010)*, 2010.

[33] John Brown and Alan Carrington. *Rotational Spectroscopy of Diatomic Molecules*. Cambridge University Press, Cambridge, 2003.

[34] J. H. Van Vleck. The coupling of angular momentum vectors in molecules. *Rev. Mod. Phys.*, 23:213–227, 1951.

[35] Richard Zare. *Angular Momentum: Understanding Spatial Aspects in Chemistry and Physics*. Wiley, New York, 1988.

[36] R. N. Townes and A. L. Schawlow. *Microwave Spectroscopy*. Dover, New York, 1975.

[37] Peter J. Mohr and Barry N. Taylor. CODATA recommended values of the fundamental physical constants: 1998. *Rev. Mod. Phys.*, 72:351, 2000.

PART III

THE FERMI RESONANCE HAMILTONIAN

CHAPTER 5

MICROSCOPIC MODEL FOR FESHBACH INTERACTING FERMIONS IN AN OPTICAL LATTICE WITH ARBITRARY SCATTERING LENGTH AND RESONANCE WIDTH

Abstract:[27] We numerically study the problem of two fermions in a three dimensional optical lattice interacting via a zero-range Feshbach resonance, and display the dispersions of the bound states as a two-particle band structure with unique features compared to typical single-particle band structures. We show that the exact two-particle solutions of a projected Hamiltonian may be used to define an effective two-channel, few-band model for the low energy, low density physics of many fermions at arbitrary s-wave scattering length. Our method applies to resonances of any width, and can be adapted to multichannel situations or higher-ℓ pairing. In strong contrast to usual Hubbard physics, we find that pair hopping is significantly altered by strong interactions and the presence of the lattice, and the lattice induces multiple molecular bound states.

The crossover of a system of attractive two-component fermions from a condensate of loosely bound Cooper pairs to a condensate of tightly bound bosonic molecules has a long history [1], and appears in many contexts, including high-temperature superconductivity [2] and ultracold atoms [3]. Furthermore, near the crossover such a system enters the *unitary regime* where the scattering length is larger than any other length scale in the problem. The physics of this regime is relevant to many different fields, bringing together quantum chromodynamics, holographic duality, and ultracold quantum gases [4]. Theoretical study of the unitary regime is generally difficult due

---

[27] Published previously as *Microscopic Model for Feshbach Interacting Fermions in an Optical Lattice with Arbitrary Scattering Length and Resonance Width*, M. L. Wall and L. D. Carr, Phys. Rev. Lett. **109**, 055302 (2012).

to the absence of any small parameter.

Theoretical analysis becomes even more difficult in a lattice, as the center of mass, relative, and internal degrees of freedom become coupled, leading to composite particles whose properties depend on their center of mass motion [5]. Furthermore, strong interactions require the inclusion of a large number of Bloch bands for an accurate description, and this cannot be handled efficiently by modern analytical or numerical many-body techniques. In addition to general theoretical interest in how fermions pair to form bosons in a discrete lattice setting, the study of pairing in lattices is of significant practical importance. For example, an accurate, systematically correctable, and computationally feasible many-body Hamiltonian is necessary for calibrating fermionic quantum simulators as has been done in the bosonic case [6].

In this Letter, we describe a general method to derive an effective few-band low-energy Hamiltonian for Feshbach interacting fermions in a lattice from the numerical solution of the two-body problem. We call this Hamiltonian the *Fermi Resonance Hamiltonian* (FRH). This method applies to Feshbach resonances of any width and for arbitrary scattering length, and all parameters appearing in the effective model can be computed microscopically from the properties of the two-body solution. The difference between the bare model and the FRH is sketched in Figure 5.1.

The simplest approach to describing Feshbach interacting fermions is to replace the interaction with a pseudo-potential chosen to reproduce the correct scattering length. When restricted to a single Bloch band, this leads to the popular Hubbard model [7] which has been shown to break down for scattering lengths which are far from being resonant, even when the parameters appearing in the model are determined self-consistently from few-body physics [8]. Our work instead defines a “dressed” closed channel whose properties are chosen to reproduce both the scattering and bound states correctly. In contrast to past two-channel approaches [9], we construct the dressed fields using the full lattice solution and not an approximation where

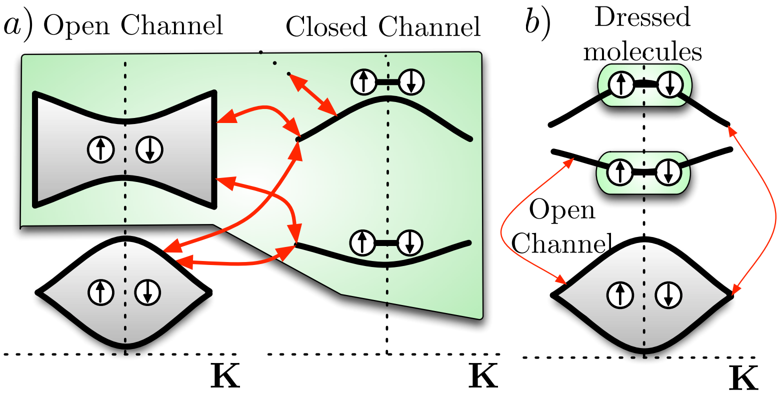


Figure 5.1: (Color online) *Schematic of the FRH transformation.* (a) In a broad Feshbach resonance, all two-particle scattering continua (gray shading) are strongly coupled to bare molecular bands (solid lines). Thus all scattering continua are virtually strongly coupled. (b) By correctly dressing the molecular bands, one obtains a single scattering continuum (gray) plus well-separated dressed molecular bands (green), with much simpler couplings. This is our efficient, numerically tractable, FRH.

the center of mass and relative coordinates separate, such as the harmonic oscillator potential. The use of any separable approximation leads to qualitative errors, such as the lack of tunneling along non-principal axes, and quantitative errors, such as the underestimation of principal axis tunneling matrix elements, often by an order of magnitude.

5.1 Exact Solution for Two Particles.

The basic concept of a two-channel model is for an *open* channel to describe scattering between two atoms and a separate *closed* channel to describe bound pairs. While each channel represents a single scattering or bound state in the continuum, in the lattice it also acquires a band index. Because of an inter-channel coupling, the actual molecule is a superposition of bands from both channels.

To treat this problem, we begin with the nonlinear eigenvalue problem developed by Büchler [8] for $E_{\mathbf{K}}$, the bound state energy at total quasimomentum $\mathbf{K}$, and $R^{\mathbf{K}}_{\mathbf{s}}$, the coefficients of the closed channel portion of the wavefunction. As shown in [8], for the bound states of two fermions in an optical lattice interacting via a zero-range Feshbach resonance in a two-channel model:

$$\left[E_{\mathbf{K}} - \nu - E^{M}_{\mathbf{sK}}\right] R^{\mathbf{K}}_{\mathbf{s}} = \frac{g^2}{a^3} \sum_{\mathbf{t}} \chi^{\mathbf{K}}_{\mathbf{st}}\left(E_{\mathbf{K}}\right) R^{\mathbf{K}}_{\mathbf{t}}\,, \tag{5.1}$$

$$\chi^{\mathbf{K}}_{\mathbf{st}}\left(E_{\mathbf{K}}\right) \equiv \int \frac{d\mathbf{q}}{v_0} \sum_{\mathbf{mn}} \frac{h^{\mathbf{nm}}_{\mathbf{sK}}(\mathbf{q}) h^{\mathbf{nm}\star}_{\mathbf{tK}}(\mathbf{q})}{E_{\mathbf{K}} - E^{\mathbf{K}}_{\mathbf{nm}}(\mathbf{q}) + i\eta} - \bar{\chi}^{\mathbf{K}}_{\mathbf{st}}\,, \tag{5.2}$$

$$\bar{\chi}^{\mathbf{K}}_{\mathbf{st}} \equiv -\int \frac{d\mathbf{q}}{v_0} \sum_{\mathbf{mn}} \bar{h}^{\mathbf{nm}}_{\mathbf{sK}}\left(\mathbf{q}\right) \bar{h}^{\mathbf{nm}\star}_{\mathbf{tK}}\left(\mathbf{q}\right) / \bar{E}^{\mathbf{K}}_{\mathbf{nm}}\left(\mathbf{q}\right)\,. \tag{5.3}$$

Here ν is the renormalized detuning between the open and closed channels, g is the inter-channel coupling, a is the lattice spacing, v_0 is the volume of the Brillouin zone (BZ), and the bars in Eqs. (5.2-5.3) denote quantities computed in the absence of an optical lattice. We assume that spin-spin interactions which change the orbital angular momentum are negligible so that the scattering is purely s-wave. The optical lattice is assumed to be simple cubic with lattice spacing a and potential $V\left(\mathbf{x}\right) = V\sum_{j\in\{x,y,z\}} \sin^2\left(\pi j/a\right)$. The overlaps of the dimensionless coupling between the open and closed channels are

$$\frac{h^{\mathbf{nm}}_{\mathbf{sK}}\left(\mathbf{q}\right)}{\sqrt{N^3 a^3}} = \int d\mathbf{x} d\mathbf{y} \left[\psi_{\mathbf{nq}}\left(\mathbf{x}\right) \psi_{\mathbf{m},\mathbf{K}-\mathbf{q}}\left(\mathbf{y}\right)\right]^{\star} \alpha\left(\mathbf{r}\right) \phi_{\mathbf{sK}}\left(\mathbf{R}\right)\,,$$

where N^3 is the number of unit cells, the $\psi_{\mathbf{nq}}\left(\mathbf{x}\right)$ are Bloch functions with energies $E_{\mathbf{nq}}$ for particles with mass m spanning the open channel and $\phi_{\mathbf{sK}}\left(\mathbf{z}\right)$ are Bloch functions with energies $E^{M}_{\mathbf{sK}}$ for particles with mass $2m$ in a lattice potential $2V$ spanning the closed channel. We have also defined relative $\mathbf{r} \equiv \mathbf{x} - \mathbf{y}$ and center of mass $2\mathbf{R} = \mathbf{x} + \mathbf{y}$ coordinates, and $\alpha\left(\mathbf{r}\right)$ is a regularization of the inter-channel coupling. The sum of the noninteracting energies of the open channel is denoted $E^{\mathbf{K}}_{\mathbf{nm}}\left(\mathbf{q}\right) = E_{\mathbf{nq}} + E_{\mathbf{m},\mathbf{K}-\mathbf{q}}$ and the zero of energy is $E^{\mathbf{0}}_{\mathbf{11}}\left(\mathbf{0}\right)$. Here and throughout the rest of this work $\mathbf{n}$ and $\mathbf{m}$ are band indices for the open channel, $\mathbf{s}$ and $\mathbf{t}$ are band indices for the closed channel, $\mathbf{q}$ is a single-particle quasimomentum, and $\mathbf{K}$ is the total quasimomentum.

While Eqs. (5.1)-(5.3) apply to resonances of any width, we focus on the experimentally relevant limit of a broad resonance. Narrow resonances are treated in the

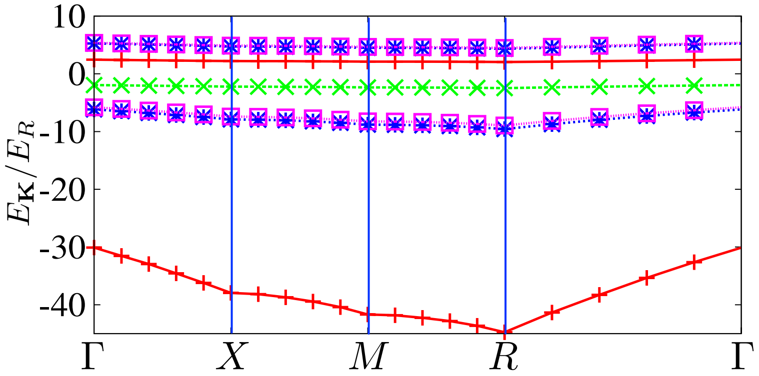


Figure 5.2: (Color online) *Exact two-particle band structures for various a_s in a strong optical lattice.* The bound state energies for $a_s/a = -5$ (purple boxes), -0.1 (red pluses), 0.1 (green crosses), and 5 (blue asterisks) as a function of the total quasimomentum $\mathbf{K}$ along a path connecting the high-symmetry points in the irreducible BZ $\Gamma = (0,0,0)$, $X = (-\pi/a, 0, 0)$, $M = (-\pi/a, -\pi/a, 0)$, $R = (-\pi/a, -\pi/a, -\pi/a)$ for a lattice with $V/E_R = 12$. The near-resonant points $a_s/a = \pm 5$ lie nearly on top of one another, demonstrating universality.

supplementary material. A broad resonance in the few-body sense is the limit of effective range much smaller than the lattice spacing, $r_B \ll a$, and so we can take the limits $g/E_R a^{3/2} = 4\sqrt{a/r_B\pi^3} \to \infty$, $\nu/E_R \to \infty$, $a_s/a = -\pi g^2/8a^3 E_R \nu = \text{const.}$, to obtain $(8a_s E_R/\pi a)\chi^{\mathbf{K}}(E_{\mathbf{K}})\,\mathbf{R}^{\mathbf{K}} - \mathbf{R}^{\mathbf{K}} = 0$, where $E_R = \hbar^2\pi^2/2ma^2$ is the recoil energy and a_s is the s-wave scattering length. How can we then obtain the dispersion relation $E_{\mathbf{K}}$ for fixed a_s, ν, etc.? First, fix the energy eigenvalue $E_{\mathbf{K}}$ and solve the resulting *linear* eigenproblem for $1/a_s$. This provides exact eigentuples $\left(E_{\mathbf{K}}, a_s, \mathbf{R}^{\mathbf{K}}\right)$ of the nonlinear eigenproblem, though it may not be the a_s we seek. Second, fix a_s and use the exact tuple nearest this value as initialization for a Newton-Raphson iteration [10]. This two-stage approach converges to a relative accuracy of 0.01% in a few tens of iterations [11].

Because the eigenequation Eq. (5.1) is invariant under translation by any Bravais lattice vector, its eigenvalues can be classified according to the total quasimomentum and shown as a two-particle band structure. A complete classification of the solutions is given in the supplemental material. In Figure 5.2 we show only the energy of the low-energy bound states with completely even parity under inversion as a function of $\mathbf{K}$ for several a_s/a in a lattice with $V/E_R = 12$. We see the appearance of several bound states for a fixed s-wave scattering length, in contrast to the continuum. These additional bound states arise from the coupling of quasimomenta modulo a reciprocal lattice vector induced by the reduced translational symmetry. One salient feature is the emergence of universality, which is the independence of the dispersion from the sign of a_s when $|a_s/a|$ becomes large. For non-resonant and negative a_s/a, picturing the bound states as Fermi pairs with twice the mass and twice the polarizability captures the relative spacings between energy levels quite accurately, but generally *overestimates* the effective mass of the bound states. This effective mass difference is an indication of the coupling between the center of mass and relative motion which leads to important properties of the FRH.

5.2 Fermi Resonance Hamiltonian.

A promising route to describing Feshbach interacting ultracold gases is by a lattice projection of a two-channel model in which the closed channel appears explicitly in the Hamiltonian. However, for a typical broad resonance such models require a large number of both open and closed channel bands to solve accurately, and so cannot be treated efficiently. Because the modern context of this problem involves extremely low temperatures and densities, we can look for an effective model valid in these limits which still reproduces the correct physics. This is done by replacing the model containing couplings between all open channel bands with all closed channel bands with a model describing an effective resonance between the lowest open channel band with a suitable set of effective closed channel bands whose properties are set by the two-body solution for low densities. This process is displayed schematically in Figure 5.1. The purpose of this section is to derive such an effective Hamiltonian using our two-particle theory.

We begin by separating our two-particle Hamiltonian using projectors $\mathbb{L}$ into the lowest open channel band and $\mathbb{D} = 1 - \mathbb{L}$ into all excited open channel bands and all closed channel bands. A similar approach was taken in Ref. [12] for the 1D case. An analysis analogous to that leading to Eqs. (5.1)-(5.2) gives a nonlinear eigenequation for the closed channel components of $\mathbb{D}|\psi\rangle$ as

$$\left[E_{\mathbf{K}} - \nu - E^{M}_{\mathbf{sK}}\right] R^{\mathbf{K}}_{\mathbf{s}} = \tfrac{g^2}{a^3} \textstyle\sum_{\mathbf{t}} \tilde{\chi}^{\mathbf{K}}_{\mathbf{st}}\left(E_{\mathbf{K}}\right) R^{\mathbf{K}}_{\mathbf{t}}\,, \tag{5.4}$$

$$\tilde{\chi}^{\mathbf{K}}_{\mathbf{st}}\left(E_{\mathbf{K}}\right) \equiv \textstyle\sum'_{\mathbf{mn};\mathbf{q}} \tfrac{h^{\mathbf{nm}}_{\mathbf{sK}}(\mathbf{q}) h^{\mathbf{nm}\star}_{\mathbf{tK}}(\mathbf{q})}{E_{\mathbf{K}} - E^{\mathbf{K}}_{\mathbf{nm}}(\mathbf{q}) + i\eta} - \bar{\chi}^{\mathbf{K}}_{\mathbf{st}}\,, \tag{5.5}$$

where the prime on the sum indicates $(\mathbf{m}, \mathbf{n}) \neq (\mathbf{1}, \mathbf{1})$. Here $\tilde{\chi}$ differs from χ in Eqs. (5.1)-(5.2) in that the summation excludes the lowest band. We emphasize that the renormalization $\bar{\chi}$ includes all bands and so the detuning and scattering length used in this projected model are those of the full (non-projected) and properly renormalized two-body problem. We call the eigenstates of this projected system *dressed molecules*. Here we label distinct eigenstates of Eq. (5.4) by the parameter α. These solutions share many features of the full solution presented above. However, the divergence of the s-wave scattering length for the lowest energy completely even parity state occurs near $E_{\mathbf{K}} = 0$, indicating that scattering resonances in the lowest open channel band are generated by coupling to this state.

We now assume that, at low temperatures and low densities, two particles which are separated by a distance large compared to the effective range of the potential will remain in the lowest band to minimize their energy. When two particles come together they interact strongly and populate many of the excited open channel bands as well as the closed channel bands. Because it is rare for more than two particles to come together, the particular populations of the excited states are fixed by the two-particle solution. The dressed molecules encapsulate the short distance, high energy physics and couple it to the long wavelength, low energy physics of the lowest band fermions through the Feshbach coupling. The point of connection between the few-and many-body physics is the two-particle scattering length (equivalently g and ν for narrow resonances), which appears as a parameter in the equation Eq. (5.4) defining the dressed molecules.

The FRH is

$$\begin{aligned}\hat{H}_{\text{eff}} = &-t_f \textstyle\sum_{\sigma\in\{\uparrow,\downarrow\}} \sum_{\langle i,j\rangle} \hat{a}^{\dagger}_{i\sigma}\hat{a}_{j\sigma} + E_0 \sum_{\sigma\in\{\uparrow,\downarrow\}} \sum_i \hat{n}^{(f)}_{i\sigma} \\ &- \textstyle\sum_{\alpha\in\mathcal{M}} \sum_{i,j} t^{\alpha}_{i,j} \hat{d}^{\dagger}_{i,\alpha}\hat{d}_{j,\alpha} + \sum_{\alpha\in\mathcal{M}} \bar{\nu}_{\alpha} \sum_i \hat{n}^{(b)}_{i\alpha} \\ &+ \textstyle\sum_{\alpha\in\mathcal{M}} \sum_{ijk} g^{\alpha}_{i-k,k-j} \left[\hat{d}^{\dagger}_{i,\alpha}\hat{a}_{j,\uparrow}\hat{a}_{k,\downarrow} + \text{h.c.}\right] ,\end{aligned} \tag{5.6}$$

where $\hat{a}^{\dagger}_{i\sigma}$ creates a particle with spin σ in the lowest open channel band Wannier state centered at lattice site i, $w_i(\mathbf{x})$; $\hat{d}^{\dagger}_{i,\alpha}$ creates a particle in the α^{th} dressed molecule Wannier state centered at site i, $\mathcal{W}_{i,\alpha}(\mathbf{x},\mathbf{y})$; $\hat{n}^{(f)}_{i\sigma}$ is the number operator for fermions in the lowest Bloch band; and $\hat{n}^{(b)}_{i\alpha}$ is the number operator for the α^{th} dressed molecule state. The set of dressed molecules $\mathcal{M}$ which are included dynamically can be determined on energetic and symmetry grounds from the two-particle solution. At low energies, only the completely even parity dressed molecule in the lowest sheet is relevant to the set $\mathcal{M}$, as all others either have vanishing on-site couplings from parity considerations or are very far off-resonance. In order, the terms in Eq. (5.6) represent tunneling of atoms in the lowest Bloch band between neighboring lattice sites i and j; the energy $E_0 = \sum_{\mathbf{q}} E_{\mathbf{1},\mathbf{q}}/N^3$ of a fermion in the lowest band with respect to the zero of energy; tunneling of the dressed molecular center of mass between two lattice sites i and j, not necessarily nearest neighbors; detunings of the dressed molecules from the lowest band two-particle scattering continuum; and resonant coupling between the lowest band fermions at sites j and k in different internal states and a dressed molecule at site i. The FRH is a two-channel resonance model, between unpaired fermions in the lowest band, and dressed molecules nearby in energy.

We now describe how to calculate the Hubbard parameters appearing in Eq. (5.6). The first term is well-known from single-band Hubbard models [13] and we do not discuss it here. Due to the fact that the solutions of the projected nonlinear eigenequation Eq. (5.4) are also eigenstates of the total quasimomentum, the second and third terms may be written as $\bar{\nu}_\alpha = \int d\mathbf{K} E_{\mathbf{K}}^{\alpha}/v_0$ and $t_{i,j}^{\alpha} = -\int d\mathbf{K} e^{i\mathbf{K}\cdot(\mathbf{R}_i - \mathbf{R}_j)} E_{\mathbf{K}}^{\alpha}/v_0$. Because the band structure is not separable, $E_{\mathbf{K}} \neq \sum_{i=\{x,y,z\}} E_{K_i}$, dressed molecules can tunnel along directions which are not the principal axes of the lattice. This is in stark contrast to single-particle tunneling in Bravais lattices which always occurs along the principal axes. Thus *diagonal hopping* is a key feature neglected in previous approaches. In Figure 5.3(a) we show that diagonal hopping is often of the same order of magnitude as the tunneling of open channel fermions in the lowest band. The signs and magnitudes of the tunnelings and particularly the dressed-molecule atom couplings are crucially affected by the parities of the dressed molecular Wannier functions. We stress that only a full lattice solution can reproduce these important properties of the Hubbard parameters; the frequently used harmonic oscillator approximation will fail even qualitatively to do so.

The remaining Hubbard parameter is the dressed molecule-atom coupling, which becomes in the limit of a broad resonance $g/E_R a^{3/2} \to \infty$

$$g_{i-k,k-j}^{\alpha} = \sum_{\mathbf{s}} \int \tfrac{d\mathbf{K}}{v_0} R_{\alpha\mathbf{s}}^{\mathbf{K}} g_{\alpha\mathbf{K}} \int \tfrac{d\mathbf{q}}{v_0} e^{i(\mathbf{K}\cdot\mathbf{R}_{ik} + \mathbf{q}\cdot\mathbf{R}_{kj})} h_{\mathbf{sK}}^{\mathbf{11}}(\mathbf{q}) \tag{5.7}$$

where the renormalized coupling is $g_{\alpha\mathbf{K}} = E_R/\sqrt{-\mathbf{R}_{\alpha}^{\mathbf{K}} \cdot (\partial\tilde{\chi}_{\mathbf{K}}/\partial E_{\mathbf{K}}^{\alpha}) \cdot \mathbf{R}_{\alpha}^{\mathbf{K}}}$ and $\mathbf{R}_{ij} = \mathbf{R}_i - \mathbf{R}_j$. We emphasize that $g_{j,k}^{\alpha}$ has only implicit dependence on the divergent parameter $g/E_R a^{3/2}$ through $g_{\alpha\mathbf{K}}$ and so remains finite, see Figure 5.3(b). As $g_{\alpha\mathbf{K}} \ll g/a^{3/2}$, the transformation to the FRH has the effect of narrowing the resonance. In Figure 5.3(b) we also see that the on-site coupling $g_{000,000}$ is the dominant energy scale of the problem for large a_s/a, and that off-site couplings can also be large compared to other Hubbard parameters such as the open channel tunneling. Atoms which do not lie along a principal axis can pair to form a molecule, but this effect is much weaker than diagonal tunneling for the completely even parity dressed molecule.

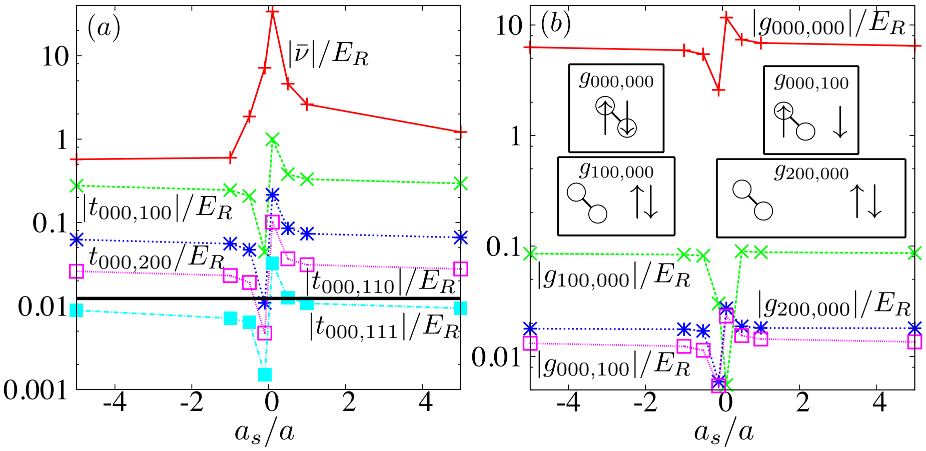


Figure 5.3: (Color online) *Hubbard parameters for the FRH.* (a) The detunings and tunnelings of the completely even parity dressed molecule in the lowest sheet as a function of a_s/a. The detuning is negative for $a_s < 0$ and positive otherwise. The solid black horizontal line is the tunneling of a single open channel fermion in the lowest band. The nearest neighbor dressed molecular tunneling is nearly two orders of magnitude larger than the open channel tunneling near resonance. (b) The effective atom-dressed molecule couplings of the completely even parity dressed molecule in the lowest sheet as a function of a_s/a. Schematics of the spatial dependence of the various coupling processes are shown in the boxes.

In the derivation of the FRH we use only the bound states of the projected problem and neglect scattering states in higher bands. This captures the scattering states in the lowest band and nearby bound states, but will fail to capture the physics at higher two-particle relative energy where scattering states in higher bands can play a role. In order to accommodate these scattering states, one can project out higher bands from χ as was done for the lowest band, and then include these bands dynamically in the many-body Hamiltonian with renormalized couplings. In this way, the energetic domain of application of the FRH can be extended arbitrarily at the expense of more dynamical fields. Within the confines of the two-channel model and the constraint of low energies, the FRH is an accurate representation of the many-body Hamiltonian. However, intrinsic three-body processes which are not captured by the two-channel model play a role at higher density and lead to corrections to the FRH. A discussion of these three-body processes is outside the scope of this paper.

In summary, we have studied the bound state properties of two Feshbach interacting fermions in an optical lattice at a range of scattering lengths and quasimomenta. The bound states of a projected Hamiltonian were used to identify a numerically tractable, efficient Hamiltonian for a low density many-body collection of lattice fermions at arbitrary scattering length and low energies, the Fermi Resonance Hamiltonian. Our results provide the appropriate starting point for future investigations of strongly interacting lattice fermions.

We acknowledge useful discussions with J. L. Bohn, H. P. Büchler, C. W. Clark, D. E. Schirmer, D. M. Wood, and Zhigang Wu. We also thank H. P. Büchler for providing computer code for comparison. This work was supported by the Alexander von Humboldt Foundation, AFOSR, NSF, and GECO.

5.3 Supplemental Material: Derivation of the Nonlinear Eigenequation.

Here we review the derivation of Eq. (2) of the main text for the bound states of two fermions in an optical lattice interacting via a zero-range Feshbach resonance. All quantities have their same meaning as in the main text. The starting point of our analysis is a two channel model with the open channel spanned by states of two fermions in different internal states with equal mass m and the closed channel spanned by molecular states with twice the fermionic mass and twice the fermionic polarizability. We describe their interaction via an inter-channel coupling g which couples the pair of open channel fermions to a closed channel molecule at the center of mass and a detuning ν between the two channels. This gives rise to the scattering amplitude

$$f(\mathbf{k}) = -\frac{1}{1/a_s + ik + r_b k^2}\,, \tag{5.8}$$

with s-wave scattering length $a_s = -2\mu g^2/4\pi\hbar^2\nu$ and effective range $r_B = \pi\hbar^4/\mu^2 g^2$. Here μ is the reduced mass and $\mathbf{k}$ the incident wavevector.

Denoting the wave function of the two fermions in the open channel as $\psi(\mathbf{x},\mathbf{y})$ and the wave function of the closed channel molecules as $\phi(\mathbf{z})$, the two-channel Schrödinger equation in position representation is

$$\begin{aligned}
[E-\hat{H}_0(\mathbf{x})-\hat{H}_0(\mathbf{y})]\psi(\mathbf{x},\mathbf{y}) &= g\textstyle\int d\mathbf{z}\alpha(\mathbf{r})\phi(\mathbf{z})\delta(\mathbf{z}-\mathbf{R})\,,\\
[E-\nu_0-\hat{H}_0^M(\mathbf{z})]\phi(\mathbf{z}) &= g\textstyle\int d\mathbf{x}d\mathbf{y}\alpha(\mathbf{r})\psi(\mathbf{x},\mathbf{y})\delta(\mathbf{z}-\mathbf{R})\,.
\end{aligned}$$

In this expression $\hat{H}_0(\mathbf{x})=-\frac{\hbar^2}{2m}\nabla_{\mathbf{x}}^2+V(\mathbf{x})$ is the single particle Hamiltonian for the open channel and $\hat{H}_0^M(\mathbf{z})=-\frac{\hbar^2}{4m}\nabla_{\mathbf{z}}^2+2V(\mathbf{z})$ is the single particle Hamiltonian for the closed channel. The subscript 0 in ν_0 denotes that this is a bare detuning entering the microscopic theory which is related to the physically observable detuning ν in the limit as the regularization cutoff $\Lambda\to\infty$. Additionally, we note that the Feshbach regularization $\alpha(\mathbf{r})\to\delta(\mathbf{r})$ in the limit $\Lambda\to\infty$, where $\delta(\mathbf{r})$ is the Dirac delta function.

The open channel solution with total quasimomentum $\mathbf{K}$ may be parameterized as

$$\psi_{\mathbf{K}}(\mathbf{x},\mathbf{y})=\frac{1}{\sqrt{N^3}}\sum_{\mathbf{nm}}\sum_{\mathbf{q}}\varphi_{\mathbf{nm}}^{\mathbf{Kq}}\psi_{\mathbf{n},\mathbf{q}}(\mathbf{x})\psi_{\mathbf{m},\mathbf{K}-\mathbf{q}}(\mathbf{y})\,, \tag{5.9}$$

where N^3 is the total number of unit cells in a 3D lattice with periodic boundary conditions and $\psi_{\mathbf{nq}}(\mathbf{x})$ is a Bloch eigenfunction of the single-particle Hamiltonian. As in the main text, quantities denoted in bold represent three-component vectors, e.g. $\mathbf{n}=(n_x,n_y,n_z)$. Similarly, we parameterize the closed channel wave function as a sum over Bloch states computed for twice the mass and twice the polarizability $\phi_{\mathbf{sK}}(\mathbf{z})$ as

$$\phi_{\mathbf{K}}(\mathbf{z})=\sum_{\mathbf{s}}R_{\mathbf{s}}^{\mathbf{K}}\phi_{\mathbf{sK}}(\mathbf{z})\,. \tag{5.10}$$

Inserting these expansions into the two-channel Schrödinger equation yields

$$\left[E_{\mathbf{K}} - E^{\mathbf{K}}_{\mathbf{nm}}(\mathbf{q})\right] \varphi^{\mathbf{Kq}}_{\mathbf{nm}} = \frac{g}{\sqrt{a^3}} \sum_{\mathbf{s}} h^{\mathbf{nm}}_{\mathbf{sK}}(\mathbf{q}) R^{\mathbf{K}}_{\mathbf{s}}\,,$$

$$\left[E_{\mathbf{K}} - \nu_0 - E^{M}_{\mathbf{sK}}\right] R^{\mathbf{K}}_{\mathbf{s}} = \frac{g}{\sqrt{a^3}} \sum_{\mathbf{nm}} \int \frac{d\mathbf{q}}{v_0} h^{\mathbf{nm}\star}_{\mathbf{sK}}(\mathbf{q}) \varphi^{\mathbf{Kq}}_{\mathbf{nm}}\,.$$

Formally solving the first of the two equations with a Green's function and inserting into the second equation gives

$$\left[E_{\mathbf{K}} - \nu_0 - E^{M}_{\mathbf{sK}}\right] R^{\mathbf{K}}_{\mathbf{s}} = \frac{g^2}{a^3} \int \tfrac{d\mathbf{q}}{v_0} \textstyle\sum_{\mathbf{mnt}} \tfrac{h^{\mathbf{nm}}_{\mathbf{sK}}(\mathbf{q}) h^{\mathbf{nm}\star}_{\mathbf{tK}}(\mathbf{q})}{E_{\mathbf{K}} - E^{\mathbf{K}}_{\mathbf{nm}}(\mathbf{q}) + i\eta} R^{\mathbf{K}}_{\mathbf{t}}\,,$$

where η is a positive infinitesimal. This expression diverges in the limit $\Lambda \to \infty$, as is well known for two-channel theories involving a pointlike boson [14]. We remove this divergence through renormalization, replacing the bare detuning ν_0 with the physical detuning ν by subtracting the infinite constant $\bar{\chi}^{\mathbf{K}}$, yielding Eq. (2) of the main text. The divergent parts of Eq. (3) in the main text cancel and we may safely take the limit $\Lambda \to \infty$.

Following Ref. [15], we use the regularization

$$\alpha(\mathbf{r}) = \int_{v(\Lambda)} \frac{d\mathbf{k}}{(2\pi)^3} e^{i\mathbf{k}\cdot\mathbf{r}}\,, \tag{5.11}$$

where the cubical volume $v(\Lambda) = v_0\Lambda^3$ is centered around $\mathbf{k} = 0$ with v_0 the volume of the BZ. We also define a shell summation over bands with shell parameter S, $\sum_{\mathbf{nm};S}$, as the summation over all band indices $\mathbf{n}$ and $\mathbf{m}$ less than or equal to S with at least one of the band indices being S. The correct limiting procedure to obtain $\chi^{\mathbf{K}}_{\mathbf{st}}(\mathbf{K})$ in the limit of an infinite summation over bands and vanishing short-distance cutoff is

$$\lim_{\Lambda\to\infty} \left[\lim_{S\to\infty} \chi^{\mathbf{K}}_{\mathbf{st}}(E_{\mathbf{K}})\right]\,. \tag{5.12}$$

The Λ limit is taken using the asymptotic scaling relation

$$\left[\chi^{\mathbf{K}}_{\mathbf{st}}(E_{\mathbf{K}})\right](\Lambda) = a_{\mathbf{st}}/\Lambda + \chi^{\mathbf{K}}_{\mathbf{st}}(E_{\mathbf{K}})\,. \tag{5.13}$$

On the right hand side, $a_{\mathbf{st}}$ the slope defining the scaling with Λ and $\chi^{\mathbf{K}}_{\mathbf{st}}(E_{\mathbf{K}})$ is the value as $\Lambda \to \infty$.

One may be concerned that the scaling relation Eq. (5.13) only holds for $\chi^{\mathbf{K}}$ of the full model and not for $\tilde{\chi}^{\mathbf{K}}$ in the projected model. To show that this is not the case, we note that in the limit of an infinite number of unit cells $N \to \infty$ the overlaps $h^{\mathbf{nm}}_{\mathbf{sK}}(q)$ may be written as products of 1D overlaps $h^{nm}_{sK}(q)$ of the form

$$\begin{aligned} h^{nm}_{sK}(q) = \lim_{\ell\to\infty} \sum_{r,r'=-\ell}^{\ell} & c^{r}_{nq} c^{r'}_{mK-q} c^{r+r'+f/2\pi}_{M;sK} \\ & \times \operatorname{rect}\Big(\frac{2q-K-f+2\pi(r-r')}{2\pi\Lambda}\Big)\,, \end{aligned} \tag{5.14}$$

where f is an integer multiple of 2π which shifts $K-q$ into the BZ, $\operatorname{rect}(x)$ denotes the rectangle function, and the vectors $\mathbf{c}_{nq}$ denote the Fourier expansion of the open channel Bloch functions as

$$\psi_{nq}(x) = e^{iqx} \lim_{\ell\to\infty} \sum_{r=-\ell}^{\ell} c^{r}_{nq} e^{-2\pi i r x/a}/\sqrt{Na}\,. \tag{5.15}$$

ℓ represents a finite Fourier cutoff used in numerics. Similarly, $\mathbf{c}_{M;sK}$ denote the Fourier coefficients of the closed channel Bloch functions. $\tilde{\chi}^{\mathbf{K}}$ differs from $\chi^{\mathbf{K}}$ in the exclusion of all terms with $n = m = 1$. However, provided that Λ is large enough to capture the support of the vectors $\mathbf{c}_{nq}$ with $n = 1$, Eq. (5.14) demonstrates that these terms are no longer functions of Λ. Thus, the scaling relation Eq. (5.13) also holds for $\tilde{\chi}^{\mathbf{K}}$. Similar arguments show that the same scaling holds for $\tilde{\chi}^{\mathbf{K}}$ when any finite number of open channel bands have been projected out.

5.4 Supplemental Material: Classification of the Two-Particle Bound States

For the simple cubic lattice we consider the Hamiltonian is invariant under reflection in any Cartesian direction: $H(\theta_{\mathcal{R}}\mathbf{x}) = H(\mathbf{x}') = H(\mathbf{x})$ where $\mathbf{x}' \equiv \theta_{\mathcal{R}}\mathbf{x}$ is related to $\mathbf{x}$ by changing the sign of all coordinates in some set $\mathcal{R}$: $x_j \to -x_j$, $j \in \mathcal{R}$. Because the generators of reflection and translation do not commute we cannot find simultaneous eigenfunctions except at high-symmetry points of the BZ. However, the fact that the Hamiltonian commutes with both operators implies that parity transformations yield relationships between degenerate sets of Bloch functions. In particular, for the given lattice potential, the invariance under the reflection symmetry generated by $\theta_{\mathcal{R}}$ implies that the Bloch functions transform as

$$\psi_{\mathbf{nq}}(\mathbf{x}') = \prod_{j\in\mathcal{R}} (-1)^{n_j+1} \psi_{\mathbf{n},\mathbf{q}'}(\mathbf{x}) , \tag{5.16}$$

where we begin indexing the bands from 1. We can thus characterize the bands according to whether they are even or odd under inversions by the triple $\mathbf{p} = (p_x, p_y, p_z)$, where $p_\nu = (-1)^{n_\nu+1}$. This inversion relationship implies that the inter-channel overlaps transform as

$$h^{\mathbf{nm}}_{\mathbf{sK}'}(\mathbf{q}') = \prod_{j\in\mathcal{R}} (-1)^{n_j+m_j+s_j+1} h^{\mathbf{nm}}_{\mathbf{sK}}(\mathbf{q}) , \tag{5.17}$$

and $\chi^{\mathbf{K}}$ transforms as

$$\chi^{\mathbf{K}'}_{\mathbf{st}}(E_{\mathbf{K}}) = \prod_{j\in\mathcal{R}} (-1)^{s_j+t_j} \chi^{\mathbf{K}}_{\mathbf{st}}(E_{\mathbf{K}}) . \tag{5.18}$$

It can be proven that this transformation leaves the eigenvalues invariant, but the eigenvectors $\mathbf{R}^{\mathbf{K}}_{\alpha}$ transform according to

$$R^{\mathbf{K}'}_{\mathbf{s}\alpha} = \prod_{j\in\mathcal{R}} (-1)^{s_j+1} R^{\mathbf{K}}_{\mathbf{s}\alpha} . \tag{5.19}$$

For a total quasimomentum $\mathbf{K}$ all of whose components consist of either 0 or $-\pi/a$, this implies that only molecular bands which transform identically under *complete* inversions mix. Hence, at these exceptional points of the BZ, we can unambiguously determine the parity of the two-particle state $\psi_{\mathbf{K}\alpha}(\mathbf{x},\mathbf{y})$ by the components of its associated eigenvector $\mathbf{R}_{\alpha}^{\mathbf{K}}$. The parity is then chosen to depend only on the eigenstate index α by requiring that $\mathbf{R}_{\alpha}^{\mathbf{K}}$ is a smooth function of $\mathbf{K}$. This construction follows that of the 1D case studied by Kohn [16], which leads to maximally localized Wannier functions.

With this construction, there is still an undefined global phase under inversion that we can fix in the following way. The complete two-particle bound state solution is

$$\begin{aligned}\Psi_{\mathbf{K}\alpha}(\mathbf{x},\mathbf{y}) = \frac{1}{\mathcal{N}_{\mathbf{K}\alpha}}\Big[\sum_{\mathbf{s}} R_{\mathbf{s}\alpha}^{\mathbf{K}}\phi_{\mathbf{sK}}(\mathbf{x})\tilde{r}(\mathbf{x}-\mathbf{y}) \\ + \frac{g}{\sqrt{N^3a^3}}\sum_{\mathbf{nms;q}}\frac{R_{\mathbf{s}\alpha}^{\mathbf{K}}h_{\mathbf{sK}}^{\mathbf{nm}}(\mathbf{q})\psi_{\mathbf{nq}}(\mathbf{x})\psi_{\mathbf{mK-q}}(\mathbf{y})}{E_{\mathbf{K}}^{\alpha}-E_{\mathbf{nm}}^{\mathbf{K}}(\mathbf{q})}\Big],\end{aligned} \tag{5.20}$$

where $\mathcal{N}_{\mathbf{K}\alpha}$ is a normalizing factor and $\tilde{r}(\mathbf{x}-\mathbf{y})$ denotes a relative wavefunction for the closed channel which has characteristic width a/Λ. As the theory has already been regularized, we may take $\Lambda \to \infty$ with the understanding that this relative wavefunction has a probability density of 1, and forces the closed channel to contribute only at the center of mass. Because of the partitioning of Hilbert space into open and closed channels, the normalization coefficient is

$$\mathcal{N}_{\mathbf{K}\alpha}^2 = 1 - \left(\frac{g}{E_R a^{3/2}}\right)^2 \mathbf{R}_{\alpha}^{\mathbf{K}}\cdot\chi'(E_{\mathbf{K}}^{\alpha}/E_R)\cdot\mathbf{R}_{\alpha}^{\mathbf{K}}\,. \tag{5.21}$$

Here $\chi'(E)$ is the derivative of χ with respect to E. Using the transformation properties under $\theta_{\mathcal{R}}$, we find

$$\Psi_{\mathbf{K}'\alpha}(\mathbf{x},\mathbf{y}) = P_{\alpha}\Psi_{\mathbf{K}\alpha}(\mathbf{x}',\mathbf{y}')\,. \tag{5.22}$$

Accordingly, we set $P_\alpha = \prod_{j\in\mathcal{R}} p_\nu$. This implies that the dressed molecular Wannier functions transform as $\mathcal{W}_{i\alpha}(\mathbf{x}', \mathbf{y}') = P_\alpha \mathcal{W}_{i\alpha}(\mathbf{x}, \mathbf{y})$.

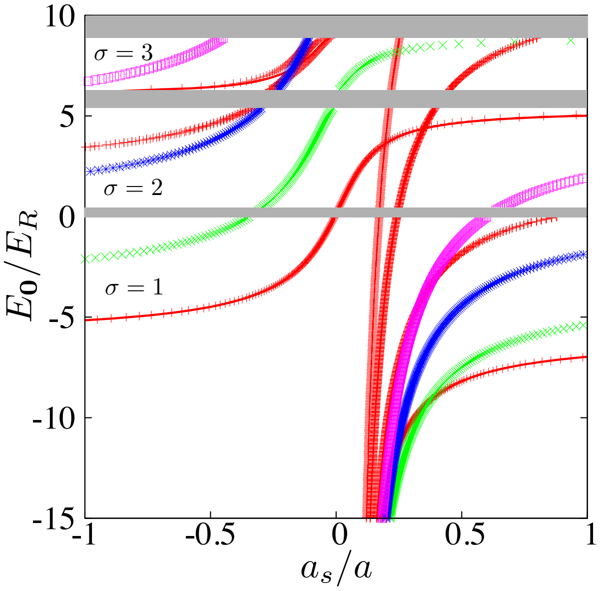


Figure 5.4: (Color online) *Classification of two-particle bound states.* The bound state energies at $\mathbf{K} = 0$ for a lattice of depth $V/E_R = 12$ are classified according to their parity and sheet indices. Red corresponds to $\mathbf{p} = (1, 1, 1)$, green to $\mathbf{p} = (1, 1, -1)$, blue to $\mathbf{p} = (1, -1, -1)$, and magenta to $\mathbf{p} = (-1, -1, -1)$. In addition, higher molecular bands for the $\mathbf{p} = (1, 1, 1)$ level are shown. These give rise to weak scattering resonances with the lowest open channel band for $a_s/a > 0$ and avoided crossings in the higher sheets for $a_s/a < 0$.

In Figure 5.4 we display the bound state energies at $\mathbf{K} = 0$ for a lattice of depth $V/E_R = 12$ classified according to their parity. The red points correspond to $\mathbf{p} = (1, 1, 1)$, the green points to $\mathbf{p} = (1, 1, -1)$ et cyc, the blue points to $\mathbf{p} = (1, -1, -1)$ et cyc, and the magenta points to $\mathbf{p} = (-1, -1, -1)$. In contrast to the continuum where there exists at most one bound state for fixed scattering length a_s, there is the possibility of several bound states for fixed a_s in the lattice due to the reduced translational symmetry. Thus, the parity and the quasimomentum are not sufficient to completely describe the states. For a fixed s-wave scattering length a_s, we provide two other indices which we call the *sheet index* $\sigma \in \{1, 2, \ldots, \infty\}$ and the *molecular* *band index* $s \in \{1, 2, \ldots, \infty\}$. The sheet index labels the open channel two-particle scattering bands, with the convention that the first sheet lies below the first band, the second sheet lies between the first and second bands, etc. as indicated on the figure. The open channel scattering bands are denoted by solid grey stripes. The molecular band index labels eigenstates which have the same parity and sheet indices but differ in energy. The number of molecular bands obtained is restricted by the number of closed channel bands used to construct $\chi^{\mathbf{K}}$. Let us define m to be the maximum value of the closed channel band index along any Cartesian direction. In Figure 5.4, the solid red line corresponds to the completely even parity state computed with $m = 2$ and the red points correspond to the complete even parity states computed with $m = 3$. The choice $m = 2$ captures the physics well near the lowest open channel scattering band. For $a_s/a > 0$, the higher molecular bands cross the lowest open channel scattering continuum at narrow ranges of a_s/a, leading to weak scattering resonances. For $a_s/a < 0$, the higher molecular bands are present at extended ranges of a_s/a, and avoided crossings between these molecular bands can lead to differences with lower m computations, see e.g. the third sheet near $a_s/a = -0.2$.

5.5 Supplemental Material: Finite Width Resonances

We now turn our attention briefly to the case where $g/E_R a^{3/2}$ and ν/E_R are finite. In this case we rearrange Eq. (2) of the main text to read

$$\sum_{\mathbf{t}} \left[(E_{\mathbf{K}} - E^{M}_{\mathbf{sK}})\delta_{\mathbf{st}} - \frac{g^2}{a^3}\chi^{\mathbf{K}}_{\mathbf{st}}(E_{\mathbf{K}}) \right] R^{\mathbf{K}}_{\mathbf{t}} = \nu R^{\mathbf{K}}_{\mathbf{s}} \tag{5.23}$$

which is an ordinary eigenvalue equation for the detuning ν when $E_{\mathbf{K}}$ and g are treated as fixed. We note that g cannot be scaled out of this equation as the molecular band energies $E^{M}_{\mathbf{sK}}$ depend only on the lattice strengths and masses and not on the resonance width. The solution of this equation for various r_B and $\mathbf{K} = 0$ is shown in Figure 5.5.

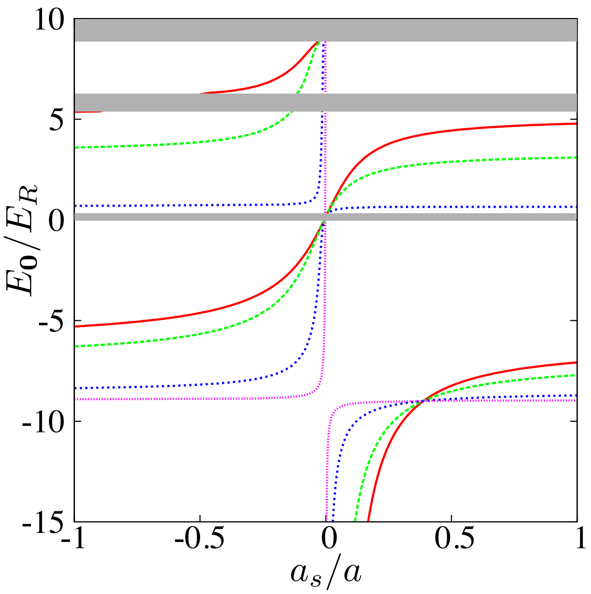


Figure 5.5: (Color online) *Bound state energies for finite width resonances.* Shown are the lowest energy bound state energies for $\mathbf{p} = (1, 1, 1)$ at $\mathbf{K} = 0$ in a strong optical lattice with $V/E_R = 12$. The red solid line is $r_B/a = 0.01$, the green dashed line is $r_B/a = 0.1$, the blue dotted line is $r_B/a = 1$, and the magenta short-dashed line is $r_B/a = 10$. For narrow resonances (large r_B) the divergence of a_s is sharply pronounced around a narrow energy range, and is shifted downwards from the broad resonance value, compare Figure 5.4.

We characterize the width of the resonance in terms of the experimentally measurable effective range r_B which defines the width as $g/E_R a^{3/2} = \sqrt{16a/\pi^3 r_B}$. For narrow resonances with large r_B only the lowest resonance can be seen, and a_s/a is greater than 1, corresponding to strong interactions, only in a very narrow energy range. As the resonance becomes broader the energy range over which the system is strongly interacting widens, and we begin to see resonant behavior near higher scattering continua. Additionally, the positions of the narrow resonances are shifted downwards in energy with respect to the broad resonances, eventually becoming the free molecular band energies. We note that the broadest resonance shown is in fact narrower than typical broad resonances found in the experimentally relevant ultracold atomic systems, but differs from the infinitely broad resonance results by at most a few percent. This justifies our use of the infinitely broad resonance limit in the main text.

5.6 References Cited

[1] D. M. Eagles, Phys. Rev. **186**, 456 (1969); Q. Chen, J. Stajic, S. Tan, and K. Levin, Phys. Rep. **412**, 1 (2005).

[2] J. Ranninger and J. M. Robin, Physica C **253**, 279 (1995).

[3] . M. Kohl *et al.*, Phys. Rev. Lett. **94**, 080403 (2005).

[4] A. Adams, L. D. Carr, T. Schaefer, P. Steinberg, and J. E. Thomas, N. J. Phys., Focus Issue in press (2012).

[5] A. A. Cafolla, S. E. Schnatterly, and C. Tarrio, Phys. Rev. Lett. **55**, 2818 (1985).

[6] S. Trotzky *et al.*, Nat. Phys. **6**, 998 (2010).

[7] T. Esslinger, Ann. Rev. Cond. Matt. Phys. **1**, 129 (2010).

[8] . H. P. Büchler, Phys. Rev. Lett. **104**, 090402 (2010); H. P. Büchler, Phys. Rev. Lett. **108**, 069903 (2012).

[9] L. D. Carr and M. J. Holland, Phys. Rev. A **72**, 033622(R) (2005); D. B. M. Dickerscheid, U. A. Khawaja, D. van Oosten, and H. T. C. Stoof, Phys. Rev. A **71**, 043604 (2005); R. B. Diener and T.-L. Ho, Phys. Rev. Lett. **96**, 010402 (2006); L.-M. Duan, Phys. Rev. Lett. **95**, 243202 (2005); L.-M. Duan, EPL **81**, 20001 (2008); X. Cui, Y. Wang, and F. Zhou, Phys. Rev. Lett. **104**, 153201 (2010).

[10] P. Lancaster, *Lambda-matrices and Vibrating Systems,* (Pergamon Press, Oxford, 1966).

[11] M. L. Wall and L. D. Carr, in preparation.

[12] J. von Stecher, V. Gurarie, L. Radzihovsky, and A. M. Rey, Phys. Rev. Lett. **106**, 235301 (2011).

[13] D. Jaksch *et al.*, Phys. Rev. Lett. **81**, 3108 (1998).

[14] S. J. J. M. F. Kokkelmans, J. N. Milstein, M. L. Chiofalo, R. Walser, and M. J. Holland, Phys. Rev. A **65**, 053617 (2002).

[15] . H. P. Büchler, Phys. Rev. Lett. **108**, 069903 (2012).

[16] . W. Kohn, Phys. Rev. **115**, 809 (1959).

PART IV

MATRIX PRODUCT STATES

CHAPTER 6

MATRIX PRODUCT STATES: FOUNDATIONS

The bulk of the numerical results for many-body systems contained in this thesis are obtained by variational algorithms on a class of states known as matrix product states (MPSs). The theory of MPSs as a variational ansatz for eigenstates and dynamics of general finite-sized one-dimensional (1D) systems is expounded at length in Chapter 7. In addition, the definitive review of MPSs at the time of the writing of this thesis is Ref. [1]. In this chapter, we aim instead to give intuitive notions of what MPSs are and why they are useful as variational ansätze.

6.1 Bird's Eye View of Matrix Product States

The quantum many-body problem is in principle completely solved given the microscopic degrees of freedom and their interactions, as the relevant Schrödinger equation is known once the Hamiltonian has been specified. However, in practice, the Hilbert space of a typical many-body ensemble grows exponentially with the number of constituents of the system, and so an algorithm which explicitly forms a matrix representation of the Hamiltonian and diagonalizes it to find eigenstates is limited to very small systems. Even when all symmetries of the Hamiltonian have been carefully accounted for and sparse diagonalization routines such as the Lanczos [2] or Davidson [3] algorithms are used to find only extremal eigenstates, cutting edge *exact diagonalization* is limited to approximately 40 two-component spins or a 20-site fermionic Hubbard model on the square lattice at half filling.

The full Hilbert space of a many-body system is in fact too big for physical discussions, as can be shown by the following argument [4]. Consider a time-dependent Hamiltonian acting on N particles

$$\hat{H}(t) = \sum_{X \in \{1,2,\ldots,N\}} \hat{H}_X(t) \; , \tag{6.1}$$

where X labels subsets of the N particles. We restrict each term in the summation to have bounded norm $\|\hat{H}_X(t)\| \leq E$ and to act on no more than k particles at a time, where k is independent of the number of particles. An operator acting on k particles simultaneously is said to be *k-local.* Note that we make no assumptions about the range of the Hamiltonian, and so our arguments apply to the long-ranged Hamiltonians relevant for molecules. We now wish to expand the full time-ordered propagator of this Hamiltonian in terms of a series of at most k-local unitary operators. This can be accomplished by using the time-ordered Trotter expansion for two non-commuting terms $\hat{H}_1(t)$ and $\hat{H}_2(t)$ [5]

$$\hat{U}(t, t+\delta t) = \mathcal{T}\exp\left[-i\int_t^{t+\delta t} dt' \left(\hat{H}_1(t') + \hat{H}_2(t')\right)\right] , \tag{6.2}$$

$$\approx \hat{U}_{\text{Trotter}}(t, t+\delta t) = \mathcal{T}\exp\left[-i\int_t^{t+\delta t} dt' \hat{H}_1(t')\right] \mathcal{T}\exp\left[-i\int_t^{t+\delta t} dt' \hat{H}_2(t')\right] , \tag{6.3}$$

which is accurate in the operator norm as

$$\|\hat{U}(t, t+\delta t) - \hat{U}_{\text{Trotter}}(t, t+\delta t)\| \leq c_{\text{Trotter}} (\delta t)^2 \; , \tag{6.4}$$

$$c_{\text{Trotter}} = \frac{1}{(\delta t)^2} \int_t^{t+\delta t} dt' \int_t^{t'} dt'' \| \left[\hat{H}_1(t''), \hat{H}_2(t')\right] \| \, . \tag{6.5}$$

Here $\mathcal{T}\exp[\bullet]$ is the time-ordered exponential [6]. We note that these bounds do not depend on the smoothness of the Hamiltonian, and so are also valid for non-analytic time dependence. Iterating this expansion for two operators $\log_2(L)$ times, where $L \in \text{poly}(N)$ is the number of k-particle terms in Eq. (6.1), we may write our full propagator as a product of time-ordered k-local unitaries with the total error bounded by

$$\frac{1}{2}\left|\max_X \sup_{0\le t'\le t}\|\hat{H}_X\left(t'\right)\|\right|^2 L^2\left(\delta t\right)^2 \ . \tag{6.6}$$

The notation $L\in \mathrm{poly}\left(N\right)$ denotes that L is a polynomial function of N. This error may be made as small as desired by taking $\left(\delta t\right)^{-1}\in \mathrm{poly}\left(N\right)$. The number of k-body unitaries which is required to bound this error by ϵ is inversely proportional to ϵ and polynomial in both t, the final time desired, and L. One can replace these polynomially many k-body unitaries with a discrete set of fixed one and two-body operators which is also polynomial in L due to the Solovay-Kitaev theorem [7–9].

From the above, we conclude that any state which can be produced from a given state $|0\rangle$ under evolution by an arbitrary time-dependent Hamiltonian can be well approximated by the state $|0\rangle$ acted on by a polynomial-size quantum circuit with a fixed set of discrete operations. It is known [8] that such circuits can only reach an exponentially small subset of states, and so we come to the conclusion that physically relevant states, those that can be reached from a reference state to a desired accuracy from any time-dependent Hamiltonian in a time polynomial in the size of the system, form an exponentially small subset of the complete Hilbert space. We note that this general argument is independent of dimension, and can be extended from pure states to open quantum systems described by a k-local time-dependent Liouvillian [10]. Hence, the argument also applies at finite temperature. The emphasis of the many-body problem now shifts from, "How do we deal with the largeness of Hilbert space?" to, "How do we parameterize the set of states relevant to our particular many-body model?"

The answer to this latter question can be provided assuming that our Hamiltonian has finite-range interactions and a gap to excitations[28] and we consider our system

---

[28] By which we mean a *spectral gap* in the thermodynamic limit, $E_1-E_0>0$ where E_1 is the energy of the first excited state and E_0 is the energy of the ground state. Finite-sized systems whose infinite counterparts are gapless typically have a gap which vanishes as an inverse polynomial in the system volume.

at zero temperature. The ground states of such systems obey an *area law* [11–15], which is to say that the von Neumann entropy of entanglement of the ground state in a region A,

$$S_{\mathrm{vN}}\left(A\right) = -\mathrm{Tr}\left[\hat{\rho}_A \log \hat{\rho}_A\right] \, , \tag{6.7}$$

where $\hat{\rho}_A$ is the reduced density operator describing region A, scales only with the border of region A and not with its volume. For 1D systems, the presence of an area law implies that the entanglement between any two complementary subregions is independent of their size. This is in contrast with a random state in Hilbert space, for which the entropy of entanglement is extensive. The existence of area laws has been proven rigorously in 1D for spin-1/2 systems obeying the above constraints by Hastings [16], and has been observed to hold for the vast majority of other known systems satisfying our hypotheses regardless of dimension.[29] At finite temperature a condition similar to the area law can be made rigorous in any dimension as the *mutual information* between a region A and its complement $\bar{A}$ in a system at thermal equilibrium,

$$M\left(A;\bar{A}\right) = S_{\mathrm{vN}}\left(A\right) + S_{\mathrm{vN}}\left(\bar{A}\right) - S_{\mathrm{vN}}\left(A+\bar{A}\right) \, , \tag{6.8}$$

satisfies an area law whose bound is inversely proportional to the temperature [18]. Here the density matrix used to obtain $S_{\mathrm{vN}}\left(X\right)$, $\hat{\rho}_X$, is the reduced density matrix describing region X in a system at thermal equilibrium.

A related concept in physics which gives an enlightening alternative perspective on area laws for entanglement entropy is the notion of area laws for black holes [19]. Here, it is the thermodynamic entropy of a black hole that scales with the area

---

[29] For a counterexample in a non-translationally invariant 1D chain, see Ref. [17]. Here, a variant of the Dasgupta-Ma-Fisher renormalization group procedure for random spin chains is used to explicitly construct a spin chain satisfying a volume law. Additionally, as this system is not translationally invariant, the volume law depends crucially on how the system is divided. In fact, there exist divisions for which the entropy of entanglement is identically zero.

enclosing the black hole rather than its volume [20]. As the entropy of any volume of spacetime is bounded by the entropy of a black hole that fits inside of that volume [21], we have that entropy in gravity is always sub-extensive. This fact coupled with *holographic dualities* [19] mapping the properties of $D+1$-dimensional bulk spacetime to a D-dimensional quantum field theory on its boundary, which are being proved on a case-by-case basis, provide strong evidence that area laws are ubiquitous in strongly correlated quantum systems. A similar holographic principle applies for matrix product states and their generalizations, as expectation values of operators may be determined in terms of the dynamics of a dissipative boundary theory [22, 23].

A notable class of exceptions to our hypotheses are systems at the critical point of a quantum phase transition in which the gap vanishes. For systems of free fermions in D dimensions, the corrections to the area law are logarithmic in the linear dimension of region A, L_A, [24, 25]

$$S_{\mathrm{vN}}\left(A\right) \sim L_A^{D-1}\log L_A\,, \tag{6.9}$$

while area laws in bosonic systems in dimensions $D \geq 2$ appear to be insensitive to criticality [11, 26–28]. For 1D systems whose universal critical theory is a conformal field theory [29] the logarithmic corrections to the area law are made precise by the *Calabrese-Cardy formula* [12, 14, 30]

$$S_{\mathrm{vN}}\left(A\right) \sim \frac{c}{6}\log L_A\,. \tag{6.10}$$

Here c is the *central charge* of the conformal field theory [29] which, in an intuitive picture, counts the number of universal bosonic degrees of freedom.[30] For example, the central charge of a free bosonic system is 1 and a free fermionic system has $c = 1/2$, corresponding to “half a boson.” The above results suggest that a reasonable class of

---

[30] Rigorously, the central charge is determined by the coefficient of the anomalous term in the commutator of the energy-momentum tensor at two different positions [31]. Hence, c describes the behavior of a conformally invariant system when a macroscopic length scale is introduced.

states to be used for variational studies of 1D systems should satisfy an area law, and possibly be able to handle weak logarithmic violations efficiently.

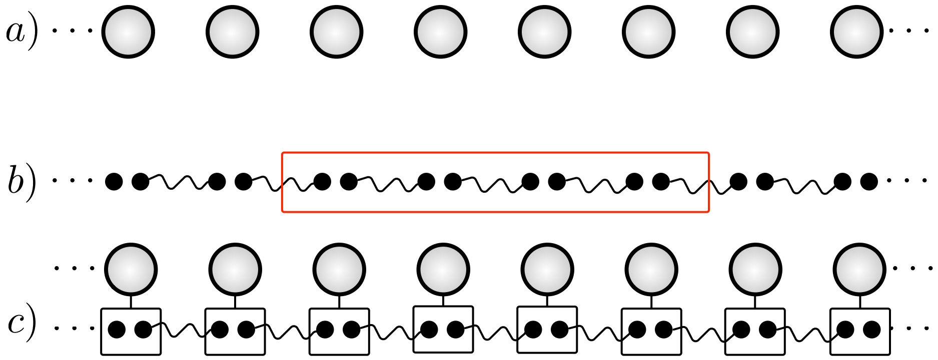


Figure 6.1: Schematic of the MPS construction. a) A 1D system comprised of local Hilbert spaces (spheres) arranged in a regular pattern. b) Each Hilbert space is replaced by two fictitious maximally entangled χ-level systems, where the wiggly lines indicate which pairs are maximally entangled. Partitioning the system as with the red box, entanglement of the boxed region with its complement occurs only at the boundary. c) The maximally entangled fictitious systems are projected to produce an entangled state in the physical degrees of freedom satisfying an area law.

A construction of a 1D state which obeys an area law is to take a chain of sites and replace each site with a pair of χ-level systems which are maximally entangled with their neighbors [32]. This construction is shown in Figure 6.1. If we are now to isolate any region A such as that enclosed by the red box in Figure 6.1(b), the entanglement of this region with the remainder of the system is generated only by the entanglement between the pairs of χ-level systems at the boundaries. This boundary entanglement scales as $\log\chi$. These χ-level subsystems thus provide a fictitious set of states which can be manipulated such as to produce the desired entanglement structure in the physical degrees of freedom. We manipulate these fictitious systems by projecting from this fictitious Hilbert space onto the physical Hilbert space via a set of isometric tensors. These tensors form the parameters of an MPS. Finally, we note that the relationship $S_{\mathrm{vN}} \sim \log\chi$ implies that χ, the Hilbert space dimension of the fictitious system, is exponential in the entanglement entropy S_{vN}.[31] For systems with weak logarithmic violations of the area law obeying Eq. (6.10), this gives that χ is polynomially related to the system size, with polynomial degree depending on the central charge [33]. Hence, any algorithm which is polynomial in χ represents a method which is at most polynomial in the system size, an exponential improvement over exact diagonalization![32]

An enlightening alternative view of MPSs is that they are states which are the result of a real-space *renormalization group* (RG) iteration [35]. The essential idea of the renormalization group is to coarse-grain a system by integrating out the irrelevant degrees of freedom. As coarse graining is iteratively applied, only the degrees of freedom which describe the macroscopic behavior of the system emerge, and we can characterize these macroscopic degrees of freedom by the *universality classes* which represent fixed points of the RG iteration. Models whose microscopic details differ greatly may fall into the same universality class, and hence have the same macroscopic behavior. The renormalization group provides a quantitative means to sort systems according to their universal degrees of freedom via determining which RG fixed point they move towards as irrelevant degrees of freedom are removed.

The procedure of the original numerical renormalization group algorithm of Wilson is to coarse-grain a system by iterative exact diagonalization. In one dimension, we begin by assuming that our system is comprised of *blocks* $B^{(0)}$, where each block may represent a single site or multiple sites and are taken to be identical for simplicity. Each block is indexed by some set of χ many-body states which span its Hilbert space. We now group two contiguous blocks together, diagonalize the Hamiltonian of this two-block system, and keep only the χ states which are lowest in energy. The

---

[31] That is to say, the χ that is required to accurately reflect the entanglement structure grows exponentially with the von Neumann entropy.

[32] Strictly speaking, this is true only for fixed error. If we require that the error be bounded by an inverse polynomial in the system size, finding a ground state with the given representation is still very difficult [34].

transformation to the new configuration where two of the blocks $B^{(0)}$ are combined into a single *superblock* $B^{(1)} = B^{(0)} \otimes B^{(0)}$ may be represented by an isometric matrix A[33] mapping from $B^{(0)} \otimes B^{(0)} \rightarrow B^{(1)}$. We now combine $B^{(1)}$ and one of its neighboring blocks $B^{(0)}$ together, diagonalize the Hamiltonian in this space, and keep only the low energy states again, resulting in an isometric tensor mapping from $B^{(1)} \otimes B^{(0)} \rightarrow B^{(2)}$. This outlines the general structure of the iteration.

Let us now assume that out initial blocks are the two leftmost sites of a 1D lattice, and denote the isometric tensors constructed at iteration n as $A^{[n]i_n}_{\alpha\beta}|\alpha\rangle|i\rangle\langle\beta|$, where $|\alpha\rangle$ are the states spanning $B^{(n-1)}$, $|i\rangle$ are the states spanning $B^{(0)}$, and $|\beta\rangle$ are the states spanning $B^{(n)}$. Furthermore, let us take open boundaries on a chain of L sites, which amounts to the left basis of $A^{[1]i_1}$ and the right basis of $A^{[L]i_L}$ being one-dimensional.[34] Then, the many-body state resulting at iteration L may be written as

$$|\psi\rangle = \sum_{i_1 \dots i_L} A^{[1]i_1} \dots A^{[L]i_L} |i_1 \dots i_L\rangle \,, \tag{6.11}$$

which, as we shall see, is the form of an MPS.

This procedure produced excellent results for the Kondo problem [35], but fails miserably for the toy model of a single particle on a 1D chain! The reasons for this failure were investigated by White [36], and led him to the *density-matrix renormalization group* (DMRG) algorithm, in which the relevant states which are kept after diagonalization of the superblock Hamiltonian are not those with the lowest energy, but those which have the largest eigenvalues in the block reduced density matrix. Examining Eq. (6.7), we can see that this corresponds intuitively to the states which

---

[33] By isometric, we mean that this matrix has orthonormal rows. If it were square, it would be unitary, but we are transforming from a χ^2 dimensional space to a χ dimensional space, and so only the rows are orthonormal.

[34] That is, at the first iteration we map from the product of the vacuum and a block $B^{(0)}$ to a new set of states which is of course identical to the block $B^{(0)}$. A similar comment applies to the last iteration.

are most strongly entangled to the "environment" formed by all previous iterations.[35] The more precise statement we can make is that by keeping the states which have largest eigenvalues in the reduced density matrix we minimize the 2-norm distance between the system in the truncated Hilbert space and the true ground state with the given environment, which is borne out in the *Schmidt decomposition* [8, 38]

$$|\psi\rangle = \sum_{\alpha} \lambda_{\alpha} |A_{\alpha}\rangle |B_{\alpha}\rangle \,. \tag{6.12}$$

The Schmidt decomposition represents a general quantum state $|\psi\rangle$ in terms of orthonormal bases $|A\rangle$ and $|B\rangle$ for two complementary subsystems and the eigenvalues of the reduced density matrix $\hat{\rho}_A$, $\{\lambda_{\alpha}^2\}$. With this change to the original numerical renormalization group procedure, DMRG not only overcomes the failure for the single-particle case, but turns this real-space renormalization group procedure into a method of unparalleled power for strongly-correlated 1D systems.

After the development of DMRG, it was realized that DMRG can be formulated as a variational method in terms of MPSs [39, 40]. While this does not lead to any significant numerical improvement in the algorithm to find the ground state of 1D Hamiltonians [1], other variational algorithms, for example to find excited states or perform generic time evolution, benefit greatly from this observation. The reason is that in DMRG all states involved in a calculation are represented as a single MPS, which requires that they share common bases in the matrix product. Hence, representing states together as an MPS often requires vastly more resources than representing each state separately as an MPS for a fixed error. These issues are discussed at length in Chapter 7. Hence, in this thesis, we focus on the explicit formulation of variational algorithms within the class of MPSs rather than the implicit MPS representation used in DMRG.

---

[35] Note that, strictly speaking, DMRG does not maximize entanglement due to a renormalization of the density-matrix spectra induced by truncation [37]. However, for most physical systems this intuition does not cause any pitfalls.

6.2 Definitions

We now move to explicit definitions of the mathematical objects which are used when discussing MPSs. We define a *tensor* as a map from a product of Hilbert spaces to the complex numbers

$$T : \mathbb{H}_1 \otimes \mathbb{H}_2 \otimes \cdots \otimes \mathbb{H}_r \to \mathbb{C}\,. \tag{6.13}$$

Here r is the *rank*[36] of the tensor. If we evaluate the elements of the tensor T in a fixed basis $\{|i_j\rangle\}$ for each Hilbert space $\mathbb{H}_j$, then equivalent information is carried in the multidimensional array $T_{i_1\dots i_r}$. We will also refer to this multidimensional array as a tensor. The information carried in a tensor does not change if we change the order in which its indices appear. We will call such a generalized transposition a *permutation* of the tensor. As an example, the permutations of the rank-3 tensor T are

$$T_{ijk} = [T']_{kij} = [T'']_{jki} = [T''']_{jik} = [T'''']_{kji} = [T''''']_{ikj}\ . \tag{6.14}$$

Here, the primes indicate that the tensor differs from its unprimed counterpart only by a permutation of indices. Similarly, by combining two such indices together using the Kronecker product we can define an equivalent tensor of lower rank, a process we call *index fusion*. We denote the Kronecker product of two indices a and b using parentheses as (ab), and a representation is provided by

$$(ab) = (a-1)\,d_b + b\,, \tag{6.15}$$

where d_b is the dimension of $\mathbb{H}_b$ and a and b are both indexed starting from 1. An example of fusion is

---

[36] This should not be confused with the rank of a matrix, which is the number of nonzero singular values. We shall avoid confusion in this text by referring to the number of nonzero singular values as the *matrix rank* or rank of a matrix and referring to the definition given here as the *tensor rank*, the rank of a tensor, or a rank-r tensor.

$$T_{ijk} = [T']_{i(jk)} \ . \tag{6.16}$$

Here, T is a rank-3 tensor of dimension $d_i \times d_j \times d_k$ and T' is a matrix of dimension $d_i \times d_j d_k$. The inverse operation of fusion, which involves creating a tensor of higher rank by splitting a composite index, we refer to as *index splitting*.

Just as permutations generalize the notion of matrix transposition, *tensor contraction* generalizes the notion of matrix multiplication. In a contraction of two tensors A and B some set of indices $\mathbf{c}_A$ and $\mathbf{c}_B$ which describe a common Hilbert space are summed, and the resulting tensor C consists of products of the elements of A and B as

$$C_{\bar{\mathbf{c}}_A \bar{\mathbf{c}}_B} = \sum_{\mathbf{c}} A_{\bar{\mathbf{c}}_A \mathbf{c}} B_{\mathbf{c}\bar{\mathbf{c}}_B} \ . \tag{6.17}$$

Here $\bar{\mathbf{c}}_B$ denotes the indices of A which are not contracted and likewise for $\bar{\mathbf{c}}_B$. The rank of C is $r_A + r_B - 2n_c$, where n_c is the number of indices contracted (i.e., the number of indices in $\mathbf{c}$) and r_A and r_B are the ranks of A and B, respectively. In writing expression Eq. (6.17) we have permuted all of the indices $\mathbf{c}_A$ to be contracted to the furthest right position in A and the indices $\mathbf{c}_B$ to the furthest leftmost position in B for notational simplicity. If we were also to fuse the elements in $\bar{\mathbf{c}}_A$, $\mathbf{c}$, and $\bar{\mathbf{c}}_B$ together, then we would recognize Eq. (6.17) as a matrix-matrix multiplication. This is a valuable insight, as matrix-matrix multiplication routines such as DGEMM[37] in BLAS[38] [41] have been highly optimized and this lends efficiency to tensor contraction algorithms.

---

[37] Double precision General Matrix Multiply.

[38] Basic Linear Algebra Subprograms, a large collection of numerical routines which were designed to take advantage of the cache structure of modern computers. Using BLAS routines versus naive loops for contractions leads to speedups often of a factor of 4 or more, even when using aggressively optimizing compilers.

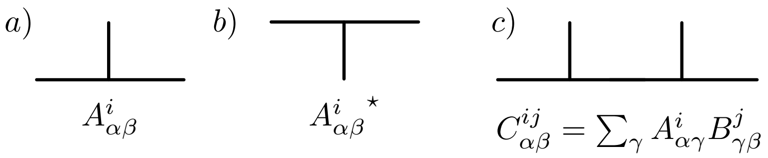


Figure 6.2: Examples of basic tensor operations in diagrammatic notation. a) A rank-3 tensor. b) The conjugate of a rank-3 tensor. c) The contraction of two rank-3 tensors over a single index produces a rank-4 tensor.

At this stage, it is advantageous to develop a graphical notation for tensors and their operations [42]. A tensor is represented graphically by a point with lines extending upwards from it. The number of lines is equal to the rank of the tensor. The order of the indices from left to right is the same as the ordering of lines from left to right. A contraction of two tensors is represented by a line connecting two points. Finally, the complex conjugate of a tensor is denoted by a point with lines extending downwards. Some basic tensor operations are shown in graphical notation in Figure 6.2.

Following a similar line of reasoning as for contractions above, we may also *decompose* tensors into contractions of tensors using permutation, fusion, and any of the well-known matrix decompositions such as the singular value decomposition (SVD) or the QR decomposition. For example, a rank-3 tensor T can be factorized as

$$T_{ijk} = \sum_{l} U_{(ij)l} S_l V_{lk} \,, \tag{6.18}$$

where U and V are unitary and S is a positive semidefinite real vector. Such decompositions are of great use in enforcing canonical forms on MPSs, see Sec. 6.3.

A *tensor network* is now defined as a set of tensors whose indices are connected in a network pattern, see Figure 6.3. Let us consider that some set of the network's indices $\mathbf{c}$ are contracted over, and the complement $\bar{\mathbf{c}}$ remain uncontracted. Then, this network is a decomposition of some tensor $T_{\bar{\mathbf{c}}}$. The basic idea of tensor network algorithms utilizing MPSs and their higher dimensional generalizations such as projected

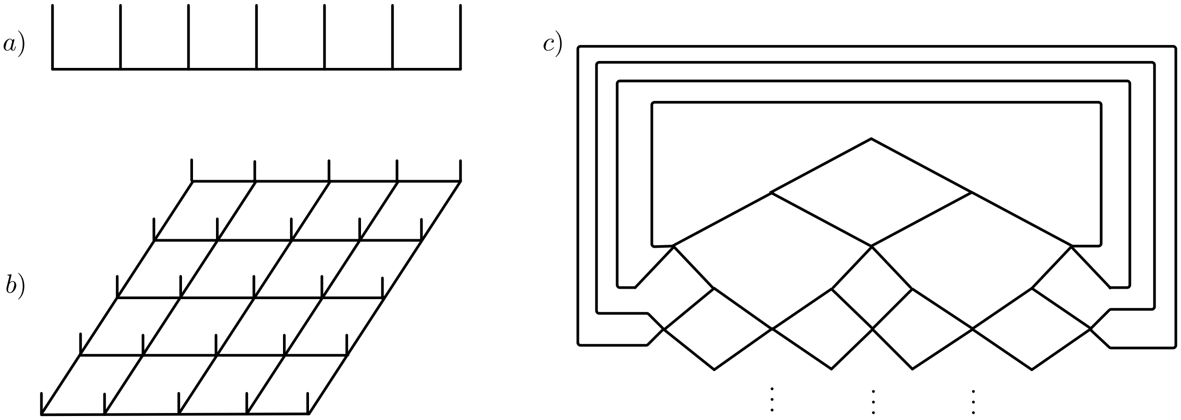


Figure 6.3: Examples of tensor networks. a) An MPS with 7 sites and open boundary conditions. b) A square lattice PEPS with 25 sites and open boundary conditions. c) Two levels of a 1D MERA with periodic boundary conditions. This network consists of alternating rows of rank-3 and rank-4 tensors.

entangled-pair states (PEPS) [43, 44] and the multiscale entanglement renormalization algorithm (MERA) [45, 46] are to represent the high-rank tensor $c_{i_1 \dots i_L}$ encoding a many-body wavefunction in a Fock basis,

$$|\psi\rangle = \sum_{i_1 \dots i_L} c_{i_1 \dots i_L} |i_1 \dots i_L\rangle \, , \tag{6.19}$$

as a tensor network with tensors of small rank. We set the convention in the remainder of this chapter that indices which are contracted over in the tensor network decomposition will be denoted by Greek indices, and indices which are left uncontracted will be denoted by Roman indices. The former type of index will be referred to as a bond index, and the latter as a physical index.

In particular, an MPS imposes a one-dimensional topology on the tensor network such that all the tensors appearing in the decomposition are rank-3. The resulting decomposition has the structure shown in Figure 6.3(a). Explicitly, an MPS may be written in the form

$$|\psi_{\mathrm{MPS}}\rangle=\sum_{i_1,\dots i_L=1}^{d}\mathrm{Tr}\left(A^{[1]i_1}\dots A^{[L]i_L}\right)|i_1\dots i_L\rangle\,. \tag{6.20}$$

Here, $i_1 \dots i_L$ label the L distinct sites, each of which contains a d-dimensional Hilbert space. We will call d the *local dimension*. The superscript index in brackets $[j]$ denotes that this is the tensor of the j^{th} site, as these tensors are not all the same in general. Finally, the trace effectively sums over the first and last dimensions of $A^{[1]i_1}$ and $A^{[L]i_L}$ concurrently, and is necessary only for periodic boundary conditions where these dimensions are greater than 1. Obscured within the matrix product of Eq. (6.20) is the size of the matrix $A^{[j]i_j}$ formed from the tensor $A^{[j]}$ with its physical index held constant. We will refer to the left and right dimensions of this matrix as χ_j and χ_{j+1}, and the maximum value of χ_j for any tensor, the *bond dimension*, will be denoted as χ. The bond dimension is the parameter which determines the efficiency of an MPS simulation, and also its dominant computational scaling. From the relation $S_{\mathrm{vN}} \leq \log\chi$, we also have that χ represents an entanglement cutoff for MPSs. The scaling of variational MPS algorithms is discussed in Chapter 7.

In what follows, we consider MPSs with open boundary conditions (OBC) unless explicitly indicated otherwise. Details on algorithms for periodic boundary conditions (PBC) are provided in Chapter 9. The reasoning for using OBC is twofold. First, finding the normalized eigenvector corresponding to the minimum eigenvalue of a Hermitian operator $\hat{Q}$ may be expressed as the minimization of the Rayleigh quotient [2]

$$\min_{|\psi\rangle}\langle\psi|\hat{Q}|\psi\rangle/\langle\psi|\psi\rangle\,. \tag{6.21}$$

In variational MPS algorithms this quotient is minimized locally by holding all tensors except for some subnetwork $\mathcal{A}$ fixed and minimizing the Rayleigh quotient with respect to the parameters of $\mathcal{A}$. The minimization problem for the subnetwork $\mathcal{A}$

becomes a generalized eigenvalue problem

$$\mathcal{Q}_{\mathrm{eff}}\mathcal{A} = \lambda\mathcal{N}\mathcal{A}\,, \tag{6.22}$$

where $\mathcal{Q}_{\mathrm{eff}}$ is the action of the operator $\hat{Q}$ on the subnetwork to be optimized with the rest of the network held fixed, and $\mathcal{N}$ is the action of the identity on the subnetwork to be optimized with the rest of the network held fixed. There exists a canonical form for MPSs with OBC such that $\mathcal{N}$ is the identity, and the generalized eigenvalue problem becomes an eigenvalue problem. For PBC this operator cannot generally be made the identity, and may be singular or poorly conditioned, leading to numerical instabilities and a reduction in accuracy.

The second reason for avoiding PBC is deeper. When we divide a system with PBC into two contiguous regions, one of length L and one of length ℓ, generally the two ends of subsystem L which surround that of subsystem ℓ are correlated. This is to be contrasted with OBC, where an arbitrary partition creates an environment for ℓ which is uncorrelated. This idea is demonstrated in Figure 6.4. A consequence of this is that the entanglement entropy of a critical system with PBC grows twice as fast as one with OBC, that is

$$S_{\mathrm{PBC}}\left(A\right) \sim \frac{c}{3}\log L_A\,. \tag{6.23}$$

This should be compared with Eq. (6.10) for OBC. Hence, the corresponding bond dimension for PBC grows as the square of the bond dimension for OBC. This is sometimes referred to as $\mathcal{O}\left(\chi^6\right)$ scaling, as introducing a fictitious OBC system with a long range interaction between the first and last sites will require a bond dimension χ_{PBC} that scales as the square of the bond dimension χ_{OBC} in the absence of the long-range interaction to represent the system with the same level of accuracy. That is, the system with the long range interaction will require $\mathcal{O}\left(\chi^3_{\mathrm{PBC}}\right) = \mathcal{O}\left(\chi^6_{\mathrm{OBC}}\right)$ time for a fixed accuracy. This scaling can be brought down to $\mathcal{O}\left(\chi^5\right)$ by introducing an

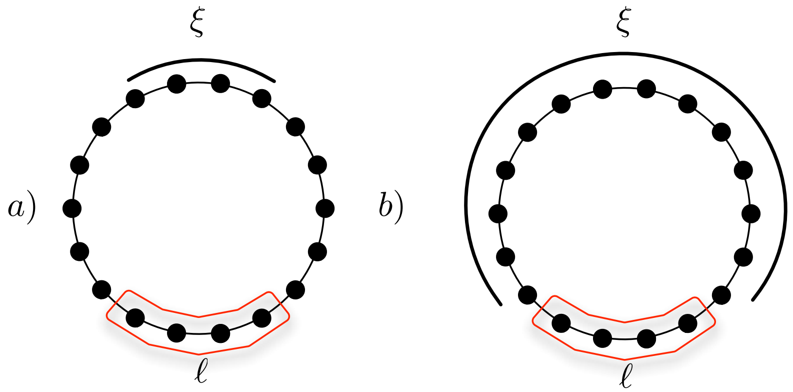


Figure 6.4: Correlation structure of PBC states. a) For a state with a correlation length $\xi \ll L$, the two ends of subregion L are weakly correlated and the entanglement of L with ℓ is constant, as in the OBC case. b) As the correlation length grows comparable to L, the two ends of subregion L are strongly correlated and give rise to twice as much entanglement between L and ℓ as in the case of OBC.

MPS which has the correct entanglement structure [32],[39] and recent efforts [47, 48] have shown how to reduce this scaling to $\mathcal{O}\left(p\chi^3\right)$, where p is the number of relevant correlation lengths of the transfer operator, see Sec. 6.5.

6.3 Canonical Forms for Matrix Product States

The matrix product structure of an MPS implies that we can insert any invertible matrix X and its inverse between any two matrices appearing in Eq. (6.20) without affecting the state $|\psi_{\mathrm{MPS}}\rangle$. Hence, the MPS decomposition is highly non-unique. We shall now describe canonical forms for MPSs which remove this non-uniqueness.

The first canonical form requires that a given tensor A satisfies

$$\sum_i {A^i}^\dagger A^i = I\,. \tag{6.24}$$

We say that a tensor satisfying Eq. (6.24) is in *left-canonical* form, or it is left-canonical for succinctness. We can enforce left-canonical form on a particular tensor via fusing the leftmost two indices and performing a singular value decomposition

---

[39] That is, the algorithm scales as $\mathcal{O}\left(\chi^5\right)$ but $\chi_{\mathrm{PBC}} = \chi_{\mathrm{OBC}}$ for a fixed error in the improved ansatz.

$$A_{(\alpha i)\beta} = A^{[\ell]i}_{\alpha\beta}\,, \tag{6.25}$$

$$\sum_\gamma U_{(\alpha i)\gamma} S_\gamma V^\dagger_{\gamma\beta} = A_{(\alpha i)\beta}\,. \tag{6.26}$$

Because the matrix U returned from the singular value decomposition is unitary, Eq. (6.24) is satisfied by making the replacement

$$A^{[\ell]i}_{\alpha\gamma} = U_{(\alpha i)\gamma}\,. \tag{6.27}$$

In order for the total state to remain unchanged, we must contract the remaining information returned from the singular value decomposition into the neighboring tensor as

$$A^{[\ell+1]i}_{\gamma\eta} = \sum_\beta S_\gamma V^\dagger_{\gamma\beta} A^{[\ell+1]i}_{\beta\eta}\,. \tag{6.28}$$

Hence, the procedure of bringing a tensor into canonical form affects two tensors at a time. By carrying out this recursion all the way from the left (open) boundary to the right boundary, we end up with an MPS in which each tensor satisfies Eq. (6.24) and a 1 × 1 matrix whose trace is the norm of the MPS. Such an MPS is said to be in left-canonical form.

One can imagine performing the recursion Eq. (6.25)-(6.28) in the opposite direction, instead replacing A with the unitary matrix V as

$$A_{\alpha(i\beta)} = A^{[\ell]i}_{\alpha\beta}\,, \tag{6.29}$$

$$\sum_\gamma U_{\alpha\gamma} S_\gamma V^\dagger_{\gamma(i\beta)} = A_{\alpha(i\beta)}\,, \tag{6.30}$$

$$A^{[\ell]i}_{\gamma(i\beta)} = V^\dagger_{\gamma(i\beta)}\,, \tag{6.31}$$

$$A^{[\ell-1]i}_{\eta\gamma} = \sum_\alpha A^{[\ell-1]i}_{\eta\alpha} U_{\alpha\gamma} S_\gamma\,. \tag{6.32}$$

The resulting tensor $A = A^{[\ell]}$ satisfies the *right-canonical* condition

$$\sum_i A^i {A^i}^\dagger = I\,. \tag{6.33}$$

As with the left-canonical form, one can perform the recursions Eq. (6.29)-(6.32) from the right boundary to the left boundary to obtain a representation in which all tensors satisfy Eq. (6.33) together with a 1×1 matrix whose trace is the norm of the MPS. In the remainder of this chapter, we will reserve the notation A for a left-canonical tensor and B for a right-canonical tensor. The left-canonical and right-canonical MPS forms thus read

$$|\psi_{\mathrm{left-canonical}}\rangle = \sum_{i_1,\dots i_L=1}^{d} \mathrm{Tr}\left(A^{[1]i_1}\dots A^{[L]i_L}\right)|i_1\dots i_L\rangle\,, \tag{6.34}$$

$$|\psi_{\mathrm{right-canonical}}\rangle = \sum_{i_1,\dots i_L=1}^{d} \mathrm{Tr}\left(B^{[1]i_1}\dots B^{[L]i_L}\right)|i_1\dots i_L\rangle\,. \tag{6.35}$$

The consequences of left and right canonical form in graphical notation are provided as part of the publication in Chapter 7.

We now consider taking a state $|\psi\rangle$ and performing Eq. (6.25)-(6.28) to put all tensors $A^{[j]}$ from $j = 1, \dots, k$ into left-canonical form and performing Eq. (6.29)-(6.32) such that all tensors $B^{[j]}$ from $j = k+1, \dots, L$ are right-canonical. In order for the state to be consistent, we must insert a matrix M between $A^{[k]}$ and $B^{[k+1]}$ which represents $S^{[k]}{V^{[k]}}^\dagger$ from the right-moving contraction Eq. (6.28) multiplied by $U^{[k+1]}S^{[k+1]}$ from the left-moving contraction Eq. (6.32). That is, the state is

$$|\psi\rangle = \sum_{i_1,\dots i_L=1}^{d} \mathrm{Tr}\left(A^{[1]i_1}\dots A^{[k]i_k} M B^{[k+1]i_{k+1}}\dots B^{[L]i_L}\right)|i_1\dots i_L\rangle\,, \tag{6.36}$$

where $M = S^{[k]}{V^{[k]}}^\dagger U^{[k+1]}S^{[k+1]}$. Performing a singular value decomposition $M = U\Lambda V$ and absorbing U into $A^{[k]}$ and V into $B^{[k+1]}$,[40] we may write this as

---

[40] Note that this does not affect the canonical form of these two tensors.

$$|\psi\rangle = \sum_{i_1,\dots i_L=1}^{d} \mathrm{Tr}\left(A^{[1]i_1}\dots A^{[k]i_k}\Lambda B^{[k+1]i_{k+1}}\dots B^{[L]i_L}\right)|i_1\dots i_L\rangle\,, \tag{6.37}$$

where now Λ is a diagonal matrix. This form of the state corresponds identically with the Schmidt decomposition mentioned in Sec. 6.1. As the matrix Λ lies at the boundary between the left-orthogonal and right-orthogonal parts of the state, we will call it the *orthogonality center*. To be more precise, we call it the *bond-centered orthogonality center* as it resides on the bond between two sites. The state Eq. (6.37) is said to be in *bond-centered mixed canonical form*. We may absorb Λ into $A^{[k]}$ to give a tensor which is no longer left-canonical, but does carry all the information that was contained in the orthogonality center. We will denote this tensor with a tilde to indicate its lack of left-canonical form:

$$|\psi\rangle = \sum_{i_1,\dots i_L=1}^{d} \mathrm{Tr}\left(A^{[1]i_1}\dots \tilde{A}^{[k]i_k}B^{[k+1]i_{k+1}}\dots B^{[L]i_L}\right)|i_1\dots i_L\rangle\,. \tag{6.38}$$

We now say that $\tilde{A}^{[k]}$ is the orthogonality center of the MPS, more precisely the *site-centered orthogonality center*, and we call this particular canonical form *site-centered mixed canonical form*. When there is no chance of confusion, we will refer to both Eq. (6.37) and Eq. (6.38) as mixed canonical form. The most important feature of the orthogonality center is that it expresses the wavefunction in terms of a tensor whose indices run over orthonormal bases.

The final canonical form we consider takes the mixed canonical form to its logical conclusion by ensuring that a bipartite splitting at any bond within the lattice results in the Schmidt decomposition, and we shall call it the *Vidal canonical form* [49, 50]:

$$|\psi_{\mathrm{VCF}}\rangle = \sum_{\alpha_1\dots\alpha_{L+1}}\sum_{i_1\dots i_L} \lambda^{[1]}_{\alpha_1}\Gamma^{[1]i_1}_{\alpha_1\alpha_2}\lambda^{[2]}_{\alpha_2}\dots\lambda^{[L]}_{\alpha_L}\Gamma^{[L]i_L}_{\alpha_L\alpha_{L+1}}\lambda^{[L+1]}_{\alpha_{L+1}}\,. \tag{6.39}$$

The Γ tensors and λ tensors are chosen such that a bipartite splitting of our system at the bond between sites l and $l+1$ is exactly the Schmidt decomposition

$$|\psi\rangle = \sum_{\alpha_l=1}^{\chi_S} \lambda_{\alpha_l}^{[l+1]} |\phi_{\alpha_l}^{[1\dots l]}\rangle |\phi_{\alpha_l}^{[l+1\dots n]}\rangle \,, \tag{6.40}$$

with the Schmidt vectors

$$|\phi_{\alpha_l}^{[1\dots l]}\rangle = \sum_{\alpha_0,\dots,\alpha_{l-1}}^{\chi_S} \lambda_{\alpha_0}^{[1]} \Gamma_{\alpha_0\alpha_1}^{[1]i_1} \lambda_{\alpha_1}^{[2]} \Gamma_{\alpha_1\alpha_2}^{[2]i_2} \lambda_{\alpha_2}^{[3]} \Gamma_{\alpha_2\alpha_3}^{[3]i_3} \dots \Gamma_{\alpha_{l-1}\alpha_l}^{[l]i_l} |i_1\rangle \dots |i_l\rangle \tag{6.41}$$

and

$$|\phi_{\alpha_l}^{[l+1\dots L]}\rangle = \sum_{\alpha_{l+1},\dots,\alpha_L}^{\chi_S} \Gamma_{\alpha_l\alpha_{l+1}}^{[l+1]i_{l+1}} \lambda_{\alpha_{l+2}}^{[l+1]} \Gamma_{\alpha_{l+1}\alpha_{l+2}}^{[l+2]i_{l+2}} \dots \Gamma_{\alpha_{L-1}\alpha_L}^{[L]i_L} \lambda_{\alpha_L}^{[L]} |i_{l+1}\rangle \dots |i_L\rangle \,, \tag{6.42}$$

and the Schmidt coefficients $\lambda_{\alpha_l}^{[l+1]}$. We can translate between the left, right, and Vidal canonical forms by using the translation formulae

$$A_{\alpha\beta}^{[j]i_j} = \lambda_{\alpha}^{[j]} \Gamma_{\alpha\beta}^{[j]i_j} \,, \tag{6.43}$$

$$B_{\alpha\beta}^{[j]i_j} = \Gamma_{\alpha\beta}^{[j]i_j} \lambda_{\beta}^{[j+1]} \,. \tag{6.44}$$

These will be useful in Chapter 8, when we discuss translationally invariant MPSs.

As an example of why the various MPS canonical forms are useful, we now consider finding the expectation value of an observable $\hat{O}_k$ which acts only on a single lattice site k. Assuming that site k is the orthogonality center, this expectation value is

$$\begin{aligned}
\langle\psi|\hat{O}_k|\psi\rangle &= \sum_{i_1\dots i_k\dots i_L} \sum_{i'_k} O_k^{i_k i'_k} \mathrm{Tr}\left(A^{[1]i_1} \dots \tilde{A}^{[j]i'_k} \dots B^{[L]i_L} B^{[L]i_L\,\dagger} \dots \tilde{A}^{[k]i_k\,\dagger} \dots A^{[1]i_1\,\dagger}\right) \,, \\
&= \sum_{i_1\dots i_k\dots i_L} \sum_{i'_k} O_k^{i_k i'_k} \mathrm{Tr}\left(A^{[1]i_1\,\dagger} A^{[1]i_1} \dots \tilde{A}^{[k]i'_k} \dots B^{[L]i_L} B^{[L]i_L\,\dagger} \dots \tilde{A}^{[k]i_k\,\dagger} \dots\right) \,, \\
&= \sum_{i_2\dots i_k\dots i_{L-1}} \sum_{i'_k} O_k^{i_k i'_k} \mathrm{Tr}\left(A^{[2]i_2\,\dagger} A^{[2]i_2} \dots \tilde{A}^{[k]i'_k} \dots B^{[L-1]i_{L-1}} B^{[L-1]i_{L-1}\,\dagger} \dots \tilde{A}^{[k]i_k\,\dagger} \dots\right) \,,
\end{aligned}$$

$$= \sum_{i_k i'_k} O_k^{i_k i'_k} \mathrm{Tr}\left(\tilde{A}^{[k]i'_k} \tilde{A}^{[k]i_k\dagger}\right) . \tag{6.45}$$

Here, the second line used the cyclic nature of the trace and the third and fourth lines used the left and right-canonical conditions Eqs. (6.24) and (6.33) for the tensors to the left and right of k, respectively. Thus, mixed canonical form projects most operations involving the entire wavefunction into operations involving only the orthogonality center.

6.4 Examples of Matrix Product States

Several important and well-known states can be cast exactly as MPSs with constant bond dimension. The purpose of this section is to provide some examples of states with exact MPS representations.

6.4.1 Product State

The simplest MPS is a product state $|k_1 k_2 \dots k_L\rangle$, which is a product of 1×1 matrices $A^{[j]i_j} = \left(\delta_{i_j k_j}\right)$. For the Vidal canonical form, all of the λ tensors are (1) and the Γ tensors are represented by the A tensors here. In this case, $\chi = 1$ for any subsystem. This is the only class of states in which all of the canonical forms above have identical MPS representations.

6.4.2 GHZ State

A nontrivial MPS with bond dimension two is the (unnormalized) Greenberger-Horne-Zeilinger (GHZ) state

$$|\mathrm{GHZ}\rangle = |00\dots 0\rangle + |11\dots 1\rangle . \tag{6.46}$$

This state represents a realization of Schrödinger's famous cat paradox [51] in which a quantum system exists in two very different macroscopic states simultaneously.

Also, the GHZ state is closely related to *NOON states* in which N particles exist in a superposition of all particles in state a and all particles in state b simultaneously. The GHZ state garnered interest because of its very strong non-classical correlations [52]. The state is given in left or right canonical form by the matrices

$$A^i = \begin{pmatrix} \delta_{i,0} & 0 \\ 0 & \delta_{i,1} \end{pmatrix} . \tag{6.47}$$

Here we have neglected the boundary conditions. For PBC all matrices are equivalent, and for OBC the first and last sites are 1×2 and 2×1 matrices whose elements are the diagonal elements of A^i, respectively. The normalized GHZ state can be represented in mixed canonical form by introducing the orthogonality center

$$A^i = \begin{pmatrix} \delta_{i,0}/\sqrt{2} & 0 \\ 0 & \delta_{i,1}/\sqrt{2} \end{pmatrix} . \tag{6.48}$$

In the Vidal canonical form, the normalized GHZ state takes the form

$$\lambda^{[j]} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} , \Gamma^{[j]i} = \begin{pmatrix} \delta_{i,0}\sqrt{2} & 0 \\ 0 & \delta_{i,1}\sqrt{2} \end{pmatrix} , \tag{6.49}$$

for $2 \leq j \leq L-1$ together with the boundaries

$$\lambda^{[1]} = \lambda^{[L+1]} = (1) \ , \ \Gamma^{[1]i} = \begin{pmatrix} \delta_{i,0} & \delta_{i,1} \end{pmatrix} \ , \ \Gamma^{[L]i} = \begin{pmatrix} \delta_{i,0} \\ \delta_{i,1} \end{pmatrix} . \tag{6.50}$$

As this is a state with $\chi = 2$, it has minimally nontrivial spatial entanglement.

6.4.3 W State

The W state is the equal superposition of all translates of $|10\dots0\rangle$,

$$|W\rangle = \frac{1}{\sqrt{L}} \sum_{i=1}^{L} |0\rangle^{\otimes i-1} |1\rangle |0\rangle^{\otimes L-i} , \tag{6.51}$$

and can also be represented as an MPS with bond dimension 2. If we interpret $|0\rangle$ as being spin down and $|1\rangle$ as being spin up, the W state with $L=2$ constituents is the spin triplet. If $|0\rangle$ and $|1\rangle$ are interpreted as lattice sites containing 0 and 1 particles, respectively, the W state represents the state of a single particle on a ring with quasimomentum $q=0$. The Vidal canonical form is

$$\lambda^{[j]}=\left(\begin{array}{c}\sqrt{\frac{L-j+1}{L}}\\ \sqrt{\frac{j-1}{L}}\end{array}\right),\Gamma^{[j]i}=\left(\begin{array}{cc}\delta_{i,0}\sqrt{\frac{L}{L-j+1}} & \delta_{i,1}\sqrt{\frac{L}{j(L-j+1)}}\\ \delta_{i,1}\sqrt{\frac{L}{j}} & 0\end{array}\right), \tag{6.52}$$

for $2\le j\le L-1$ together with the same boundary tensors as the GHZ state. In the MPS representation we can choose

$$A^{[1]i}=\left(\begin{array}{c}\delta_{i,0}/\sqrt{L}\\ \delta_{i,1}/\sqrt{L}\end{array}\right),\ A^{[j]i}=\left(\begin{array}{cc}\delta_{i,0} & 0\\ \delta_{i,1} & \delta_{i,0}\end{array}\right),\ A^{[L]i}=\left(\begin{array}{cc}\delta_{i,0} & \delta_{i,1}\end{array}\right), \tag{6.53}$$

which amounts to mixed canonical form with the first site being the orthogonality center.

6.4.4 AKLT State

Affleck, Kennedy, Lieb, and Tasaki (AKLT) considered the following Hamiltonian [53]

$$\hat{H}_{\mathrm{AKLT}}=\sum_i\hat{\mathbf{S}}_i\cdot\hat{\mathbf{S}}_{i+1}+\frac{1}{3}\left(\hat{\mathbf{S}}_i\cdot\hat{\mathbf{S}}_{i+1}\right)^2. \tag{6.54}$$

Here, $\hat{\mathbf{S}}_i$ is the vector of three-component spin operators at site i. We will call a three-component quantum system, isomorphic to the internal space of a particle with a spin of 1, a *qutrit*. Similarly, a two-component quantum system will be referred to as a *qubit*. By defining a projector onto the Hilbert space of total spin equal to two as

$$\hat{P}=\frac{1}{6\cdot 4}\left(\hat{\mathbf{S}}^2_{\mathrm{total}}-2\right)\hat{\mathbf{S}}^2_{\mathrm{total}}, \tag{6.55}$$

and setting $\hat{\mathbf{S}}_{\text{total}} = \hat{\mathbf{S}}_i + \hat{\mathbf{S}}_{i+1}$, we find that the projector of the spin on bond i onto the spin-2 subspace is

$$\hat{P}_i = \frac{1}{6 \cdot 4}\left(\hat{\mathbf{S}}_i^2 + \hat{\mathbf{S}}_{i+1}^2 + 2\hat{\mathbf{S}}_i \cdot \hat{\mathbf{S}}_{i+1} - 2\right)\left(\hat{\mathbf{S}}_i^2 + \hat{\mathbf{S}}_{i+1}^2 + 2\hat{\mathbf{S}}_i \cdot \hat{\mathbf{S}}_{i+1}\right) \tag{6.56}$$

$$= \frac{1}{2}\left[\frac{2}{3} + \hat{\mathbf{S}}_i \cdot \hat{\mathbf{S}}_{i+1} + \frac{1}{3}\left(\hat{\mathbf{S}}_i \cdot \hat{\mathbf{S}}_{i+1}\right)^2\right] . \tag{6.57}$$

Thus,

$$\hat{H}_{\text{AKLT}} = \sum_i \left(2\hat{P}_i - \frac{2}{3}\hat{I}\right) . \tag{6.58}$$

Let us now consider each qutrit to be comprised of two qubits with internal states $\{|\uparrow\rangle, |\downarrow\rangle\}$. For this to be consistent, we must require these two qubits to be completely symmetrized so as to lie in the subspace with total spin equal to 1. Now, consider the many-body state where adjacent pairs of these qubits not forming a qutrit are joined in a singlet state such that the bonds have spin zero. The projector $\hat{P}_i$ acting on this state gives zero, and from $\langle \hat{P}_i \rangle \geq 0$ we have that the given construction produces a ground state of the AKLT Hamiltonian. It can also be shown that this state is unique [53]. We will call the state formed from the above construction the *AKLT state*.

We can write the AKLT state as an MPS by considering the chain of qubits which has length $2L$, with L being the length of the original qutrit chain. We define the qubit sites a_i and b_i, $i = 1, \ldots L$, such that the bonds $a_i b_i$ are connected in a symmetric fashion and $b_i a_{i+1}$ are connected in an antisymmetric fashion. We will also consider periodic boundaries so as to easily facilitate taking the limit of an infinite chain. A matrix encapsulating the state of the singlet bonds, $(|\uparrow\downarrow\rangle - |\downarrow\uparrow\rangle)/\sqrt{2}$, is

$$S = \begin{pmatrix} 0 & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & 0 \end{pmatrix} . \tag{6.59}$$

Similarly, matrices representing the three triplet bonds $|1\rangle = |\uparrow\uparrow\rangle$, $|0\rangle = (|\uparrow\downarrow\rangle + |\downarrow\uparrow\rangle)/\sqrt{2}$, and $|-1\rangle = |\downarrow\downarrow\rangle$, are

$$T^i = \begin{pmatrix} \delta_{i,1} & \frac{1}{\sqrt{2}}\delta_{i,0} \\ \frac{1}{\sqrt{2}}\delta_{i,0} & \delta_{i,-1} \end{pmatrix} . \tag{6.60}$$

Hence, the state of the qutrit chain may be parameterized as

$$|\psi\rangle = \sum_{i_1 a_1 b_1 \ldots i_L a_L b_L} T^{i_1}_{a_1 b_1} S_{b_1 a_2} T^{i_2}_{a_2 b_2} S_{b_2 a_3} \ldots S_{b_{L-1} a_L} T^{i_L}_{a_L b_L} S_{b_L a_1} |i_1 \ldots i_L\rangle , \tag{6.61}$$

$$= \sum_{i_1 \ldots i_L} \mathrm{Tr}\left(A^{[1]i} \ldots A^{[L]i_L}\right) |i_1 \ldots i_L\rangle , \tag{6.62}$$

where

$$A^i = T^i S = \begin{pmatrix} -\frac{1}{2}\delta_{i,0} & \frac{1}{\sqrt{2}}\delta_{i,1} \\ -\frac{1}{\sqrt{2}}\delta_{i,-1} & \frac{1}{2}\delta_{i,0} \end{pmatrix} . \tag{6.63}$$

Noting that $\sum_i A^{i\dagger} A^i = \frac{3}{4} I$, we can normalize the state in the thermodynamic limit[41] by scaling by $2/\sqrt{3}$ to obtain

$$A^i = \begin{pmatrix} -\frac{1}{\sqrt{3}}\delta_{i,0} & \sqrt{\frac{2}{3}}\delta_{i,1} \\ -\sqrt{\frac{2}{3}}\delta_{i,-1} & \frac{1}{\sqrt{3}}\delta_{i,0} \end{pmatrix} . \tag{6.64}$$

We close this section by expressly pointing out the similarity between the AKLT construction and the general construction of MPSs as projections from maximally entangled pairs of χ-level subsystems onto physical states mentioned in Sec. 6.1. In fact, the similarity extends from the state construction to the Hamiltonian, as every MPS is the ground state of a *parent Hamiltonian* built of projectors which is gapped, is *frustration-free* in the sense that each term in the Hamiltonian minimizes the energy locally, is k-local with $k \sim 2\log\chi/\log d$, and allows for a detailed analysis of the ground state degeneracy [54, 55].

---

[41] See Eq. (6.69) for the condition that a left-canonical MPS on an infinite lattice is normalized.

6.5 Correlations within Matrix Product States and the Transfer Operator

Let us now turn to the structure of correlations within MPSs. By a correlation, we mean a two-point correlation function, for example the density-density correlation function $\langle \hat{n}_p \hat{n}_q \rangle$. Assuming that the state is in mixed canonical form with the orthogonality center k satisfying$p \leq k \leq q$, the correlation becomes a finite tensor network contraction between sites p and q. Written out explicitly, we have

$$\begin{aligned}
&\langle \hat{O}_p \hat{O}_q \rangle && (6.65)\\
&= \mathrm{Tr}\left(\sum_{i_p i'_p} \hat{O}_{i_p i'_p} A^{[p]i_p\star} \otimes A^{[p]i'_p} \prod_{j=p+1}^{q-1} \left(\sum_{i_j} A^{[j]i_j\star} \otimes A^{[j]i_j} \right) \sum_{i_q i'_q} \hat{O}_{i_q i'_q} A^{[q]i_q\star} \otimes A^{[q]i'_q} \right) ,\\
&= \mathrm{Tr}\left(E^{[p]}_{\hat{O}} \prod_{j=p+1}^{q-1} E^{[j]}_{\hat{I}} E^{[q]}_{\hat{O}} \right) , && (6.66)
\end{aligned}$$

where we have defined the transfer operator

$$E_{\hat{O}} = \sum_{ii'} \hat{O}_{ii'} A^{i\star} \otimes A^{i'} . \tag{6.67}$$

We will also use the alternate notation

$$E_{\hat{O}}\left[M \right] = \sum_{ii'} A^{i\dagger} \hat{O}_{ii'} M A^{i'} , \tag{6.68}$$

which expresses the action of the transfer operator on a matrix M [1]. The difference in notation is demonstrated by comparing the expectation Eq. (6.65) to Eq. (6.45). The transfer operator $E_{\hat{O}}$ either acts upon length χ^2 vectors or takes $\chi \times \chi$ matrices to $\chi \times \chi$ matrices,[42] depending on which interpretation we use.

While the transfer operator is not generally symmetric, we can still venture to find its eigenvalues and (left and right) eigenvectors/eigenmatrices. It can be shown that the transfer operator of a normalized state has a spectral radius of 1 [1]. For

---

[42] Strictly speaking, the transfer operator using the MPS tensors at site j take $\chi_j \times \chi_j$ matrices to $\chi_{j-1} \times \chi_{j-1}$ matrices with the given order of operations, but the great numerical use is in infinite systems where χ is uniform across all bonds.

matrices which are left-canonical, we have that

$$E_{\hat{I}}\left[\hat{I}\right] = \sum_i {A^i}^\dagger A^i = \hat{I}\,, \tag{6.69}$$

and so the left eigenmatrix of $E_{\hat{I}}$ with eigenvalue 1 is the identity matrix. Similarly, an appropriately defined transfer operator of right-canonical MPS matrices has the identity matrix as a right eigenmatrix with eigenvalue 1. For simplicity, let us now consider a state which is translationally invariant such that all matrices appearing in the contraction Eq. (6.65) are identical, and let $E_{\hat{I}}$ have a non-degenerate maximal eigenvalue 1. Expanding the product of transfer operators $\prod_{j=p+1}^{q-1} E_{\hat{I}}^{[j]}$ in terms of its eigenspectrum, we have that

$$\langle \hat{O}_p \hat{O}_q \rangle = \sum_k \langle 1|E_{\hat{O}}^{[p]}|k\rangle \lambda_k^{q-p-1} \langle k|E_{\hat{O}}^{[q]}|1\rangle\,, \tag{6.70}$$

where $|k\rangle$ and $\langle k|$ are the right and left eigenvectors corresponding to eigenvalue λ_k. There are now two possibilities. The first is that this correlation function is long ranged, which occurs when $\langle 1|E_{\hat{O}}^{[p]}|1\rangle\langle 1|E_{\hat{O}}^{[q]}|1\rangle$ is nonzero, and the second is a superposition of exponential decays with decay lengths $\xi_k = -1/\log\lambda_k$. This may be written compactly as

$$\frac{\langle\psi|\hat{O}^p\hat{O}^q|\psi\rangle}{\langle\psi|\psi\rangle} = \langle 1|E_{\hat{O}}^{[p]}|1\rangle\langle 1|E_{\hat{O}}^{[q]}|1\rangle + \sum_{k=2}^{\chi^2} \langle 1|E_{\hat{O}}^{[p]}|k\rangle\langle k|E_{\hat{O}}^{[q]}|1\rangle e^{-|q-p-1|/\xi_k}\,. \tag{6.71}$$

To illustrate these ideas, we can use the AKLT state Eq. (6.64) from above. The transfer operator is

$$E_{\hat{I}} = \sum_i A^i \otimes A^i = \begin{pmatrix} \frac{1}{3} & 0 & 0 & \frac{2}{3} \\ 0 & -\frac{1}{3} & 0 & 0 \\ 0 & 0 & -\frac{1}{3} & 0 \\ \frac{2}{3} & 0 & 0 & \frac{1}{3} \end{pmatrix}\,, \tag{6.72}$$

which has eigenvalues 1 and $-1/3$, the latter being triply degenerate. The dominant eigenmatrix is $\hat{I}$, as discussed for left-canonical matrices above. The other eigenspace is spanned by $\hat{\sigma}_z$, $\hat{\sigma}_+$, and $\hat{\sigma}_-$, the spin-1/2 Pauli matrices. Using the transfer operators

$$E_{\hat{S}_z} = \left(\begin{array}{cccc} 0 & 0 & 0 & \frac{2}{3} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{2}{3} & 0 & 0 & 0 \end{array}\right) , \tag{6.73}$$

$$E_{\exp\left(i\pi\hat{S}_z\right)} = \left(\begin{array}{cccc} \frac{1}{3} & 0 & 0 & -\frac{2}{3} \\ 0 & -\frac{1}{3} & 0 & 0 \\ 0 & 0 & -\frac{1}{3} & 0 \\ -\frac{2}{3} & 0 & 0 & \frac{1}{3} \end{array}\right) , \tag{6.74}$$

we find exponential decay of antiferromagnetic correlations, $\langle \hat{S}_z^i \hat{S}_z^j \rangle \sim (-1/3)^{i-j}$, but long-range order in the string order parameter $\langle \hat{S}_z^i \prod_{k=i+1}^{j-1} \exp\left(i\pi \hat{S}_z^k\right) \hat{S}_z^j \rangle = -4/9$. Here string order refers to the fact that, although true long range antiferromagnetic order of the classical Néel type $|\dots 1, -1, 1, -1 \dots\rangle$ is absent, any site with $S^z = \pm 1$ is followed by a site with the opposite spin projection $S^z = \mp 1$, and these two sites are connected by a string of $S^z = 0$ sites which can have arbitrarily long length.

Given that an MPS has correlations which decay exponentially asymptotically, how can they accurately represent a critical state which displays power-law decay of some correlator? The answer is that the general correlation structure of MPSs is a sum of exponentials, and so on length scales short compared to the dominant correlation length this sum can approximate an algebraic decay. This idea is demonstrated in Figure 6.5, where approximations to the function $1/r^3$ are provided by fitting a sum of exponentials to this function. The accuracy of the fit extends to longer distances as more exponentials are included. The minimization procedure used to fit the exponentials is precisely that used to define matrix product operators with long-range interactions, see Sec. 7.2.2.

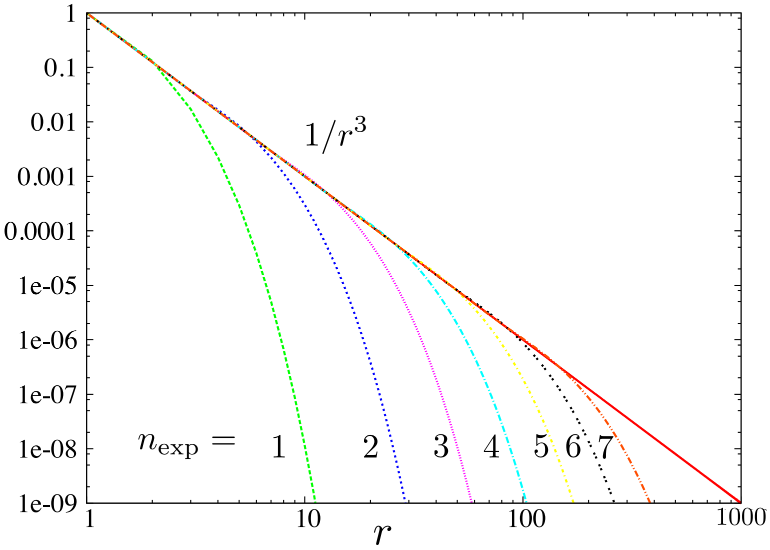


Figure 6.5: Approximating algebraic decay by sums of exponentials. The solid red line represents the function $1/r^3$, and the other lines represent approximations to $1/r^3$ obtained by a least squares minimization of a sum of exponentials with the indicated number of terms.

Given that the accuracy of an MPS representation of a state monotonically increases with the bond dimension and the correlation length of a critical state diverges, we find that the correlation length must increase with χ. In fact, χ obeys a scaling relationship with the correlation length [56]

$$\chi \sim \xi^{\kappa}\,, \tag{6.75}$$

where

$$\kappa = \frac{6}{\sqrt{12c}+c}\,. \tag{6.76}$$

Here c is the central charge of the conformal field theory describing criticality, see Eq. (6.10) and the surrounding discussion. Results such as Eq. (6.75) fall under the heading of *finite-entanglement scaling*, as contrasted with finite-size scaling [57]. A nice feature of finite-entanglement scaling is that scaling relationships depend only on truly universal quantities such as the central charge rather than scaling dimensions as in finite-size scaling. Provided we are in the finite-entanglement scaling regime rather than the finite-size scaling regime [58], for example because we have taken the size of the system much larger than the correlation length,[43] finite-entanglement scaling provides us with a powerful means of extracting universal quantities for relatively small amounts of computational effort, see Sec. 8.4.

6.6 Symmetry-Adapted Matrix Product States

A particularly important numerical optimization for finite size MPS algorithms is the explicit conservation of symmetries [59–65]. The theory developed below extends readily to compact, completely reducible groups $\mathcal{G}$, which includes finite groups such as the cyclic groups $\mathbb{Z}_q$ and Lie groups such as SO(n), U(n) and SU(n). To keep the notation simple, we develop the theory only for Abelian groups, taking U(1) as a particular example.

For compact, completely reducible symmetry groups $\mathcal{G}$, there exists a unitary representation $\hat{U} : \mathcal{G} \to \mathbb{H}$ of $\mathcal{G}$ on the space $\mathbb{H}$ of a single site such that for each element $g \in \mathcal{G}$, we have that $\hat{U}_g$ is unitary, $\hat{U}_g\hat{U}_g^{\dagger} = I$ and $\hat{U}_{gg'} = \hat{U}_g\hat{U}_{g'}$ [66]. For U(1) the elements of the group can be labeled by an angle $\phi \in [0, 2\pi)$ such that the unitary representations $\hat{U}_\phi$ satisfy

$$\hat{U}_\phi^{\dagger}\hat{U}_\phi = \hat{U}_\phi\hat{U}_\phi^{\dagger} = \hat{I}\,, \tag{6.77}$$

$$\hat{U}_{\phi_1}\hat{U}_{\phi_2} = \hat{U}_{\phi_2}\hat{U}_{\phi_1} = \hat{U}_{\phi_3}\,, \tag{6.78}$$

where $\phi_3 = \phi_1 + \phi_2 \mod 2\pi$. $\mathbb{H}$ now decomposes into possibly degenerate one-dimensional[44] irreducible representations (*irreps*) of $\mathcal{G}$ as

$$\mathbb{H}_j = \oplus_q \mathbb{H}_q\,, \tag{6.79}$$

---

[43] For critical systems, the diverging of the correlation length requires us to consider systems with an infinite number of sites. For an MPS algorithm which operates in this limit, see Chapter 8.

[44] These irreps are only one-dimensional in the Abelian case. In the non-Abelian case it is still true that the space decomposes into degeneracy spaces and irreps; however, the irreps can have dimensions larger than one.

where the d_q-dimensional subspaces $\mathbb{H}_q$ are labeled by an integer q which we call the *charge* of the irrep. Here d_q, the *degeneracy dimension*, denotes the number of copies of the irrep of charge q which are present in $\mathbb{H}$. Because the irreps of Abelian groups are one-dimensional, these are in fact copies and not higher-dimensional irreps. We will refer to the spaces $\mathbb{H}_q$ as *degeneracy spaces*. We can construct the unitary representations of U(1) appearing in Eq. (6.77) by using a local Hermitian operator $\hat{Q}$ whose expectation over all sites gives the total conserved charge as

$$\hat{U}_\phi = \exp\left(-i\hat{Q}\phi\right) . \tag{6.80}$$

The eigenstates of $\hat{Q}$ with eigenvalue q form a basis for $\mathbb{H}_q$. We will index the states in the degeneracy space $\mathbb{H}_q$ by the *degeneracy index* t_q which runs from $1, \ldots, d_q$ such that

$$\hat{Q}|qt_q\rangle = q|qt_q\rangle , \tag{6.81}$$

$$\hat{U}_\phi|qt_q\rangle = e^{-iq\phi}|qt_q\rangle , \tag{6.82}$$

$$\langle qt_q|q't'_q\rangle = \delta_{q,q'}\delta_{t_q t'_q} . \tag{6.83}$$

Hence, any state $|\psi\rangle$ which is an eigenstate of $\hat{Q}$ transforms symmetrically as

$$\hat{U}_\phi|\psi\rangle = \exp\left(-iq\phi\right)|\psi\rangle , \tag{6.84}$$

and so can be expanded in the basis states $|qt_q\rangle$ which span the degeneracy space $\mathbb{H}_q$:

$$|\psi\rangle = \sum_{t_q}\langle qt_q|\psi\rangle|qt_q\rangle . \tag{6.85}$$

Similarly, a linear operator $\hat{T}$ which is covariant in the sense that

$$\hat{U}_\phi\hat{T}\hat{U}^\dagger_\phi = e^{-i\Delta q\phi}\hat{T} \tag{6.86}$$

decomposes as

$$\hat{T} = \oplus_q \hat{T}_{q,q-\Delta q}\,, \tag{6.87}$$

where the $\hat{T}_{q,q-\Delta q}$ are $d_q \times d_{q-\Delta q}$ operators mapping $\mathbb{H}_{q-\Delta q} \to \mathbb{H}_q$. We will denote the matrix elements of $T_{q,q'}$ in the basis $|qt_q\rangle\langle q't'_q|$ as $[T_{qq'}]_{t_q t'_q}$. A linear operator which is covariant with $\Delta q = 0$ is said to be invariant, and commutes with the group operation. This is Schur's lemma [67]. Symmetric states and covariant operators are strongly constrained by the fact that they act only in the smaller subspaces $\mathbb{H}_q$ rather than the entire space $\mathbb{H}$.

To make these ideas more concrete, let us consider some explicit examples. First, let us consider a single lattice site which can accommodate up to 2 bosons. The charge in this case is the number of bosons, and so the operator $\hat{Q}$ is $\hat{n}$, the boson number operator. In the basis $\{|0\rangle, |1\rangle, |2\rangle\}$ where $|N\rangle$ is the Fock state with N bosons, this operator is

$$\hat{n} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{pmatrix}. \tag{6.88}$$

The Fock states $|0\rangle$, $|1\rangle$, and $|2\rangle$ span the spaces $\mathbb{H}_0$, $\mathbb{H}_1$, and $\mathbb{H}_2$ with charges 0, 1, and 2, respectively. Hence, all degeneracy spaces are one-dimensional, $d_q = 1\ \forall q$. The operator $\hat{n}$ is invariant, as it does not connect states with different charges. It may be written in the basis $|qt_q\rangle \in \{|01\rangle, |11\rangle, |21\rangle\}$ as

$$\hat{n} = \hat{n}_{00} \oplus \hat{n}_{11} \oplus \hat{n}_{22} = \begin{pmatrix} (0) & 0 & 0 \\ 0 & (1) & 0 \\ 0 & 0 & (2) \end{pmatrix}. \tag{6.89}$$

The fact that this operator is invariant has constrained that only the elements in the blocks denoted by parentheses may be nonzero. As the blocks $\hat{n}_{qq}$ are 1×1 matrices, the operator $\hat{n}$ is in fact specified by three numbers. An example of an operator which is covariant but not invariant is the bosonic destruction operator

$$\hat{b} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & \sqrt{2} \\ 0 & 0 & 0 \end{pmatrix} . \tag{6.90}$$

This operator reduces the particle number by 1, and so $\Delta q = -1$. In may be written in the basis $|qt_q\rangle \in \{|01\rangle, |11\rangle, |21\rangle\}$ as

$$\hat{b} = \hat{b}_{01} \oplus \hat{b}_{12} = \begin{pmatrix} 0 & (1) & 0 \\ 0 & 0 & \left(\sqrt{2}\right) \\ 0 & 0 & 0 \end{pmatrix} . \tag{6.91}$$

As a second example, consider a single lattice site in a system comprised of spin-1/2 fermions. The charge is now the total number of fermions. In the basis $\left\{|00\rangle, |1-\frac{1}{2}\rangle, |1\frac{1}{2}\rangle, |20\rangle\right\}$, where $|NS_z\rangle$ is the state with N particles and a total spin projection of S_z along the z direction, the total fermion number operator is

$$\hat{n} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix} . \tag{6.92}$$

Hence, the degeneracy space $\mathbb{H}_1$ spanned by $|1 \pm \frac{1}{2}\rangle$ is two-dimensional due to the spin degree of freedom. In the basis $|qt_q\rangle \in \{|01\rangle, |11\rangle, |12\rangle, |21\rangle\}$ we may write this invariant operator as

$$\hat{n} = \hat{n}_{00} \oplus \hat{n}_{11} \oplus \hat{n}_{22} = \begin{pmatrix} (0) & \begin{matrix} 0 & 0 \end{matrix} & 0 \\ \begin{matrix} 0 \\ 0 \end{matrix} & \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} & \begin{matrix} 0 \\ 0 \end{matrix} \\ 0 & \begin{matrix} 0 & 0 \end{matrix} & (2) \end{pmatrix} . \tag{6.93}$$

It is specified by the 1× 1 matrices $\hat{n}_{00}$ and $\hat{n}_{22}$ and the 2× 2 matrix $\hat{n}_{11}$. An example of an invariant operator with off-diagonal matrix elements in the degeneracy space is $\hat{S}^x = \hat{a}^\dagger_\uparrow \hat{a}_\downarrow + \hat{a}^\dagger_\downarrow \hat{a}^\dagger_\uparrow$, where $a_\uparrow$ destroys a fermion with $S_z = \frac{1}{2}$ and $a_\downarrow$ destroys a fermion with $S_z = -\frac{1}{2}$. This operator may be written in invariant form as

$$\hat{S}^x = \begin{pmatrix} (0) & \begin{matrix} 0 & 0 \end{matrix} & 0 \\ \begin{matrix} 0 \\ 0 \end{matrix} & \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} & \begin{matrix} 0 \\ 0 \end{matrix} \\ 0 & \begin{matrix} 0 & 0 \end{matrix} & (0) \end{pmatrix} . \tag{6.94}$$

These examples illustrate the general utility of symmetry conservation in tensor network algorithms. Conservation of a symmetry breaks a covariant operator into two pieces. The first piece is the structure of nonzero elements of the operator, specified by a direct sum of operators appropriately transforming between degeneracy spaces. This piece is completely determined by the symmetry. The remaining piece, which distinguishes a particular operator from all others which transform under the group operation in the same way, is the matrix elements of the operators acting on the degeneracy spaces.

Let us now consider the product of two spaces $\mathbb{H}^A$ and $\mathbb{H}^B$ which admit representations of U(1). The action of U(1) on the coupled system $\mathbb{H}^{AB} = \mathbb{H}^A \otimes \mathbb{H}^B$ is generated by the total charge operator $\hat{Q}^{AB} = \hat{Q}^A \otimes \hat{1}^B + \hat{1}^A \otimes \hat{Q}^B$, and so the space $\mathbb{H}^{AB}$ decomposes into irreps with total charge q_{AB} and degeneracy $d_{q_{AB}}$. This coupled basis carries the same information as the decoupled basis indexed by the charges of the subsystems q_A and q_B, and so a one-to-one correspondence must exist between the two. We define

$$C^{q_{AB}t_{AB}}_{q_A t_A; q_B t_B} \equiv \langle q_{AB} t_{AB} | q_A t_A q_B t_B \rangle \tag{6.95}$$

as the elements of the unitary matrix which encapsulate this one-to-one correspondence. For the Abelian case at hand, each element of this transformation is either zero or one, but for non-Abelian groups these elements represent the corresponding Clebsch-Gordan coefficients.[45]

---

[45] In the physics literature, the Clebsch-Gordan coefficients typically refer to this unitary transformation for the case of SU(2). Here we use it in the more general mathematical sense as the unitary matrix connecting the tensor product of the representation spaces of two irreps of a group to a direct sum of irreducible representation spaces [68].

As an example of the transformation to the coupled system, consider two lattice sites which can contain up to two bosons. Using Eq. (6.89), we can write the total charge operator in the uncoupled basis $|q_1 t_{q_1} q_2 t_{q_2}\rangle$=$\{|0101\rangle, |0111\rangle, |0121\rangle, |1101\rangle, |1111\rangle, |1121\rangle, |2101\rangle, |2111\rangle, |2121\rangle\}$ as

$$\hat{n} = \hat{n}_1 \otimes \hat{I} + \hat{I} \otimes \hat{n}_2 = \begin{pmatrix} 0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0 \\ 0&0&2&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&2&0&0&0&0 \\ 0&0&0&0&0&3&0&0&0 \\ 0&0&0&0&0&0&2&0&0 \\ 0&0&0&0&0&0&0&3&0 \\ 0&0&0&0&0&0&0&0&4 \end{pmatrix}. \tag{6.96}$$

Ordering the vectors in terms of their total charges, we can identify the invariant form of the total charge operator

$$\hat{n} = \hat{n}_{00} \oplus \hat{n}_{11} \oplus \hat{n}_{22} \oplus \hat{n}_{33} \oplus \hat{n}_{44} \tag{6.97}$$

$$= \begin{pmatrix} (0) & \begin{matrix}0&0\end{matrix} & \begin{matrix}0&0&0\end{matrix} & \begin{matrix}0&0\end{matrix} & 0 \\ \begin{matrix}0\\0\end{matrix} & \begin{pmatrix}1&0\\0&1\end{pmatrix} & \begin{matrix}0&0&0\\0&0&0\end{matrix} & \begin{matrix}0&0\\0&0\end{matrix} & \begin{matrix}0\\0\end{matrix} \\ \begin{matrix}0\\0\\0\end{matrix} & \begin{matrix}0&0\\0&0\\0&0\end{matrix} & \begin{pmatrix}2&0&0\\0&2&0\\0&0&2\end{pmatrix} & \begin{matrix}0&0\\0&0\\0&0\end{matrix} & \begin{matrix}0\\0\\0\end{matrix} \\ \begin{matrix}0\\0\end{matrix} & \begin{matrix}0&0\\0&0\end{matrix} & \begin{matrix}0&0&0\\0&0&0\end{matrix} & \begin{pmatrix}3&0\\0&3\end{pmatrix} & \begin{matrix}0\\0\end{matrix} \\ 0 & \begin{matrix}0&0\end{matrix} & \begin{matrix}0&0&0\end{matrix} & \begin{matrix}0&0\end{matrix} & (4) \end{pmatrix}, \tag{6.98}$$

and the corresponding nonzero Clebsch-Gordan coefficients $C^{01}_{01;01} = C^{11}_{01;11} = C^{12}_{11;01} = C^{21}_{01;21} = C^{22}_{11;11} = C^{23}_{21;01} = C^{31}_{11;21} = C^{32}_{21;11} = C^{41}_{21;21} = 1$. This construction also demonstrates that even if the irreps of a single lattice site are non-degenerate, a *combinatoric degeneracy* in the space of fixed total charge arises when multiple lattice sites are considered.

If we now consider the general case of an L-fold tensor product of a space admitting a representation of U(1) then the total charge operator is

$$\hat{Q} = \sum_{i=1}^{L} \hat{Q}_i \,, \tag{6.99}$$

where $\hat{Q}_i$ is the local charge operator at site i. The total charge operator generates the unitary transformations

$$\hat{U}_{\phi;L} = \exp\left(-i\hat{Q}\phi\right) = \left(\hat{U}_\phi\right)^{\otimes L} \,. \tag{6.100}$$

The total tensor product space thus decomposes into spaces of fixed total charge. From Schur's lemma [67], an invariant Hamiltonian has no matrix elements between states that have differing total charge, and so we can diagonalize the Hamiltonian in subspaces of fixed total charge. Thus, all eigenstates of an invariant Hamiltonian can be chosen to be symmetric in the sense of Eq. (6.84). This makes rigorous the intuitive notion that the eigenvectors of a Hamiltonian which conserves the number of particles can be chosen to have a definite number of particles.[46]

We can generalize the commutation relation Eq. (6.86) for linear operators to tensors by introducing the notion of *incoming* and *outgoing* indices which encapsulate the overall charge flow described by the tensor. The distinction is that incoming indices transform as $\hat{U}_\phi$ under group action while outgoing indices transform as $\hat{U}_\phi^\dagger$. Intuitively, incoming indices denote charge flowing into the tensor and outgoing indices describe charge flow out from the tensor. With this definition, we have that a tensor T is invariant under the group action if

$$\sum_{\mathbf{i}'\in I}\left[\hat{U}_\phi \otimes \cdots \otimes \hat{U}_\phi\right]_{\mathbf{i}\mathbf{i}'} \sum_{\mathbf{j}'\in O}\left[\hat{U}_\phi^\dagger \otimes \cdots \otimes \hat{U}_\phi^\dagger\right]_{\mathbf{j}\mathbf{j}'} T_{\mathbf{i}'\mathbf{j}'} = T_{\mathbf{i}\mathbf{j}} \,. \tag{6.101}$$

Here I denotes all incoming indices and O all outgoing indices, we have permuted the indices of T such that all incoming indices lie to the left and all outgoing indices to the

---

[46] Here we say only that the eigenvectors can be chosen in this way to account for possible energetic degeneracies of two states with differing total particle number. In the presence of such a degeneracy, any linear combination of the degenerate states is also an eigenstate.

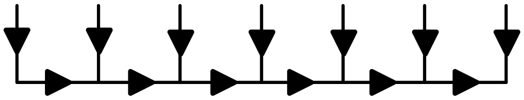


Figure 6.6: Structure of charge flow through an MPS in diagrammatic notation.

right, and the notation $[A\otimes B]_{ij}$ denotes the ij^{th} component of the tensor product. Now, we consider decomposing each on-site state $|i_j\rangle$ into its local degeneracy spaces spanned by $|q_j t_{q_j}\rangle$ using Eq. (6.84). This implies that the tensor decomposes as

$$T_{i_1\dots i_L}|i_1\dots i_L\rangle = \left[T_{q_1\dots q_L}\right]_{t_{q_1}\dots t_{q_L}}|q_1t_{q_1}\dots q_Lt_{q_L}\rangle\,, \tag{6.102}$$

and the condition that the tensor be invariant becomes that the total incoming and outgoing charge be the same, where the incoming and outgoing charges are defined as $Q_{\mathrm{incoming}}=\sum_{i\in I}q_i$ and $Q_{\mathrm{outgoing}}=\sum_{i\in O}q_i$, respectively. This implies that any invariant tensor takes the form

$$T_{i_1\dots i_L} = \left[T_{q_1\dots q_L}\right]_{t_{q_1}\dots t_{q_L}}\delta_{Q_{\mathrm{incoming}}Q_{\mathrm{outgoing}}}\,. \tag{6.103}$$

This canonical form is the key result of our analysis. Using the symmetry, we have broken each invariant tensor into a part determined by the symmetry (the delta function) and a part which is not (indexed by the t_j). This canonical form may be viewed as a conservation of charge by invariant tensors, as the amount of charge flowing in is equal to the amount of charge flowing out.

For MPS tensors $A^i_{\alpha\beta}$, we define α and i to be incoming and β to be outgoing such that q_γ with a Greek index denotes the total charge to the left of a given bond and q_i with a Roman index denotes the charge of the particular on-site irrep. The tensors A then become arrays of tensors $\left[A^{q_i}_{q_\alpha q_\beta}\right]^{t_{q_i}}_{t_{q_\alpha}t_{q_\beta}}$. This convention of incoming and outgoing indices can be displayed in a modified tensor network diagram as in

Figure 6.6, compare Figure 6.3(c).[47] The arrows indicate the flow of charge through an MPS tensor, with the local states $|q_j t_{q_j}\rangle$ acting as sources of charge. We shall refer to the canonical form of a tensor which is covariant under the action of U(1) as a *q-tensor*. All of the results discussed above extend immediately to products of Abelian groups by considering the charge q to be a vector $\mathbf{q}$ whose entries are the charges of the individual Abelian symmetries under consideration. Multiple symmetries are relevant in models whose constituents are particles with spin, as the total number and the total magnetization are often independently conserved, for example.

To perform index fusion on a q-tensor we use Eq. (6.95) in the form

$$\left[T'_{q_{(\alpha i)}q_\beta}\right]_{t_{q_{(\alpha i)}}t_{q_\beta}} = \sum_{q_\alpha q_i t_{q_\alpha} t_{q_i}} C^{q_{(\alpha i)}t_{q_{(\alpha i)}}}_{q_\alpha t_{q_\alpha};q_i t_{q_i}} \left[T^{q_i}_{q_\alpha q_\beta}\right]^{t_{q_i}}_{t_{q_\alpha}t_{q_\beta}}, \tag{6.104}$$

where $q_{(\alpha i)}$ is the charge of $|\alpha\rangle|i\rangle$. The transformation tensor C does not change the total charge, and so it is an invariant tensor. Taking α and i to be incoming indices and (αi) to be an outgoing index hence requires that $q_{\alpha i} = q_\alpha * q_i$, where $*$ is the group operation and we can characterize the degeneracy index of the composite system as the Kronecker product of the degeneracy indices from the constituent systems, $t_{q_{(\alpha i)}} = (t_{q_\alpha} t_{q_i})$. That is, the symmetry determines which irreps can combine and then the degeneracy indices are combined under ordinary index fusion.[48] Writing this out explicitly, we have

$$\left[T'_{q_\alpha * q_i, q_\beta}\right]_{\left(t_{q_\alpha}t_{q_i}\right)t_{q_\beta}} = \left[T^{q_i}_{q_\alpha q_\beta}\right]^{t_{q_i}}_{t_{q_\alpha}t_{q_\beta}}. \tag{6.105}$$

Splitting is performed by reading this expression in reverse. Using these fusion and splitting rules we can also apply the matrix decompositions discussed earlier to de-

---

[47] This construction can also be compared with circuit diagrams demonstrating conservation of charge as specified by Kirchoff's laws.

[48] This is the most significant difference between the Abelian and non-Abelian cases. In the latter, the allowed irreps are enumerated by the Clebsch-Gordan series, and the elements of the unitary matrix relating $t_{q_{(\alpha i)}}$ to t_{q_α} and t_{q_i} are nontrivial.

velop canonical forms for q-tensors.

It is remarkable that the only place where the specific form of the Abelian group arises is in the charge fusion rule $q_{\alpha i} = q_\alpha * q_i$. For U(1) $*$ is ordinary addition while for $\mathbb{Z}_p$ $*$ is addition mod p. In general, we can use the Cayley table [69] of the group under consideration to construct the fusion rule.

The contraction of two q-tensors is restricted by the fact that incoming indices in one covariant tensor can only be contracted with outgoing indices of another covariant tensor in order for the result to also be a covariant tensor. Furthermore, as can be gleaned from the fusion rule above, the charges of each index which are contracted have to agree between the two tensors in order to yield a nonzero result. Hence, the procedure to contract the sets of indices A and B of two q-tensors U and V together is to find all matching charges within the two sets q_A and q_B, and then contract the degeneracy tensors using ordinary tensor contraction, Eq. (6.17), resulting in

$$\left[T_{q_{\bar{A}}q_{\bar{B}}}\right]_{t_{\bar{A}}t_{\bar{B}}} = \sum_q \sum_{t_q} \left[U_{q_{\bar{A}}q}\right]_{t_{\bar{A}}t_q} \left[V_{qq_{\bar{B}}}\right]_{t_q t_{\bar{B}}} . \tag{6.106}$$

Hence, essentially all expressions using MPSs can be translated directly for their (Abelian) symmetry-adapted counterparts by replacing summations over indices with simultaneous summations over the charges and degeneracy dimensions.

When written naively, Eq. (6.106) implies an algorithm which scales as $\mathcal{O}\left(N_U N_V\right)$, where N_T is the number of irreps in T. This scaling can be improved by sorting the quantum numbers and using a binary search to find matches. We note that, because the charges can be put into one-to-one correspondence with some subset of the integers, we can always define the vacuum charge to be zero for our finite Hilbert spaces. Rather than working directly with the charges themselves, it is useful to define a *hash function* [70] which takes unique arrays of nonnegative integers to unique values.[49] That is, the function is *injective*. A hash function satisfying this criterion, which has a number of other useful properties for our purposes, is the square-root of primes hash

$$H(\mathbf{q}) = \sum_i q_i \sqrt{p_i}\,. \tag{6.107}$$

Here p_i is the i^{th} prime, and the injectivity of H follows from the fact that the numbers $\sqrt{p_i}$ are incommensurable. A nice feature of $H(\mathbf{q})$ is that it is linear in the elements of $\mathbf{q}$, and so given the values of $\sqrt{p_i}$ the hash function can be obtained for arbitrary combinations of charges very quickly. In practice one never needs more than a few primes, and these may be generated once using a sieve and stored for later use. With hashing, the procedure for contracting two q-tensors becomes

1. Hash U and V according to A and B in $\mathcal{O}(N_U + N_V)$ time.
2. Sort the smaller of the two sets A and B according to its hashes in $\mathcal{O}(N_M \log N_M)$ time, where $N_M = \min(N_U, N_V)$.
3. Loop through the elements of the larger list and perform a binary search to find matching hashes in the smaller list in $\mathcal{O}(N_X \log N_M)$ time, where $N_X = \max(N_U, N_V)$.
4. Contract all matching hashes using the contraction algorithm for ordinary tensors on the degeneracy spaces.

Using the square-root of primes hash makes the process of contracting tensors efficient even when permutations, large numbers of indices, or multiple Abelian symmetries are involved.

As an example of a symmetric MPS, let us again consider the W state from Sec. 6.4.3

---

[49] This is in fact the definition of a perfect hash function, which is harder to find than a hash function in general but can be explicitly found for our purposes.

$$|W\rangle = \frac{1}{\sqrt{L}}\sum_{i=1}^{L}|0\rangle^{\otimes i-1}|1\rangle|0\rangle^{\otimes L-i}\,. \tag{6.108}$$

If we interpret $|0\rangle$ as a state with no particles and $|1\rangle$ as a state with 1 particle, then the W state is an eigenstate of the total particle number operator with eigenvalue 1. We will use charge interchangeably with the number of particles in what follows. We can write the W state as

$$|W\rangle = \frac{1}{\sqrt{L}}\left[|0\rangle^{\otimes j-1}|1\rangle|0\rangle^{\otimes L-j} + |W;j-1\rangle|0\rangle|0\rangle^{\otimes L-j} + |0\rangle^{\otimes j-1}|0\rangle|W;L-j\rangle\right]\,, \tag{6.109}$$

where

$$|W;\ell\rangle = \sum_{i=1}^{\ell}|0\rangle^{\otimes i-1}|1\rangle|0\rangle^{\otimes \ell-i} \tag{6.110}$$

is an unnormalized W state on ℓ sites. Hence, if site j is in state $|1\rangle$, then it follows that the state to the left of site j is $|0\rangle^{\otimes j-1}$ and the state to the right of site j is $|0\rangle^{\otimes L-j}$. The state $|1\rangle$ is the unique on-site state with charge $q=1$, and so $|1\rangle = |11\rangle$ in the $|q_j t_{q_j}\rangle$ basis. Furthermore, $|00\dots0\rangle$ is the unique state with charge 0, and so $|00\dots0\rangle = |01\rangle$ in the $|q_\alpha t_{q_\alpha}\rangle$ basis. This implies that the contribution to the MPS at site j with quantum numbers $(q_\alpha q_i q_\beta) = (010)$ is

$$\left[A_{00}^{[j]1}\right]_{11}^{1} = 1\,. \tag{6.111}$$

If site j is in state $|0\rangle$, then either the state to the left is $|W;j-1\rangle$ and the state to the right is $|00\dots0\rangle$ or the state to the left is $|00\dots0\rangle$ and the state to the right is $|W;\ell-j\rangle$. Because the W state on any number of sites has charge 1 and the W state is the only state with charge 1 relevant to any subsystem, we have the elements

$$\left[A_{01}^{[j]0}\right]_{11}^{1} = 1\,, \quad \left[A_{10}^{[j]0}\right]_{11}^{1} = 1\,. \tag{6.112}$$

Hence, the complete MPS q-tensor at site j is the direct sum of the elements

$$\left[A_{01}^{[j]0}\right]_{11}^{1} = 1\,, \quad \left[A_{10}^{[j]0}\right]_{11}^{1} = 1\,, \quad \left[A_{00}^{[j]1}\right]_{11}^{1} = 1\,, \tag{6.113}$$

compare Eq. (6.53). Both the MPS tensor Eq. (6.53) and the MPS q-tensor Eq. (6.113) have three independent elements. The difference is that Eq. (6.113) writes these three elements as a direct sum with the topology of index contraction determined by the charges of the subsystems. The utility of the q-tensor decomposition is that contractions over tensors become direct sums of contractions over tensors with smaller linear dimensions, see Eq. (6.106). Because of the polynomial scaling of contractions with the bond dimension, this leads to a significant speedup when symmetries are explicitly utilized. In the present case the reduction of the bond dimension is complete, as the tensors in Eq. (6.113) are all 1×1×1 dimensional.

The present chapter gives an intuition for why MPSs are useful as variational ansätze for strongly-correlated 1D systems, demonstrates how to extract physical information from MPSs, and explains how to express MPSs in terms of irreducible tensors such that Abelian symmetries are explicitly conserved. In Chapter 7, we discuss how to generate MPS representations of eigenstates of 1D systems through variational means. Additionally, it is shown how to time-evolve the MPS form of a wavefunction in a generic way. These algorithms all apply to lattice systems of finite extent. In Chapter 8 we present an algorithm for variationally finding the ground state of a homogenous 1D system which has infinite extent. This is done by introducing an MPS decomposition of a unit cell which, when infinitely repeated, generates the full state. This unit cell is then optimized using variations of the finite-size algorithms. Finally, in Chapter 9, we discuss how algorithms for time evolution of MPSs may be adapted to extract finite-temperature properties of 1D systems. This chapter also discusses the form of MPSs and the structure of their algorithms for systems with periodic boundary conditions.

6.7 References Cited

[1] Ulrich Schollwöck. The density-matrix renormalization group in the age of matrix product states. *Annals of Physics*, 326(1):96 – 192, 2011. ISSN 0003-4916. doi: 10.1016/j.aop.2010.09.012. URL `http://www.sciencedirect.com/science/article/pii/S0003491610001752`. <ce:title>January 2011 Special Issue</ce:title>.

[2] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins Studies in Mathematical Sciences. The Johns Hopkins University Press, Baltimore, 3 edition, 1996.

[3] Ernest R. Davidson. The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices. *Journal of Computational Physics*, 17(1):87 – 94, 1975. ISSN 0021-9991. doi: 10.1016/0021-9991(75)90065-0. URL `http://www.sciencedirect.com/science/article/pii/0021999175900650`.

[4] David Poulin, Angie Qarry, Rolando Somma, and Frank Verstraete. Quantum Simulation of Time-Dependent Hamiltonians and the Convenient Illusion of Hilbert Space. *Phys. Rev. Lett.*, 106:170501, Apr 2011. doi: 10.1103/PhysRevLett.106.170501. URL `http://link.aps.org/doi/10.1103/PhysRevLett.106.170501`.

[5] J Huyghebaert and H De Raedt. Product formula methods for time-dependent Schrodinger problems. *Journal of Physics A: Mathematical and General*, 23(24):5777, 1990. URL `http://stacks.iop.org/0305-4470/23/i=24/a=019`.

[6] A. L. Fetter and J. D. Walecka. *Quantum Theory of Many-Particle Systems*. McGraw-Hill, Inc., New York, 1971.

[7] A. Kitaev, A. H. Shen, and M. N. Vyalyi. *Classical and Quantum Information*. AMS, Providence, 2002.

[8] M. Nielsen and I. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, 2000.

[9] Christopher M. Dawson and Michael A. Nielsen. The Solovay-Kitaev algorithm. *Quant. Inf. Comp.*, 6(1):81–95, 2006.

[10] M. Kliesch, T. Barthel, C. Gogolin, M. Kastoryano, and J. Eisert. Dissipative Quantum Church-Turing Theorem. *Phys. Rev. Lett.*, 107:120501, Sep 2011. doi: 10.1103/PhysRevLett.107.120501. URL `http://link.aps.org/doi/10.1103/PhysRevLett.107.120501`.

[11] Mark Srednicki. Entropy and area. *Phys. Rev. Lett.*, 71:666–669, Aug 1993. doi: 10.1103/PhysRevLett.71.666. URL `http://link.aps.org/doi/10.1103/PhysRevLett.71.666`.

[12] Christoph Holzhey, Finn Larsen, and Frank Wilczek. Geometric and renormalized entropy in conformal field theory. *Nuclear Physics B*, 424(3):443 – 467, 1994. ISSN 0550-3213. doi: 10.1016/0550-3213(94)90402-2. URL `http://www.sciencedirect.com/science/article/pii/0550321394904022`.

[13] G. Vidal, J. I. Latorre, E. Rico, and A. Kitaev. Entanglement in Quantum Critical Phenomena. *Phys. Rev. Lett.*, 90:227902, Jun 2003. doi: 10.1103/PhysRevLett.90.227902. URL `http://link.aps.org/doi/10.1103/PhysRevLett.90.227902`.

[14] Pasquale Calabrese and John Cardy. Entanglement entropy and quantum field theory. *Journal of Statistical Mechanics: Theory and Experiment*, 2004 (06):P06002, 2004. URL `http://stacks.iop.org/1742-5468/2004/i=06/a=P06002`.

[15] J. Eisert, M. Cramer, and M. B. Plenio. *Colloquium* : Area laws for the entanglement entropy. *Rev. Mod. Phys.*, 82:277–306, Feb 2010. doi: 10.1103/RevModPhys.82.277. URL `http://link.aps.org/doi/10.1103/RevModPhys.82.277`.

[16] M B Hastings. An area law for one-dimensional quantum systems. *Journal of Statistical Mechanics: Theory and Experiment*, 2007(10):P08024, 2007.

[17] G Vitagliano, A Riera, and J I Latorre. Volume-law scaling for the entanglement entropy in spin-1/2 chains. *New Journal of Physics*, 12(11):113049, 2010. URL `http://stacks.iop.org/1367-2630/12/i=11/a=113049`.

[18] Michael M. Wolf, Frank Verstraete, Matthew B. Hastings, and J. Ignacio Cirac. Area Laws in Quantum Systems: Mutual Information and Correlations. *Phys. Rev. Lett.*, 100:070502, Feb 2008. doi: 10.1103/PhysRevLett.100.070502. URL `http://link.aps.org/doi/10.1103/PhysRevLett.100.070502`.

[19] Allan Adams, Lincoln D. Carr, Thomas Schaefer, Peter Steinberg, and John E. Thomas. Strongly Correlated Quantum Fluids: Ultracold Quantum Gases, Quantum Chromodynamic Plasmas, and Holographic Duality. *New. J. Phys. under review, arXiv:1205.5180v1*, 2012.

[20] Jacob D. Bekenstein. Black Holes and Entropy. *Phys. Rev. D*, 7:2333–2346, Apr 1973. doi: 10.1103/PhysRevD.7.2333. URL `http://link.aps.org/doi/10.1103/PhysRevD.7.2333`.

[21] Raphael Bousso. The holographic principle. *Rev. Mod. Phys.*, 74:825–874, Aug 2002. doi: 10.1103/RevModPhys.74.825. URL `http://link.aps.org/doi/10.1103/RevModPhys.74.825`.

[22] Tobias J. Osborne, Jens Eisert, and Frank Verstraete. Holographic Quantum States. *Phys. Rev. Lett.*, 105:260401, Dec 2010. doi: 10.1103/PhysRevLett.105.260401. URL `http://link.aps.org/doi/10.1103/PhysRevLett.105.260401`.

[23] J. Ignacio Cirac, Didier Poilblanc, Norbert Schuch, and Frank Verstraete. Entanglement spectrum and boundary theories with projected entangled-pair states. *Phys. Rev. B*, 83:245134, Jun 2011. doi: 10.1103/PhysRevB.83.245134. URL `http://link.aps.org/doi/10.1103/PhysRevB.83.245134`.

[24] Michael M. Wolf. Violation of the Entropic Area Law for Fermions. *Phys. Rev. Lett.*, 96:010404, Jan 2006. doi: 10.1103/PhysRevLett.96.010404. URL `http://link.aps.org/doi/10.1103/PhysRevLett.96.010404`.

[25] Dimitri Gioev and Israel Klich. Entanglement Entropy of Fermions in Any Dimension and the Widom Conjecture. *Phys. Rev. Lett.*, 96:100503, Mar 2006. doi: 10.1103/PhysRevLett.96.100503. URL `http://link.aps.org/doi/10.1103/PhysRevLett.96.100503`.

[26] M. B. Plenio, J. Eisert, J. Dreißig, and M. Cramer. Entropy, Entanglement, and Area: Analytical Results for Harmonic Lattice Systems. *Phys. Rev. Lett.*, 94:060503, Feb 2005. doi: 10.1103/PhysRevLett.94.060503. URL `http://link.aps.org/doi/10.1103/PhysRevLett.94.060503`.

[27] M. Cramer, J. Eisert, M. B. Plenio, and J. Dreißig. Entanglement-area law for general bosonic harmonic lattice systems. *Phys. Rev. A*, 73:012309, Jan 2006. doi: 10.1103/PhysRevA.73.012309. URL `http://link.aps.org/doi/10.1103/PhysRevA.73.012309`.

[28] M Cramer and J Eisert. Correlations, spectral gap and entanglement in harmonic quantum systems on generic lattices. *New Journal of Physics*, 8(5):71, 2006. URL `http://stacks.iop.org/1367-2630/8/i=5/a=071`.

[29] John Cardy. *Scaling and Renormalization in Statistical Physics*. Cambridge University Press, Cambridge, 1996.

[30] Pasquale Calabrese and John Cardy. Entanglement entropy and conformal field theory. *Journal of Physics A: Mathematical and Theoretical*, 42(50):504005, 2009. URL `http://stacks.iop.org/1751-8121/42/i=50/a=504005`.

[31] Philippe Di Francesco, Pierre Mathieu, and David Sénéchal. *Conformal field theory*. Springer, New York, 1997.

[32] F. Verstraete, D. Porras, and J. I. Cirac. Density Matrix Renormalization Group and Periodic Boundary Conditions: A Quantum Information Perspective. *Phys. Rev. Lett.*, 93(22):227205, Nov 2004. doi: 10.1103/PhysRevLett.93.227205.

[33] F. Verstraete and J. I. Cirac. Matrix product states represent ground states faithfully. *Phys. Rev. B*, 73:094423, 2006. URL `http://link.aps.org/abstract/PRB/v73/e094423`.

[34] Norbert Schuch, Ignacio Cirac, and Frank Verstraete. Computational Difficulty of Finding Matrix Product Ground States. *Phys. Rev. Lett.*, 100:250501, Jun 2008. doi: 10.1103/PhysRevLett.100.250501. URL `http://link.aps.org/doi/10.1103/PhysRevLett.100.250501`.

[35] Kenneth G. Wilson. The renormalization group: Critical phenomena and the Kondo problem. *Rev. Mod. Phys.*, 47:773–840, 1975.

[36] Steven R. White. Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.*, 69(19):2863–2866, Nov 1992. doi: 10.1103/PhysRevLett.69.2863.

[37] U. Schollwock. The density-matrix renormalization group. *Rev. Mod. Phys.*, 77:259, 2005.

[38] Erhard Schmidt. Zur Theorie der linearen und nichtlinearen Integralgleichungen. *Mathematische Annalen*, 63:433–476, 1907. ISSN 0025-5831. URL `http://dx.doi.org/10.1007/BF01449770`. 10.1007/BF01449770.

[39] Stellan Östlund and Stefan Rommer. Thermodynamic Limit of Density Matrix Renormalization. *Phys. Rev. Lett.*, 75(19):3537–3540, Nov 1995. doi: 10.1103/PhysRevLett.75.3537.

[40] Stefan Rommer and Stellan Östlund. Class of ansatz wave functions for one-dimensional spin systems and their relation to the density matrix renormalization group. *Phys. Rev. B*, 55(4):2164–2181, Jan 1997. doi: 10.1103/PhysRevB.55.2164.

[41] Basic Linear Algebra Subprograms: http://www.netlib.org/blas.

[42] Y.-Y. Shi, L.-M. Duan, and G. Vidal. Classical simulation of quantum many-body systems with a tree tensor network. *Physical Review A (Atomic, Molecular, and Optical Physics)*, 74(2):022320, 2006. doi: 10.1103/PhysRevA.74.022320. URL http://link.aps.org/abstract/PRA/v74/e022320.

[43] F. Verstraete and J. I. Cirac. Renormalization algorithms for Quantum Many-Body systems in two and higher dimensions. *arXiv:cond-mat/0407066v1*, 2004.

[44] F. Verstraete, V. Murg, and J.I. Cirac. Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems. *Advances in Physics*, 57(2):143–224, 2008. doi: 10.1080/14789940801912366. URL http://www.tandfonline.com/doi/abs/10.1080/14789940801912366.

[45] G. Vidal. Entanglement Renormalization. *Phys. Rev. Lett.*, 99:220405, 2007.

[46] G. Evenbly and G. Vidal. Entanglement Renormalization in Two Spatial Dimensions. *Phys. Rev. Lett.*, 102(18):180406, May 2009. doi: 10.1103/PhysRevLett.102.180406.

[47] Peter Pippan, Steven R. White, and Hans Gerd Evertz. Efficient matrix-product state method for periodic boundary conditions. *Phys. Rev. B*, 81(8):081103, Feb 2010. doi: 10.1103/PhysRevB.81.081103.

[48] B. Pirvu, F. Verstraete, and G. Vidal. Exploiting translational invariance in matrix product state simulations of spin chains with periodic boundary conditions. *Phys. Rev. B*, 83:125104, Mar 2011. doi: 10.1103/PhysRevB.83.125104. URL http://link.aps.org/doi/10.1103/PhysRevB.83.125104.

[49] G. Vidal. Efficient Classical Simulation of Slightly Entangled Quantum Computations. *Phys. Rev. Lett.*, 91:147902–1–4, 2003.

[50] G. Vidal. Efficient Simulation of One-Dimensional Quantum Many-Body Systems. *Phys. Rev. Lett.*, 93:040502–1–4, 2004.

[51] J.A. Wheeler and W.H. Zurek, editors. *Quantum Theory and Measurement*. Princeton university Press, Princeton, NJ, 1983.

[52] Dik Bouwmeester, Jian-Wei Pan, Matthew Daniell, Harald Weinfurter, and Anton Zeilinger. Observation of Three-Photon Greenberger-Horne-Zeilinger Entanglement. *Phys. Rev. Lett.*, 82:1345–1349, Feb 1999. doi: 10.1103/PhysRevLett.82.1345. URL http://link.aps.org/doi/10.1103/PhysRevLett.82.1345.

[53] Ian Affleck, Tom Kennedy, Elliott H. Lieb, and Hal Tasaki. Rigorous results on valence-bond ground states in antiferromagnets. *Phys. Rev. Lett.*, 59:799–802, Aug 1987. doi: 10.1103/PhysRevLett.59.799. URL `http://link.aps.org/doi/10.1103/PhysRevLett.59.799`.

[54] M. B. Hastings. Solving gapped Hamiltonians locally. *Phys. Rev. B*, 73:085115, Feb 2006. doi: 10.1103/PhysRevB.73.085115. URL `http://link.aps.org/doi/10.1103/PhysRevB.73.085115`.

[55] D. Perez-Garcia, F. Verstraete, M. M. Wolf, and J. I. Cirac. Matrix Product State Representations. *Quantum Inf. Comput*, 7(401), 2007.

[56] Frank Pollmann, Subroto Mukerjee, Ari M. Turner, and Joel E. Moore. Theory of Finite-Entanglement Scaling at One-Dimensional Quantum Critical Points. *Phys. Rev. Lett.*, 102:255701, Jun 2009. doi: 10.1103/PhysRevLett.102.255701. URL `http://link.aps.org/doi/10.1103/PhysRevLett.102.255701`.

[57] M. N. Barber. *Phase Transitions and Critical Phenomena, edited by C. Domb and J. L. Lebowitz Vol. 8.* Academic, New York, NY, 1983.

[58] B. Pirvu, G. Vidal, F. Verstraete, and L. Tagliacozzo. Matrix product states for critical spin chains: finite size scaling versus finite entanglement scaling, 2012. URL `http://arxiv.org/abs/1204.3934`.

[59] Andrew John Daley. *Manipulation and Simulation of Cold Atoms in Optical Lattices.* PhD thesis, Leopold-Franzens-Universität Innsbruck, 2005.

[60] I. P. McCulloch and M. Gulácsi. The non-Abelian density matrix renormalization group algorithm. *EPL (Europhysics Letters)*, 57(6):852, 2002. URL `http://stacks.iop.org/0295-5075/57/i=6/a=852`.

[61] Ian P. McCulloch. From density-matrix renormalization group to matrix product states. *Journal of Statistical Mechanics: Theory and Experiment*, 2007 (10):P10014, 2007. URL `http://stacks.iop.org/1742-5468/2007/i=10/a=P10014`.

[62] Sukhwinder Singh, Huan-Qiang Zhou, and Guifre Vidal. Simulation of one-dimensional quantum systems with a global SU(2) symmetry. *New Journal of Physics*, 12(3):033029, 2010. URL `http://stacks.iop.org/1367-2630/12/i=3/a=033029`.

[63] Sukhwinder Singh, Robert N. C. Pfeifer, and Guifre Vidal. Tensor network states and algorithms in the presence of a global U(1) symmetry. *Phys. Rev. B*, 83: 115125, Mar 2011. doi: 10.1103/PhysRevB.83.115125. URL http://link.aps.org/doi/10.1103/PhysRevB.83.115125.

[64] Sukhwinder Singh, Robert N. C. Pfeifer, and Guifré Vidal. Tensor network decompositions in the presence of a global symmetry. *Phys. Rev. A*, 82:050301, Nov 2010. doi: 10.1103/PhysRevA.82.050301. URL http://link.aps.org/doi/10.1103/PhysRevA.82.050301.

[65] B. Bauer, P. Corboz, R. Orús, and M. Troyer. Implementing global Abelian symmetries in projected entangled-pair state algorithms. *Phys. Rev. B*, 83:125106, Mar 2011. doi: 10.1103/PhysRevB.83.125106. URL http://link.aps.org/doi/10.1103/PhysRevB.83.125106.

[66] Eugene P. Wigner. *Group theory and its application to the quantum mechanics of atomic spectra*. Academic press, New York, 1959.

[67] Morton Hamermesh. *Group theory and its application to physical problems*. Dover, 1989.

[68] Arne Alex, Matthias Kalus, Alan Huckleberry, and Jan von Delft. A numerical algorithm for the explicit calculation of SU(N) and SL(N,$\mathbb{C}$) ClebschGordan coefficients. *J. Math. Phys.*, 52:023507, 2011.

[69] Robert McArdle Keown. *An introduction to group representation theory*. Academic Press, New York, 1975.

[70] Donald Ervin Knuth. *The art of computer programming*. Addison-Wesley, Reading, Mass, 1973.

CHAPTER 7

OUT OF EQUILIBRIUM DYNAMICS WITH MATRIX PRODUCT STATES

Abstract:[50] Theoretical understanding of strongly correlated systems in one spatial dimension (1D) has been greatly advanced by the density-matrix renormalization group (DMRG) algorithm, which is a variational approach using a class of entanglement-restricted states called Matrix Product States (MPSs). However, DRMG suffers from inherent accuracy restrictions when multiple states are involved due to multi-state targeting and also the approximate representation of the Hamiltonian and other operators. By formulating the variational approach of DMRG explicitly for MPSs one can avoid errors inherent in the multi-state targeting approach. Furthermore, by using the Matrix Product Operator (MPO) formalism, one can exactly represent the Hamiltonian and other operators relevant for the calculation. The MPO approach allows 1D Hamiltonians to be templated using a small set of finite state automaton rules without reference to the particular microscopic degrees of freedom. We present two algorithms which take advantage of these properties: eMPS to find excited states of 1D Hamiltonians and tMPS for the time evolution of a generic time-dependent 1D Hamiltonian. We properly account for time-ordering of the propagator such that the error does not depend on the rate of change of the Hamiltonian. Our algorithms use only the MPO form of the Hamiltonian, and so are applicable to microscopic degrees of freedom of any variety, and do not require Hamiltonian-specialized implementation. We benchmark our algorithms with a case study of the Ising model, where the critical point is located using entanglement measures. We then study the dynamics of this model under a time-dependent quench of the transverse field through the critical point. Finally, we present studies of a dipolar, or long-range Ising model,

---

[50] *Out of equilibrium dynamics with Matrix Product States*, M. L. Wall and L. D. Carr, New J. Physics, under review.

again using entanglement measures to find the critical point and study the dynamics of a time-dependent quench through the critical point.

7.1 Introduction

The great success of experimental ultracold atomic physics has made the study of strongly correlated one-dimensional (1D) quantum systems a major avenue of current physics research. Examples of novel condensed matter physics realized with 1D atomic systems include the role of integrability in thermalization [1] and static [2] and dynamic [3] quantum simulators of Hubbard models. Furthermore, as ultracold molecules approach quantum degeneracy [4–6], lattice models with complex internal degrees of freedom and long-range interactions become relevant [7–9]. As more and more complex models become amenable to study, the need for numerical methods which can adapt to different degrees of freedom, different Hamiltonians, and different dynamical processes thus becomes essential.

In addition to practical interest in understanding and benchmarking atomic and molecular quantum simulators, the ability to simulate the dynamics of 1D systems also provides insight into fundamental questions such as the universality of dynamics approach quantum critical points and the effects of integrability on the thermalization process [10]. The natural setting for studying dynamics near critical points is a quantum quench where one of the parameters of the Hamiltonian is driven through a quantum critical point following a time dependent protocol, for example

$$g\left(t\right) \;=\; g_0+\frac{v\left(t-t_0\right)^r}{r!}\theta\left(t-t_0\right)\,, \tag{7.1}$$

with $\theta\left(t\right)$ the step function. Such quenches pose a special difficulty for numerical studies as by definition they involve evolution with a time-dependent Hamiltonian which does not commute with itself at different times. The propagator is then generally a time-ordered exponential whose precise form may be difficult to ascertain.

Standard methods such as the Suzuki-Trotter expansion which ignore the time dependence of the Hamiltonian require [11] that the infinitesimal time step used be much less than the fluctuation time-scale of $H(t)$ to be valid, $\delta t \ll |\partial H/\partial t|^{-1}$. This can cause simulations with rapid quench rates to become numerically very costly, and invalidates the approach altogether for non-analytic time dependence.

Currently the only unbiased method available for the dynamics of quantum systems is exact diagonalization (ED). By unbiased, we refer to the fact that the other methods available for dynamics are generally variational, and so have a bias towards a particular ansatz. ED is limited in an essential way by the exponential growth of the size of the Hilbert space with the physical size of the system. The current state of the art is $\sim$ 40 spins for spin-1/2 models and 20 sites for a fermionic Hubbard model at half filling. These sizes are often too small for accurate finite-size scaling. An extremely powerful method for the low-energy properties of 1D systems is White's Density Matrix Renormalization Group (DMRG) algorithm, which uses a variational ansatz based on a class of states known as Matrix Product States (MPSs). MPSs will be reviewed in Sec. 7.2.1. DMRG uses an iterative procedure to develop a set of reduced bases that the full many-body problem is projected into, and then variationally minimizes an energy functional in this reduced space, enlarging it if necessary. DMRG uses an implicit MPS representation, which is to say that the state is not stored explicitly. This also means that the Hamiltonian and other operators in the calculation are stored in an approximate way, as they are represented within the reduced basis describing the variational state. This does not cause problems in practice when a single state is sought using the DMRG process. In fact, one can show that the algorithm to variationally find the ground state is identical when phrased in the implicit formulation of DMRG and when using an explicit MPS representation for the variational state, other than the representation of the Hamiltonian [12]. However, because of the exact representation of operators independent of the state, MPSs can put rigorous bounds on distances from exact quantum states such as eigenstates by considering quantities like the variance $\langle\psi|\left(\hat{H}-E\right)^2|\psi\rangle$ with $\hat{H}$ the Hamiltonian operator and E the energy expectation of the MPS $|\psi\rangle$. In contrast, DMRG can only return the distance of the variational state from the approximate operator $\hat{H}$ in the given variational basis, and is unable to determine how well the given variational basis represents the true operator. A particularly clear indication of the failures this can cause is given in Ref. [13] where time evolution of a particular initial state in DMRG fails because the Hamiltonian has no nonzero matrix elements in the initial DMRG basis.

The situation becomes much different when multiple states are sought using the DMRG procedure. In this case the reduced density matrix used to determine the optimal reduced bases for the algorithm is a convex sum of the reduced density matrices for the desired states. This is called *multi-state targeting*. In contrast, an explicit MPS representation stores each of the desired states separately as an MPS. In multi-state targeting, none of the states can be represented with the same accuracy available if DMRG targeted that state alone. The MPS representation also deals automatically with the fact that each state has its own optimal bases for representation, whereas in DMRG these bases are all tied together by the multi-state targeting. In this work we present two algorithms which take advantage of MPSs' ability to deal with multiple states, eMPS to find excited states of 1D Hamiltonians and tMPS to simulate the dynamics of a generic time-dependent Hamiltonian. In the first algorithm a projector orthogonal to a set of lower-lying eigenstates is constructed from their MPS representations and used to orthogonalize a variational state against this set. In the second algorithm Krylov vectors in a Lanczos approximation to the matrix exponential are stored separately as MPSs and combined in an optimal way only at the end of the calculation. While Krylov-based MPS approaches have been used [13, 14] to study time-dependent processes, the errors in these approaches were set by time derivative of the Hamiltonian. In contrast, by taking explicit account of the time ordering of the propagator, the errors in our approach are set only by commutators of the Hamiltonian at different times, and hence allow for larger time steps. Because of the explicit MPS representation, we are able to put bounds on the errors in each step of the calculations.

Finally, MPSs have a natural operator-valued extension known as Matrix Product Operators (MPOs) which allow for the exact representation of all operators used in the calculation. We present a general framework for constructing MPOs from a set of rules which is independent of the nature of the microscopic degrees of freedom. This allows for the templating of 1D Hamiltonians for general purpose software. In addition, the ability to perform arithmetic operations on MPOs exactly enables us to perform time-evolution using our tMPS algorithm with knowledge only of the MPO form of the Hamiltonian and the time-dependent functional form of its parameters. To emphasize the general nature of our algorithms, we include a generic simulation protocol for the out-of-equilibrium dynamics of strongly correlated 1D systems using the algorithms presented in this paper.

The remainder of this paper is organized as follows. In Sec. 7.2 we review the theory of MPSs, MPOs, and their canonical forms. In addition to providing a canonical form for operators within the matrix product formalism, we define finite state automaton rules for MPOs and demonstrate how 1D Hamiltonians can be constructed from a small set of such rules. In Sec. 7.3 we review the algorithm for finding ground states using MPSs as variational ansätze, and in Sec. 7.4 we present the eMPS algorithm which extends the ground state search to general excited states. Sec. 7.5 discusses how to extract observable quantities from MPSs. In Sec. 7.6 we discuss methods for time evolution with MPS. In particular, we provide the tMPS algorithm to time evolve an MPS using only the MPO representation of a Hamiltonian and the functional form of its time-dependent parameters. We contrast our approach with other

Krylov subspace approaches and identify the possible sources of error. In Sec. 7.10 we present two case studies. The first is of the Ising model in a transverse field, where we study both the statics and the dynamics of a linear quench of the transverse field through the quantum critical point. The second is of a dipolar, or long-range Ising model in a transverse field, where we also determine the critical point from the statics and study a linear quench of the transverse field. Finally, in Sec. 7.11, we conclude. Details concerning numerically exact solutions for the Ising model which are used to benchmark our algorithms are given in 7.12.

7.2 Brief Review of Matrix Product Formalism

7.2.1 Matrix Product States

The Hilbert space of a quantum mechanical many-body system is exponentially large in the physical size of the system, for example the number of unit cells in a lattice or the number of particles. Stated another way, a state picked at random from the Hilbert space of a quantum many-body system will have entanglement (as quantified by the Schmidt measure [15]) which grows exponentially with the system size. In contrast to this random state, it has been shown that the class of states which are physically relevant in the sense that they can be prepared from some reference state by generic time evolution in polynomial time [11] or are useful for quantum computation [16] is much smaller than the full Hilbert space. In 1D, the physically relevant class of states appears to be those which have entanglement which is either constant or polynomially growing as a function of system size. A convenient representation of states with entanglement restricted in this manner is known as *matrix product states* (MPSs) [12, 17, 18].

We consider our physical system to be comprised of a 1D lattice of *sites*, where each site i is a d-dimensional Hilbert space $\mathbb{H}_i$ spanned by the vectors $\{|i\rangle\}$. We will refer to d as the *local dimension*, and take all sites to be identical for simplicity. We define an MPS on a lattice with L sites as

$$|\psi_{\mathrm{MPS}}\rangle = \sum_{i_1,\dots i_L=1}^{d} \mathrm{Tr}\left(A^{[1]i_1}\dots A^{[L]i_L}\right)|i_1\dots i_L\rangle\,, \tag{7.2}$$

where the object $A^{[k]i_k}$ is a $\chi_k \times \chi_{k+1}$ matrix (with $\chi_1 = \chi_{L+1}$) and Tr denotes the matrix trace. We will refer to the maximum linear dimension of any of the matrices $A^{[k]i_k}$, $\max_k \chi_k$, as the *bond dimension* of the MPS, and denote this quantity by χ. χ may be used as a measure of the entanglement of the state [12]. In this work we will focus on systems with open boundary conditions (OBC). MPS algorithms can also be devised for systems with periodic boundary conditions, as discussed in Refs. [7, 19–21], but these algorithms have worse scaling and are generally less numerically stable than their OBC counterparts. For OBC, $\chi_1 = \chi_{L+1} = 1$, and arguments using the Schmidt decomposition demonstrate that $\chi_k \le \min(d^{k-1}, d^{L-k})$ [22].

MPSs have been used for many years to represent exact ground states of parent Hamiltonians [23] which are formed from projectors onto local high-symmetry subspaces [24, 25]. However, it was not until the great success of the density matrix renormalization group algorithm (DMRG) pioneered by White [26] that MPSs became valuable as variational *ansätze* in their own right [19]. Why are MPSs useful as variational *ansätze*? It has been shown [27] that the ground states of gapped 1D systems have bipartite entanglement which does not depend on the system size. Such states can be represented exactly as MPSs with a fixed bond dimension [28]. This is an example of an *area law* [29]; the entanglement between two disjoint subsystems depends only on the boundary of the two regions and not on their volume. For systems near a quantum critical point which is described by a conformal field theory (CFT), this area law is subject to weak logarithmic violations such that the entropy of entanglement between two subsystems of size L is given by the Calabrese-Cardy formula [30, 31]

$$S_L \sim a + \frac{c}{6} \log L \,, \tag{7.3}$$

where a is a constant and c is the central charge of the underlying CFT. Here $\sim$ denotes scaling equivalence in the bulk of an infinite system. In finite systems there are often oscillating boundary and finite size contributions [32, 33]. Hence, the bond dimension of an MPS describing a conformally invariant critical system is given as $\chi_L \sim \exp S_L \sim e^a L^{c/6}$, which grows only polynomially in the system size. Typical values of c range from 1/2 for the Ising model [34] and 1 for the Bose-Hubbard model [35] to 2 for more exotic phases like the gapless Mott insulator of the JK model [36]. We note that, strictly speaking, finding an MPS which approximates the ground state of an arbitrary 1D Hamiltonian to an accuracy which is an inverse polynomial in the system size is still NP-complete [37], but practical experience demonstrates that this method is extremely useful and robust for physical systems of interest.

We adopt the following conventions for the representation of tensors: we use roman indices for indices which correspond to physical states and greek indices for indices which are summed over in the matrix-product ansatz. Explicitly writing out Eq. (7.2) with these indices, we have

$$|\psi_{\mathrm{MPS}}\rangle = \sum_{\alpha_1 \dots \alpha_{L-1}} \sum_{i_1, \dots i_L = 1}^{d} A^{[1]i_1}_{\alpha_1 \alpha_2} \dots A^{[L]i_L}_{\alpha_{L-1}\alpha_1} |i_1 \dots i_L\rangle \,. \tag{7.4}$$

A superscript index in square brackets [] denotes the lattice site that the physical indices of the tensor describe. A superscript index in curly braces {} denotes association with a particular many-body state. For example, the MPS tensors at site j of the MPSs $|\phi_k\rangle$, $k = 1, \dots, n$ would be denoted $A^{[j]\{k\}i_j}_{\alpha\beta}$. Finally, indices which appear together in parentheses, e.g. $(\alpha\beta)$, represent a composite index which runs over the Cartesian product of the indices in the parentheses. As an example, if $\alpha = 1, \dots, \chi_\alpha$ and $\beta = 1, \dots \chi_\beta$, $(\alpha\beta) = (\alpha - 1)\chi_\beta + \beta$. To lighten the notation, we will leave off

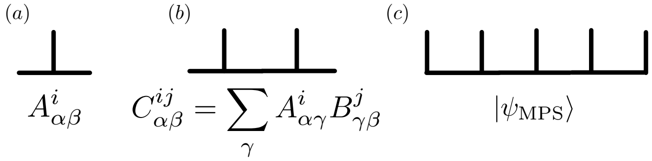


Figure 7.1: *Diagrammatic notation for tensor networks.* (a) A rank-three tensor is represented as a point with three lines extending from it. (b) Contraction of two rank-three tensors is accomplished by connecting the contracted index, and produces a tensor of rank four. (c) An MPS on 5 sites with open boundary conditions is represented as a contraction over rank three tensors with two rank-two boundary tensors.

indices when they are unnecessary.

A particularly useful means to visualize MPSs and manipulations with them is provided by *tensor network* diagrams like those shown in Figure 7.1 [38]. Here a rank-k tensor is represented by a point with k lines extending from it. Each line represents one of the indices of the tensor. Whenever a line connects two points, that index is summed over, and disconnected lines represent free indices. Hence, an MPS can be represented as a chain of rank-3 tensors as in Figure 7.1(c). Note that the first and last MPS tensors are rank two because we have assumed OBC and so $\chi_1 = \chi_{L+1} = 1$.

We note that the MPS definition Eq. (7.2) does not uniquely specify the tensors A. That is, we can insert an invertible matrix X and its inverse X^{-1} between any two MPS matrices without altering the physical content of the state:

$$|\psi_{\rm MPS}\rangle = \sum_{i_1,\dots i_L=1}^{d} \mathrm{Tr}\left(A^{[1]i_1}\dots A^{[L]i_L}\right)|i_1\dots i_L\rangle\,, \tag{7.5}$$

$$\tilde{A}^{[j]i_j} = A^{[j]i_j}X\,,\quad \tilde{A}^{[j+1]i_{j+1}} = X^{-1}A^{[j+1]i_{j+1}}\,, \tag{7.6}$$

$$|\tilde{\psi}_{\rm MPS}\rangle = \sum_{i_1,\dots i_L=1}^{d} \mathrm{Tr}\left(A^{[1]i_1}\dots \tilde{A}^{[j]i_j}\tilde{A}^{[j+1]i_{j+1}}\dots A^{[L]i_L}\right)|i_1\dots i_L\rangle = |\psi_{\rm MPS}\rangle \tag{7.7}$$

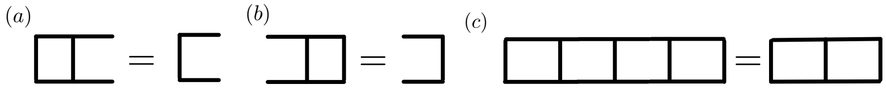


Figure 7.2: *Gauge conditions for MPSs.* (a) Left gauge condition Eq (7.8). (b) Right gauge condition Eq (7.9). (c) Using (a) and (b), the norm squared of an MPS reduces to the trace of its orthogonality center squared, here chosen to be the third site.

This is referred to as *gauge freedom* in the literature [22]. For OBC, we can specify the state uniquely[51] by choosing a site k, which we call the *orthogonality center* of the MPS, and requiring that all sites i to the left and right of k satisfy the left

$$\sum_i A^{i\dagger} A^i \quad = I \tag{7.8}$$

and right

$$\sum_i A^i A^{i\dagger} \quad = I \tag{7.9}$$

gauge conditions, respectively. In these expressions, I is the appropriately dimensioned identity matrix. These conditions are shown in graphical notation in Figure 7.2(a) and (b), respectively.[52] Graphically it is clear that the norm squared of the state is

$$\langle\psi|\psi\rangle = \sum_i \mathrm{Tr}\left(A^{[k]i\dagger} A^{[k]i}\right) , \tag{7.10}$$

as shown in Figure 7.2(c), and so this site carries all information about the norm of the state. This particular canonical form for an MPS is called *mixed canonical form* [12]. The mixed canonical form is crucial for improving the speed and numerical stability of variational algorithms with MPSs.

We can impose the left gauge conditions via the following recursion:

---

[51] The state is unique up to possible degeneracies in the Schmidt decomposition.

[52] Here we also establish the graphical convention that downwards pointing lines correspond to Hermitian conjugates of tensors.

$$\begin{aligned}
\tilde{A}_{(\alpha i)\beta} &= A^{[\ell]i}_{\alpha\beta}\,, && (7.11)\\
\sum_{\gamma} U_{(\alpha i)\gamma}\Sigma_{\gamma}V^{\dagger}_{\gamma\beta} &= \tilde{A}_{(\alpha i)\beta}\,, && (7.12)\\
A^{[\ell]i}_{\alpha\gamma} &= U_{(\alpha i)\gamma}\,, && (7.13)\\
A^{[\ell+1]i}_{\gamma\eta} &= \textstyle\sum_{\beta}\Sigma_{\gamma}V^{\dagger}_{\gamma\beta}A^{[\ell+1]i}_{\beta\eta}\,, && (7.14)
\end{aligned}$$

where Eq. (7.12) represents the singular value decomposition (SVD) of $\tilde{A}$ with Σ a diagonal matrix of singular values and U and V unitary matrices.[53] Because U returned from the SVD is unitary, Eq. (7.8) is satisfied by construction. Similarly, the recursion for the right gauge conditions is

$$\begin{aligned}
\tilde{A}_{\alpha(i\beta)} &= A^{[\ell]i}_{\alpha\beta}\,, && (7.15)\\
\sum_{\gamma} U_{\alpha\gamma}\Sigma_{\gamma}V^{\dagger}_{\gamma(i\beta)} &= \tilde{A}_{\alpha(i\beta)}\,, && (7.16)\\
A^{[\ell]i}_{\gamma(i\beta)} &= V^{\dagger}_{\gamma(i\beta)}\,, && (7.17)\\
A^{[\ell-1]i}_{\eta\gamma} &= \textstyle\sum_{\alpha}A^{[\ell-1]i}_{\eta\alpha}U_{\alpha\gamma}\Sigma_{\gamma}\,. && (7.18)
\end{aligned}$$

To put a general state into mixed canonical form with orthogonality center k we begin at site 1 and iterate Eqs. (7.11)-(7.14) until we reach site k, then start at site L and iterate Eqs. (7.15)-(7.18) until we again reach site k.

Finally, we note that the set of all MPSs with a fixed bond dimension χ is not a vector space, as the sum of two MPSs with bond dimensions χ_A and χ_B has a bond dimension χ which is bounded by the sum of the two bond dimensions $\chi \le \chi_A + \chi_B$. This can be seen from considering the sum of the two states $|0\ldots 0\rangle$ and $|1\ldots 1\rangle$, with MPS representations

---

[53] Note that any matrix decomposition of $\tilde{A}$ which returns a unitary matrix as part of the decomposition will suffice in place of the SVD. In particular, the QR decomposition [39] is particularly efficient when the rank of $\tilde{A}$ is not required.

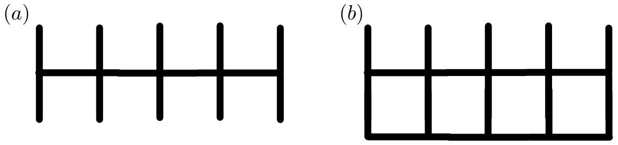


Figure 7.3: *MPO in diagrammatic notation.* (a) A matrix product operator consists of a contraction of rank-four tensors. We adopt the graphical convention that the line below the horizontal corresponds to i' and the line above the horizontal to i for an operator $O_{ii'}$ which takes the local state from $|i'\rangle$ to $|i\rangle$. (b) The product of an MPO and an MPS produces another MPS whose bond dimension is the product of the bond dimensions of the original MPS and the MPO.

$$|0\dots 0\rangle = \sum_{i_1\dots i_L} A^{i_1}\dots A^{i_L}|i_1\dots i_L\rangle\,, \quad A^i = (\delta_{i,0})\ , \tag{7.19}$$

$$|1\dots 1\rangle = \sum_{i_1\dots i_L} B^{i_1}\dots B^{i_L}|i_1\dots i_L\rangle\,, \quad B^i = (\delta_{i,1})\ . \tag{7.20}$$

The matrices A^i and B^i have a bond dimension of 1, as these are product states. Their sum is

$$|0\dots 0\rangle + |1\dots 1\rangle = \sum_{i_1\dots i_L} C^{[1]i_1}C^{[2]i_2}\dots C^{[L-1]i_{L-1}}C^{[L]i_L}|i_1\dots i_L\rangle\,, \tag{7.21}$$

$$C^{[1]i} = \left(\begin{array}{cc}\delta_{i0} & \delta_{i1}\end{array}\right)\ ; \quad C^{[j]i} = \left(\begin{array}{cc}\delta_{i0} & 0\\ 0 & \delta_{i1}\end{array}\right)\,, 2\le j\le L-1\,; \quad C^{[L]i} = \left(\begin{array}{c}\delta_{i0}\\ \delta_{i1}\end{array}\right)\,, \tag{7.22}$$

which has a bond dimension of 2.

7.2.2 Matrix Product Operators

The natural operator generalization of MPSs is a Matrix Product Operator (MPO), defined as

$$\hat{O}_{\mathrm{MPO}} = \sum_{i_1,\dots i_L=1}^{d}\sum_{i'_1,\dots i'_L=1}^{d} \mathrm{Tr}\left(W^{[1]i_1i'_1}\dots W^{[L]i_Li'_L}\right)|i_1\dots i_L\rangle\langle i'_1\dots i'_L|\,. \tag{7.23}$$

Here $W^{[k]ii'}$ is a $\chi_k^O \times \chi_{k+1}^O$ dimensional matrix, and we will again refer to the maximum value of χ^O as the bond dimension of the operator. Note that this bond dimension χ^O need not be the same as the bond dimension χ appearing in the MPS representation, Eq. (7.2). That an MPO takes MPSs to MPSs can be seen clearly from the graphical representation of Figure 7.3. We also see from this representation that the bond dimension of the MPS representing the product of an MPO and an MPS is the product of the bond dimensions of the MPO and the MPS. Because the states $|i_1 \dots i_L\rangle$ are tensor products, we can also use the notation

$$\hat{O}_{\mathrm{MPO}} = \sum_{i_1,\dots i_L=1}^{d} \sum_{i'_1,\dots i'_L=1}^{d} \mathrm{Tr}\left(\mathcal{W}^{[1]i_1 i'_1} \dots \mathcal{W}^{[L]i_L i'_L}\right) , \tag{7.24}$$

where each one of the objects $\mathcal{W}^{[j]i_j i'_j} \equiv W^{[j]i_j i'_j} |i_j\rangle\langle i'_j|$. That is to say, we can consider the matrices which appear in the matrix-product ansatz of an MPO to be operator-valued.

It is tempting to look for canonical forms for MPOs just as we did for MPSs, but the relevant norm for MPOs is the Frobenius norm $\langle \hat{O}_1, \hat{O}_2\rangle = \mathrm{Tr}(\hat{O}_1^\dagger \hat{O}_2)$ which scales exponentially in the local dimension with the number of lattice sites. Thus, for a typical many-body system with an exponentially large Hilbert space, the elements of the orthogonality center can differ in magnitude greatly, causing a catastrophic loss of precision during orthogonalization. However, most physically relevant MPOs such as one-dimensional Hamiltonians can be written down exactly in terms of an MPO canonical form which is analogous to an LU decomposition [40]. To discuss this canonical form, it is useful to recast an MPO as a *finite state automaton* (FSA) [41, 42].

To recast an MPO as a FSA, we first enumerate all of the physical operators we use to define our local Hilbert space, $\mathcal{O} = \{\hat{O}^\alpha\}$, where α labels distinct operators. We will call this set our *operator alphabet* (OA). As examples, the OA for the Ising model would be $\left\{\hat{I},\hat{\sigma}^x,\hat{\sigma}^z\right\}$ with $\hat{I}$ the identity operator and $\hat{\sigma}^{\nu}$ the Pauli operator along the ν^{th} Cartesian direction. The OA for the Bose-Hubbard model [43, 44] would be $\left\{\hat{I},\hat{b}^{\dagger},\hat{b},\hat{n}\right\}$, where $\hat{b}$ is a bosonic destruction operator and $\hat{n}=\hat{b}^{\dagger}\hat{b}$ the number operator. The particular matrix representation of the OA fixes the local basis states $\{|i\rangle\}$.

Using the OA, we now introduce a set of FSA *rules* $\{\mathcal{R}_p\left(\{\hat{O}^{p_1},\dots,\hat{O}^{p_n}\},\{h^p\},w^p\right)\}$ which generate strings of the n operators $\{\hat{O}^{p_1},\dots,\hat{O}^{p_n}\}\in\mathcal{O}$ weighted by scalar w_p and variables $\{h_p\}$. We will call the variables $\{h^p\}$ *Hamiltonian parameters* and the scalars w^p *weights*. Each rule consists of three operator-valued matrices in the case where the $\{h_p\}$ do not depend on position. The first matrix is the rightmost matrix in the MPO representation, $\mathcal{W}^{[L]}$, and represents the initial configuration. Next, we have the MPO matrix of the bulk of the chain $\mathcal{W}^{[j]}$, $2\le j\le L-1$, which takes an input vector of operators on k sites and produces a vector of operators on $k+1$ sites according to some deterministic pattern. Finally, we have the leftmost matrix in the MPO representation, $\mathcal{W}^{[1]}$, which extends the operators according to the patten of $\mathcal{W}^{[j]}$ and returns a 1×1 operator-valued matrix. The trace of this matrix as in Eq (7.23) is the desired Hamiltonian term. The generalization to position-dependent Hamiltonian parameters requires $L-2$ matrices in place of the bulk matrix $\mathcal{W}^{[j]}$, $2\le j\le L-1$, but the only modification is that h_p becomes $h_p\left(j\right)$.

As a concrete example, consider the Ising model

$$\hat{H}=-J\sum_{\langle i,j\rangle}\hat{\sigma}^z_i\hat{\sigma}^z_j-h\sum_i\hat{\sigma}^x_i\,. \tag{7.25}$$

The Hamiltonian consists of two rules. The first is a site rule $\mathcal{R}_{\mathrm{site}}(\hat{\sigma}^x,h,-1)$ which generates the string $-h\sum_i\hat{\sigma}^x_i$. The three matrices which provide this rule are

$$\begin{aligned} \mathcal{W}_{\text{site}}^{[1]} &= \begin{pmatrix} -h\hat{\sigma}^x & \hat{I} \end{pmatrix}, \quad \mathcal{W}_{\text{site}}^{[2\le j\le L-1]} = \begin{pmatrix} \hat{I} & 0 \\ -h\hat{\sigma}^x & \hat{I} \end{pmatrix}, \\ \mathcal{W}_{\text{site}}^{[L]} &= \begin{pmatrix} \hat{I} \\ -h\hat{\sigma}^x \end{pmatrix}. \end{aligned} \tag{7.26}$$

As can be verified,

$$\prod_{j=L-1}^{k} \mathcal{W}_{\text{site}}^{[j]} \mathcal{W}_{\text{site}}^{[L]} = \begin{pmatrix} \hat{I}\ldots\hat{I} \\ -h\sum_{i=k}^{L}\hat{\sigma}_i^x \end{pmatrix}, \tag{7.27}$$

and so this rule produces the desired operator. Similarly, there is a bond rule $\mathcal{R}_{\text{bond}}(\{\hat{\sigma^z}, \hat{\sigma}^z\}, J, -1)$ given by

$$\begin{aligned} \mathcal{W}_{\text{bond}}^{[1]} &= \begin{pmatrix} 0 & -J\hat{\sigma}^z & \hat{I} \end{pmatrix}, \quad \mathcal{W}_{\text{bond}}^{[2\le j\le L-1]} = \begin{pmatrix} \hat{I} & 0 & 0 \\ \hat{\sigma}^z & 0 & 0 \\ 0 & -J\hat{\sigma}^z & \hat{I} \end{pmatrix}, \\ \mathcal{W}_{\text{bond}}^{[L]} &= \begin{pmatrix} \hat{I} \\ \hat{\sigma}^z \\ 0 \end{pmatrix}, \end{aligned} \tag{7.28}$$

which produces $-J\sum_{\langle i,j\rangle}\hat{\sigma}_i^z\hat{\sigma}_j^z$, with $\langle i,j\rangle$ denoting a sum over nearest neighbors i and j. The full Hamiltonian is given by the direct sum of the matrices. Collecting rows of the direct sum which are exactly parallel, we have the MPO representation of the full operator

$$\begin{aligned} \mathcal{W}_{\text{Ising}}^{[1]} &= \begin{pmatrix} -h\hat{\sigma}^x & -J\hat{\sigma}^z & \hat{I} \end{pmatrix}, \quad \mathcal{W}_{\text{Ising}}^{[2\le j\le L-1]} = \begin{pmatrix} \hat{I} & 0 & 0 \\ \hat{\sigma}^z & 0 & 0 \\ -h\hat{\sigma}^x & -J\hat{\sigma}^z & \hat{I} \end{pmatrix}, \\ \mathcal{W}_{\text{Ising}}^{[L]} &= \begin{pmatrix} \hat{I} \\ \hat{\sigma}^z \\ -h\hat{\sigma}^x \end{pmatrix}. \end{aligned} \tag{7.29}$$

This construction can be readily extended to general sums of nearest-neighbor and on-site interactions with Hamiltonian

$$\hat{H} = -\sum_{\alpha=1}^{n_b} J_\alpha \sum_{\langle i,j \rangle} \hat{O}_i^{b_{1;\alpha}} \hat{O}_j^{b_{2;\alpha}} - \sum_{\beta=1}^{n_s} h_\beta \sum_i \hat{O}_i^{s_\beta} \tag{7.30}$$

by summing up the individual rules:

$$\begin{aligned}
\mathcal{W}^{[1]} &= \begin{pmatrix} -\sum_{\beta=1}^{n_s} h_\beta \hat{O}^{s_\beta} & -J_1 \hat{O}^{b_{1;1}} & \dots & -J_{n_b} \hat{O}^{b_{1;n_b}} & \hat{I} \end{pmatrix} \\
\mathcal{W}^{[2\le j\le L-1]} &= \begin{pmatrix} \hat{I} & 0 & \dots & 0 & 0 \\ \hat{O}^{b_{2;1}} & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \hat{O}^{b_{2;n_b}} & 0 & \dots & 0 & 0 \\ -\sum_{\beta=1}^{n_s} h_\beta \hat{O}^{s_\beta} & -J_1 \hat{O}^{b_{1;1}} & \dots & -J_{n_b} \hat{O}^{b_{1;n_b}} & \hat{I} \end{pmatrix}, \\
\mathcal{W}^{[L]} &= \begin{pmatrix} \hat{I} \\ \hat{O}^{b_{2;1}} \\ \vdots \\ \hat{O}^{b_{2;n_b}} \\ -\sum_{\beta=1}^{n_s} h_\beta \hat{O}^{s_\beta} \end{pmatrix}.
\end{aligned} \tag{7.31}$$

The bond dimension of the MPO representation of this Hamiltonian is $2 + n_B$. MPO representations are not restricted to nearest-neighbor terms. Exponentially decaying terms of the form[54]

$$\hat{H} = -J \sum_{i<j} e^{-\lambda(j-i-1)} \hat{O}_i^1 \hat{O}_j^2 \tag{7.32}$$

can also be accommodated with the rule $\mathcal{R}_{\exp}(\{\hat{O}^1, \hat{O}^2\}, \{J, \lambda\}, -1)$

$$\begin{aligned}
\mathcal{W}_{\exp}^{[1]} &= \begin{pmatrix} 0 & -J\hat{O}^1 & \hat{I} \end{pmatrix}, \quad \mathcal{W}_{\exp}^{[2\le j\le L-1]} = \begin{pmatrix} \hat{I} & 0 & 0 \\ \hat{O}^2 & e^{-\lambda}\hat{I} & 0 \\ 0 & -J\hat{O}^1 & \hat{I} \end{pmatrix}, \\
\mathcal{W}_{\exp}^{[L]} &= \begin{pmatrix} \hat{I} \\ \hat{O}^2 \\ 0 \end{pmatrix}.
\end{aligned} \tag{7.33}$$

We can also produce a general monotonically decaying term

---

[54] Note in this form that the nearest neighbor coupling is J and only longer ranged couplings contain λ.

$$\hat{H} = J \sum_{i<j} f\left(j-i\right) \hat{O}^1_i \hat{O}^2_j \tag{7.34}$$

by approximating the term as a sum of exponentials to a desired distance r_{cutoff}. That is, we minimize the functional

$$F\left(\mathbf{a}, \mathbf{b}\right) = \sum_{i=1}^{n_{\mathrm{exp}}} \sum_{x=1}^{r_{\mathrm{cutoff}}} \left| f\left(x\right) - a_i b_i^{x-1} \right|^2 \tag{7.35}$$

with respect to the exponential weights a_i and decay parameters b_i, where $\mathbf{a}$ and $\mathbf{b}$ are the elements a_i and b_i, respectively, arranged as vectors and n_{exp} is a convergence parameter controlling the number of exponentials used in the expansion [45–47]. Note that the fit is only guaranteed to be accurate to r_{cutoff} while the term has infinite range. This does not cause difficulties in practice for decaying functions, as the resulting fit is also guaranteed to be monotonically decaying. The decaying function rule is then just a sum of these n_{exp} exponential rules, $\mathcal{R}_{\mathrm{df}}(\{\hat{O}^1, \hat{O}^2\}, \{J, f(x), n_{exp}, r_{\mathrm{cutoff}}\}, 1) = \sum_{i=1}^{n_{\mathrm{exp}}} \mathcal{R}_{\mathrm{exp}}(\{\hat{O}^1, \hat{O}^2\}, \{Ja_i, -\log b_i\}, 1)$. In contrast to the other rules presented above this rule is not exact, but the number of exponentials can be increased to any desired accuracy.[55] This procedure is surprisingly accurate even for small numbers of exponentials-5 exponentials suffice to accurately represent a $1/r^3$ interaction at a distance of 1000 sites to an error of 10^{-9} [47]. While the bond dimension increases linearly with the number of exponentials, the number of nonzero terms in the MPO also grows only linearly with the number of exponentials, and so operations with the MPO scale well as n_{exp} increases. However, the eigenstate of a Hamiltonian with larger n_{exp} may be more highly entangled than with smaller n_{exp}, leading to an increase in χ and longer runtimes. That is to say, the relationship between the bond dimension of an MPO and the bond dimension of an MPS representing an eigenstate of the MPO is difficult to predict.

---

[55] By accuracy we mean that the functional Eq. (7.35) is smaller than a given tolerance.

While the pure functional interaction Eq. (7.34) is appealing from a theoretical point of view, in practical applications infinite range interactions do not represent a consistent level of approximation. That is to say, at some distance interactions are screened, where the screening length is set by energetics or lifetime constraints in the case of ultracold atoms or molecules [48]. Hence, we provide the finite-ranged rule $\mathcal{R}_{\mathrm{FiniteFunction}}(\{\hat{O}^1, \hat{O}^2\}, \{h, f, r_{\mathrm{cutoff}}\}, w)$ which generates the Hamiltonian

$$\hat{H} = wh \sum_i \sum_{j=i+1}^{i+r_{\mathrm{cutoff}}} f\left(j-i\right) \hat{O}_i^1 \hat{O}_j^2 \tag{7.36}$$

and is given by

$$\begin{aligned}
\mathcal{W}^{[1]} &= \begin{pmatrix} 0 & 0 & \dots & 0 & \dots & 0 & whf\left(1\right)\hat{O}^1 & \hat{I} \end{pmatrix} \\
\mathcal{W}^{[2\le j\le L-1]} &= \begin{pmatrix}
\hat{I} & 0 & \dots & 0 & \dots & 0 & 0 & 0 \\
\hat{O}^2 & 0 & \dots & 0 & \dots & 0 & 0 & 0 \\
\hat{O}^2 & \frac{f(r_{\mathrm{cutoff}})}{f(r_{\mathrm{cutoff}}-1)}\hat{I} & \dots & 0 & \dots & 0 & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \vdots \\
\hat{O}^2 & 0 & \dots & \frac{f(k)}{f(k-1)} & \dots & 0 & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \vdots \\
\hat{O}^2 & 0 & \dots & 0 & \dots & \frac{f(2)}{f(1)}\hat{I} & 0 & 0 \\
0 & 0 & \dots & 0 & \dots & 0 & whf\left(1\right)\hat{O}^1 & \hat{I}
\end{pmatrix}, \\
\mathcal{W}^{[L]} &= \begin{pmatrix} \hat{I} \\ \hat{O}^2 \\ \hat{O}^2 \\ \vdots \\ \hat{O}^2 \\ \vdots \\ \hat{O}^2 \\ 0 \end{pmatrix}.
\end{aligned} \tag{7.37}$$

For a vector f which contains zero elements a small modification must be made, but the bond dimension remains the same.

A small set of rules like those presented above allow us to template operators for MPS simulations. That is, given a method to produce a Hamiltonian from a set of basic rules, we can generate a wide variety of different Hamiltonians by specifying different OAs, weights, and Hamiltonian parameters as inputs to a program. The resulting MPOs are lower triangular and usually very sparse, and so sparse matrix structures can be used for efficiency. Beyond the flexibility that MPOs provide to MPS algorithms, they also allow for arithmetic operations such as addition and multiplication to be performed exactly, albeit at the expense of a growing bond dimension. The sum of two operators expressed as MPOs has MPO matrices which are the direct sums of the constituent matrices, and the product of two MPOs has MPO matrices which are the direct product of the constituent MPO matrices. Both of these operations preserve the sparse lower triangular structure of the MPOs.

7.3 Variational Ground State Search

We now turn to using MPSs as variational ansätze for the eigenstates of a Hamiltonian expressed as an MPO. The ground state is found by minimizing the functional

$$\mathcal{E}\left[|\psi\rangle\right] = \langle\psi|\hat{H}|\psi\rangle - E\langle\psi|\psi\rangle \tag{7.38}$$

with respect to the parameters of $|\psi\rangle$, where E is a Lagrange multiplier enforcing normalization. The general minimization of this functional is an NP-hard problem, so we instead adopt a local search heuristic that has proven to work well in practice. Let us consider fixing all parameters in the MPS except for a contiguous block of s MPS tensors $A^{[j]}\dots A^{[j+s-1]}$. The single-site ($s=1$) and two-site ($s=2$) algorithms are the most commonly used variants. We then find the extremum as

$$\frac{\partial}{\partial A^{[j]\star}\dots A^{[j+s-1]\star}}\Big(\langle\psi|\hat{H}|\psi\rangle - E\langle\psi|\psi\rangle\Big) = 0 \tag{7.39}$$

which corresponds to the diagrammatic equation shown in Figure 7.4. Here, the partial derivative with respect to a tensor is defined to be a tensor whose elements are the partial derivatives with respect to the elements of the tensor. If we assume that the block of tensors to be optimized contains the orthogonality center then the rightmost diagram reduces to the block of tensors being optimized and the leftmost diagram is the action of the *effective Hamiltonian* on this same block. Thus, minimization consists of finding the eigenvector corresponding to the smallest eigenvalue of the effective Hamiltonian eigenvalue problem

$$\hat{H}^{[j]}_{\mathrm{eff}} A^{[j]} \dots A^{[j+s-1]} = E A^{[j]} \dots A^{[j+s-1]}\,, \tag{7.40}$$

where the effective Hamiltonian is

$$\hat{H}^{[j] i_j \dots i_{j+s-1} i'_j \dots i'_{j+s-1}}_{\mathrm{eff}\;\alpha\beta\alpha'\beta'} = \sum_{\kappa\kappa'} \sum_{\gamma_1 \dots \gamma_s} L^{[j]}_{\kappa\alpha\alpha'} W^{[j] i_j i'_j}_{\kappa\gamma_1} \dots W^{[j+s-1] i_{j+s-1} i'_{j+s-1}}_{\gamma_s \kappa'} R^{[j+s]}_{\kappa'\beta'\beta}\,, \tag{7.41}$$

L and R are the partial overlaps of the Hamiltonian MPO with the state as in Figure 7.4, and $W^{[j]}$ is the MPO tensor at site j of the Hamiltonian. $\hat{H}^{[j]}_{\mathrm{eff}}$ represents the Hamiltonian for the variational degrees of freedom in the block to be optimized with the rest of the state held fixed. This justifies our use of E as the eigenvalue, as E obtained from the solution of this equation is the current best estimate for the energy. We can view Eq. (7.40) as a linear eigenvalue problem by combining indices as

$$\begin{aligned} \sum_{\alpha' i'_j \dots i'_{j+s-1} \beta'} \hat{H}_{\mathrm{eff}\;(\alpha i_j \dots i_{j+s-1} \beta)(\alpha' i'_j \dots i'_{j+s-1} \beta')} \left[A^{[j]} \dots A^{[j+s-1]}\right]_{(\alpha' i'_j \dots i'_{j+s-1} \beta')} \\ = E \left[A^{[j]} \dots A^{[j+s-1]}\right]_{(\alpha i_j \dots i_{j+s-1} \beta)}\,. \end{aligned} \tag{7.42}$$

The linear dimension of this matrix representation of $\hat{H}^{[j]}_{\mathrm{eff}}$ is $\chi_j d^s \chi_{j+s-1}$, and so a solution of this problem with dense methods would require $\mathcal{O}(\chi_j^3 d^{3s} \chi_{j+s-1}^3)$ basic operations, leading to a very slow algorithm of order $\mathcal{O}(\chi^6)$. In contrast, by taking advantage of the separable form of the effective Hamiltonian Eq. (7.41) multiplication of our block of tensors by the effective Hamiltonian can be done in $\mathcal{O}(\chi^3)$ time [12]. Thus, sparse eigensolvers such as the Lanczos [39] or Davidson [49] algorithms, which

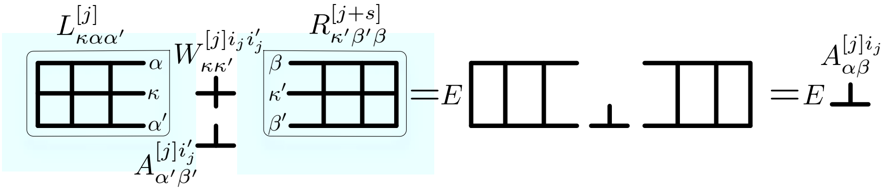


Figure 7.4: *Variational ground state search in diagrammatic notation.* Here we display the single site ($s = 1$) effective Hamiltonian eigenvalue problem for simplicity. The contractions between the block of tensors to be optimized, the MPO, and the rest of the diagram have been omitted to accentuate the structure of the effective Hamiltonian. The rightmost equality follows from assuming that the variational site is the orthogonality center.

require only matrix-vector multiplies, should be employed to solve this eigenvalue problem.

The general algorithm for ground state search is thus as follows. We begin with an initial state with orthogonality center at site k. We choose a block of tensors containing k and optimize them by solving the effective Hamiltonian eigenvalue problem. We then shift the orthogonality center and the block of tensors one site to the right and again optimize. We continue shifting to the right until we reach the right boundary. We then reverse direction, shifting the orthogonality center and the block of tensors to be optimized to the left and solving the effective Hamiltonian eigenvalue problem until we reach the left boundary, at which point we reverse direction again. A single iteration of this optimization cycle which affects each tensor twice is called a *sweep*, and sweeping is continued until convergence. In addition, using the MPO form of the Hamiltonian, it is possible to develop a caching algorithm for the overlaps L and R such that the solution of this problem requires $\mathcal{O}(L)$ scaling in the number of lattice sites [42].[56] To do so, we begin the iteration with a guess for the ground state $|\psi\rangle$ assumed to have orthogonality center k. We then use the left recursion

---

[56] This scaling does not account for possible L dependence of the bond dimension χ such as exists for conformal critical points.

$$
\begin{aligned}
L^{[1]}_{\kappa\alpha\alpha'} &= \delta_{\alpha,1}\delta_{\alpha',1}\delta_{\kappa,1}\,, && (7.43)\\
L^{[\ell+1]}_{\kappa\alpha\alpha'} &= \left[\sum\nolimits_{\gamma' i'}\left[\sum\nolimits_{i\kappa'}\left[\sum\nolimits_{\gamma} A^{[\ell]i\star}_{\gamma\alpha} L^{[\ell]}_{\kappa'\gamma\gamma'}\right] W^{[\ell]ii'}_{\kappa'\kappa}\right] A^{[\ell]i'}_{\gamma'\alpha'}\right]\,, && (7.44)
\end{aligned}
$$

to generate the L overlaps up to k and the right recursion

$$
\begin{aligned}
R^{[L+1]}_{\kappa\beta'\beta} &= \delta_{\beta,1}\delta_{\beta',1}\delta_{\kappa,1}\,, && (7.45)\\
R^{[\ell]}_{\kappa\beta'\beta} &= \left[\sum\nolimits_{\gamma' i'} A^{[\ell]i'}_{\beta'\gamma'}\left[\sum\nolimits_{\kappa' i} W^{[\ell]ii'}_{\kappa\kappa'}\left[\sum\nolimits_{\gamma} R^{[\ell+1]}_{\kappa'\gamma'\gamma} A^{[\ell]i\star}_{\beta\gamma}\right]\right]\right]\,, && (7.46)
\end{aligned}
$$

to generate the R overlaps down to $k+s$. Here the square braces indicate the order in which the contraction should be performed to achieve ideal scaling. Once the eigenvalue problem has been solved and the orthogonality center shifted, we use the recurrence Eq. (7.44) to update the overlaps when we are sweeping to the right and the recurrence Eq. (7.46) to update the overlaps when we are sweeping to the left.

Convergence is achieved when the variance

$$
\Delta \equiv \langle\left(\hat{H}^2 - E^2\right)\rangle\,, \qquad (7.47)
$$

with E the energy eigenvalue, drops below a user-specified tolerance ϵ. Given the MPO form of the Hamiltonian, the variance operator $\hat{\Delta} \equiv \hat{H}^2 - E^2$ can be constructed by constructing an MPO whose matrices $\tilde{W}^{[i]}$ consist of the direct product of the corresponding matrices from $\hat{H}$, $\tilde{W}^{[i]} = W^{[i]} \otimes W^{[i]}$, and then subtracting $-\hat{I}E^2/L$ from the lower leftmost element of each $\tilde{W}^{[i]}$, where $\hat{I}$ is the identity operator. This representation is exact, in contrast to DMRG-based approaches where the basis of the Hamiltonian is tied together with the basis of the state. The variance is a much better measure of convergence of the *state* than the so-called discarded weight which is used to measure convergence of the two-site DMRG algorithm. This is because it is a property of the actual MPS state and not of the eigenvalue. As a note of caution, the variance only guarantees that the state found is *an* eigenstate to the given tolerance, it does not specify that it is the ground state. This has not proven to cause problems in practice for non-disordered systems.

In summary, the complete algorithm for variational ground state search is:

1. *Input*: Input a Hamiltonian $\hat{H}$ in MPO form, an initial guess $|\psi\rangle$ for the ground state in MPS form, and a tolerance ϵ for the variance.

2. *Initialization*: Construct the LR overlaps using the recursions Eq. (7.44) and (7.46).

3. *Sweeping* :

   (a) Solve the effective Hamiltonian eigenvalue problem Eq. (7.40) and replace the variational block of tensors with the eigenvector corresponding to the lowest eigenvalue.

   (b) Shift the orthogonality center to the right and update the LR overlaps using the recursion Eq. (7.44).

   (c) Continue to iterate a and b (*right sweeping*) until the right boundary is reached.

   (d) Solve the effective Hamiltonian eigenvalue problem Eq. (7.40) and replace the variational block of tensors with the eigenvector corresponding to the lowest eigenvalue.

   (e) Shift the orthogonality center to the left and update the LR overlaps using the recursion Eq. (7.46).

   (f) Continue to iterate d and e (*left sweeping*) until the left boundary is reached.

   (g) Iterate a and b until k is reached.

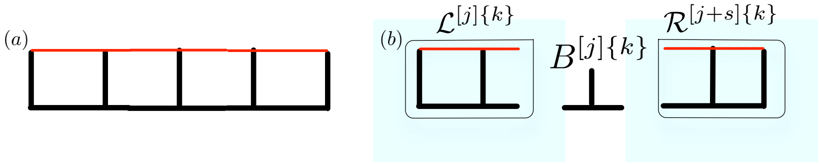


Figure 7.5: *Linear forms enforcing orthogonality.* (a) The overlap $\langle\psi|\phi_k\rangle$ in diagrammatic notation. The thin red lines correspond to $\langle\psi|$ and the thick black lines to $|\phi_k\rangle$. (b) The linear form $F^{[j]\{k\}}$ in diagrammatic notation for the single-site case. As before, we leave the bottom tensor uncontracted to accentuate the definitions of the $\mathcal{LR}$ overlaps.

4. *Check convergence*: Using the most recent estimate of the energy eigenvalue $\tilde{E}$ from the last effective Hamiltonian solution, construct the variance operator $\hat{\Delta}$ and find the variance. If $\Delta < \epsilon$, exit, otherwise return to a.

7.4 Variational Excited State Search

We now turn to finding excited states. We find the n^{th} excited state by minimizing the functional

$$\mathcal{E}\left[|\psi\rangle\right] = \langle\psi|\hat{H}|\psi\rangle - E\langle\psi|\psi\rangle - \sum_{k=0}^{n-1}\mu_k\langle\psi|\phi_k\rangle \tag{7.48}$$

where $\{|\phi_k\rangle\}$ are the $n-1$ lower-lying eigenstates of $\hat{H}$ and the $\{\mu_k\}$ are Lagrange multipliers enforcing the orthogonality constraints $\langle\psi|\phi_k\rangle = 0$. Again fixing a block of s tensors, the minimization of this functional with respect to this block is given by the *projected* effective Hamiltonian eigenvalue problem

$$\hat{P}^{[j]\,\dagger}\hat{H}_{\text{eff}}^{[j]}\hat{P}^{[j]}A^{[j]}\dots A^{[j+s-1]} = EA^{[j]}\dots A^{[j+s-1]}\,, \tag{7.49}$$

where $\hat{P}^{[j]}$ is a projector into the space orthogonal to the $\{|\phi_k\rangle\}$. Given the states $\{|\phi_k\rangle\}$ as MPSs, we construct these projectors as follows. The diagram corresponding to the overlap $\langle\psi|\phi_k\rangle$ is shown in Figure 7.5(a), with the bold lines corresponding to $|\phi_k\rangle$ and the thin lines to $\langle\psi|$. This is a *linear* form in all of the MPS tensors of $\langle\psi|$, and so the condition that $|\psi\rangle$ be orthogonal to this state for the given block of tensors with all others held fixed may thus be stated as

$$\begin{aligned} \langle\psi|\phi_k\rangle = 0 \quad \Rightarrow \big(\tfrac{\partial}{\partial A^{[j]\star}\ldots A^{[j+s-1]\star}}\langle\psi|\phi_k\rangle\big)\, A^{[j]}\ldots A^{[j+s-1]} &= 0 \qquad (7.50)\\ \equiv F^{[j]\{k\}}A^{[j]}\ldots A^{[j+s-1]} &= 0\,. \qquad (7.51) \end{aligned}$$

Here the linear form enforcing orthogonality $F^{[j]\{k\}}$ is shown diagrammatically in Figure 7.5(b). We can construct the projector $\hat{P}^{[j]}$ as

$$\hat{P}^{[j]} \;= \hat{1} - \textstyle\sum_{kk'} F^{[j]\{k\}}\left(\mathcal{N}^{-1}\right)_{kk'} F^{[j]\{k'\}\,\dagger}\,, \qquad (7.52)$$

where $(\mathcal{N}^{-1})_{kk'}$ is the kk'^{th} element of the inverse of the Gram matrix

$$\mathcal{N}_{kk'} = \mathrm{Tr}\left(F^{[j]\{k\}\,\dagger}F^{[j]\{k'\}}\right)\,. \qquad (7.53)$$

This Gram matrix inverse is important to ensure idempotency of the projector. As before, direct construction of the projected effective Hamiltonian leads to an algorithm which scales poorly as $\mathcal{O}(\chi^6)$. Hence, it is important to use sparse methods which require only the application of $\hat{P}$ and $\hat{H}$ on some block of tensors $A^{[j]}\ldots A^{[j+s-1]}$. Direct application of $\hat{P}^{[j]}$ as written requires $\mathcal{O}(\chi^4)$ operations and also scales quadratically in the number of eigenstates desired N_E due to the double sum in Eq. (7.52). To find a total of N_E eigenstates by this method thus requires $\mathcal{O}(N_E^3\chi^4)$ operations, which is unacceptably slow. A simple idea to reduce this scaling would be to find the eigenvectors of the inverse Gram matrix and re-express the projectors $F^{[j]\{k\}}$ in terms of them, rendering the double sum a single sum. However, while the Gram matrix $\mathcal{N}$ is Hermitian and positive semidefinite it may also be badly conditioned and singular. A numerically stable alternative to this idea is to construct the Moore-Penrose pseudoinverse [39] of the Gram matrix

$$\left(\mathcal{N}^{+}\right)_{kk'} \;= \textstyle\sum_{\mu=1}^{n_p} V_{k\mu}\frac{1}{\lambda_\mu}V^{\star}_{k'\mu} \qquad (7.54)$$

where V is the matrix with the eigenvectors of $\mathcal{N}$ as columns and λ are the n_p eigenvalues of $\mathcal{N}$ which are greater than $n\sqrt{\lambda_{\max}}\epsilon$, where n is the linear dimension of $\mathcal{N}$, $\lambda_{\max}$ its largest eigenvalue, and ϵ the machine precision. We use this pseudoinverse to transform to a new set of linear forms

$$G^{[j]\{\mu\}} = \frac{1}{\sqrt{\lambda_\mu}} \sum_k V_{k\mu} F^{[j]\{k\}} \tag{7.55}$$

such that

$$\hat{P}^{[j]} = \hat{1} - \sum_\mu G^{[j]\{\mu\}} G^{[j]\{\mu\}\,\dagger} \,. \tag{7.56}$$

Often, the dimension of the set $\left\{G^{[j]\{\mu\}}\right\}$ is much smaller than N_E. The diagonalization of the Gram matrix requires $\mathcal{O}(N_E^3)$ operations, independent of χ, and its construction and the construction of G in Eq. (7.55) both require $\mathcal{O}(\chi^2)$ operations. The operation of $\hat{P}^{[j]}$ on the variational block of tensors is now

$$\hat{P}^{[j]} A^{[j]} \dots A^{[j+s-1]} = A^{[j]} \dots A^{[j+s-1]} - \sum_\mu \mathrm{Tr}\left(G^{[j]\{\mu\}\,\dagger} A^{[j]} \dots A^{[j+s-1]}\right) G^{[j]\{\mu\}} \tag{7.57}$$

which is linear in N_E and scales only as $\mathcal{O}(\chi^2)$. Thus, the dominant scaling for typical parameters $\chi \gg N_E$ is still the $\mathcal{O}(\chi^3)$ scaling of the effective Hamiltonian multiply, and the algorithm to find N_E excited states scales as $\mathcal{O}(N_E\chi^3)$. A sweeping approach is used as in the ground state search algorithm, and the iteration is stopped when the variance drops below a user-specified tolerance. As before, the variance does not guarantee that the state found is the next lowest-lying eigenstate, but this does not usually cause problems in practice.

As with the LR overlaps used in the variational ground state search, one can also cache the overlaps $\mathcal{L}\mathcal{R}$ used to construct the linear forms F using the recursions

$$\mathcal{L}^{[1]\{k\}}_{\alpha\alpha'} = \delta_{\alpha,1}\delta_{\alpha',1} \,, \tag{7.58}$$

$$\mathcal{L}^{[\ell+1]\{k\}} = \sum_i A^{[\ell]i^\dagger} \mathcal{L}^{[\ell]k} B^{[\ell]\{k\}i} \tag{7.59}$$

and

$$\mathcal{R}^{[L+1]k}_{\beta'\beta} = \delta_{\beta,1}\delta_{\beta',1}\delta_{k,1}\,, \tag{7.60}$$

$$\mathcal{R}^{[\ell]\{k\}} = \sum_i B^{[\ell]\{k\}i}\mathcal{R}^{[\ell+1]\{k\}}A^{[\ell]i^\dagger} \tag{7.61}$$

where $B^{[\ell]\{k\}}$ is the MPS tensor of $|\phi_k\rangle$ at site ℓ. The linear forms are constructed using these overlaps as

$$F^{[j]\{k\}i_j\dots i_{j+s-1}}_{\alpha\beta} = \sum_{\gamma_1\dots\gamma_{s+1}}\mathcal{L}^{[j]\{k\}i_j}_{\alpha\gamma_1}B^{[j]\{k\}i_j}_{\gamma_1\gamma_2}\dots B^{[i+s-1]\{k\}i_{j+s-1}}_{\gamma_s\gamma_{s+1}}\mathcal{R}^{[j+s]\{k\}}_{\gamma_{s+1}\beta}\,, \tag{7.62}$$

see Figure 7.5(b).

The variational ground state algorithm presented above is essentially equivalent to standard DMRG, aside from the calculation of the variance [12]. When finding excited states, however, DMRG-based approaches target multiple excited states in a single MPS, which causes the bond dimensions to grow and the quality of each individual eigenstate to degrade. Furthermore, as the ground state and all excited states are solved together in that approach, the sparse eigensolver must be able to converge interior eigenvalues, which is known to be troublesome [39, 40]. We call the present algorithm, which is a sparse and numerically stable variant of that proposed in Ref. [50] for PBC, *eMPS* to accentuate the difference.

In our experience, there are two main limitations of eMPS. The first is that it is difficult to construct good variational guesses for the excited states in contrast to the ground state where the infinite size MPS algorithm [26, 51] is applicable. Here, the usefulness of the variance becomes readily apparent, as the discarded weight can be 10^{-12} or less while the variance is of order 10^{-2} in early sweeps. The second is that that the area law considerations which make MPS algorithms so practical for ground states do not in general apply to bulk eigenstates, and so the bond dimension required to accurately represent a general eigenstate may be exponential in the system size, rendering eMPS inapplicable.

The ability to find excited states is useful in many contexts. It provides access to the dynamical gap for determining the location of second order quantum phase transitions [52] and Kibble-Zurek scalings, even when the gap is not between different symmetry sectors. It can help in understanding the structure of conformal field theories by providing access to the primary scaling fields [53]. Excited states yield the structure function and other dynamic response functions of low-lying excitations. Such response functions are of great use for comparing to experimental measurements. Finally, by considering more complex functionals such as $\langle\psi|(\hat{H}-\epsilon)^2|\psi\rangle-\lambda\langle\psi|\psi\rangle$ for minimization, one can determine level spacing statistics in a desired energy range for systems much larger than are amenable to exact diagonalization. Such studies are immensely useful in discussions of integrability and quantum chaos, as well as investigations of the thermalization hypothesis [54–56].

7.5 Calculation of Observables

We now turn to how we can extract information from a state expressed as an MPS. We do so by the expectation values of Hermitian operators, or *observables*. We will demonstrate how to compute observables of three different types: local observables, two-point correlation functions, and general MPOs.

We define a local observable as an operator which acts only on the Hilbert space of a single site: $\hat{O}^{[k]}=\sum_{i,i'}O^{[k]}_{ii'}|i\rangle\langle i'|$. If this site corresponds to the orthogonality center of $|\psi\rangle$ then the expectation value reduces to

$$\langle\psi|\hat{O}^{[k]}|\psi\rangle \;=\sum_{i_k i'_k}\mathrm{Tr}\left({A^{[k]i_k}}^{\dagger}O^{[k]}_{i_k i'_k}A^{[k]i'_k}\right)\,. \tag{7.63}$$

The overall scaling for fixed site index k is $\mathcal{O}\left(\chi^2 d^2\right)$.

A two-point correlation function is an expectation value of the form $\langle\hat{O}^{[q]\dagger}\hat{O}^{[r]}\rangle$ where we take $q<r$ without loss of generality. If the orthogonality center of the MPS, k, lies within the range $q\leq k\leq r$, then we can evaluate the expectation value

using only the tensors in this range. The most efficient way to proceed is first to form the matrix

$$R^{[r]} \quad = \sum_{i_r i'_r} A^{[r] i'_r} O^{[r]}_{i_r i'_r} A^{[r] i_r \dagger} \, , \tag{7.64}$$

recursively generate

$$R^{[r-\ell]} \quad = \sum_i A^{[r-\ell] i} R^{[r-\ell+1]} A^{[r-\ell] i \dagger} \tag{7.65}$$

for $\ell = 1, \dots, r-q+1$, and then evaluate

$$\langle \hat{O}^{[q]\dagger} \hat{O}^{[r]} \rangle \quad = \sum_{i_q, i'_q} \mathrm{Tr} \left(O^{[q]}_{i_q i'_q} A^{[q] i'_q} R^{[q+1]} A^{[q] i_q \dagger} \right) \, . \tag{7.66}$$

For fixed q and r, the algorithm scales as $\mathcal{O}\left(\chi^3 d + \chi^2 d^2\right)$.

To compute the expectation of a general many-body observable $\hat{O}$ expressed as an MPO we start from the right (left) boundary and follow the recursion Eq. (7.44) (Eq. (7.46)) all the way to the opposite boundary, at which point the remaining $1 \times 1 \times 1$ tensor is the expectation value. The overall scaling is $\mathcal{O}\left(\chi^3 d \chi_O + \chi^2 d^2 M\left(\chi_O\right)\right)$ where χ_O is the bond dimension of the MPO and $M\left(\chi_O\right)$ is the number of nonzero elements in the MPO, which usually scales as χ_O.

Entanglement measures such as the bond entropy

$$S_j \equiv -\mathrm{Tr} \hat{\rho}_j \log \hat{\rho}_j \, , \tag{7.67}$$

$$\hat{\rho}_j \equiv \mathrm{Tr}_{i<j+1} |\psi\rangle\langle\psi| \, , \tag{7.68}$$

can be calculated from the singular values $\mathbf{\Sigma}$ of the MPS tensor $A^{[j]i}_{\alpha\beta}$ as

$$S_j \quad = -\sum_\gamma \Sigma^2_\gamma \log \Sigma^2_\gamma \tag{7.69}$$

when this tensor is the orthogonality center. These singular values are computed automatically as part of the algorithm to shift the orthogonality center, see Sec. 7.2.1.

7.6 Time Evolution with MPSs

We now turn our attention to a variational solution of the time-dependent Schrödinger equation

$$i\hbar\frac{\partial}{\partial t}|\psi\left(t\right)\rangle \quad = \hat{H}\left(t\right)|\psi\left(t\right)\rangle \tag{7.70}$$

using MPSs. The general strategy is to find some representation of the propagator over some time interval $[t,t+\delta t]$, $\hat{U}(t,t{+}\delta t)$, and variationally optimize the functional

$$\left||\psi\left(t+\delta t\right)\rangle-\hat{U}\left(t,t+\delta t\right)|\psi\left(t\right)\rangle\right|^2 \tag{7.71}$$

with respect to the MPS tensors of $|\psi(t{+}\delta t)\rangle$. Several complications arise in this case which were not present in the earlier algorithms. The first practical consideration is that the MPO form of the propagator may be difficult and very expensive to calculate. The second difficulty is more physical; the time-dependent state following a global quench of a Hamiltonian parameter has entanglement which generally grows linearly in time [57]. This causes the bond dimension χ to grow exponentially in time, and so there is some finite time where an MPS simulation will exhaust the available computational resources. However, many important questions regarding non-equilibrium dynamics can still be answered by considering moderately sized systems and short times. In addition, consideration of a situation in which the Hamiltonian changes only locally can greatly increase the accessible system sizes and simulation times [58–60].

The most common approach to time evolution for MPSs is to use the Suzuki-Trotter expansion

$$\begin{aligned}\exp\left[-i\delta t\left(\sum_{n=1}^{L-1}\hat{H}_n\right)\right] = &\quad \prod_{n=1}^{L-1}\exp\left(-\frac{i\delta t}{2}\hat{H}_n\right)\\ &\times\prod_{n=L-1}^{1}\exp\left(-\frac{i\delta t}{2}\hat{H}_n\right)+\mathcal{O}\left(\delta t^3\right)\end{aligned} \tag{7.72}$$

or its higher order variants to construct a series of two-site propagators which can be constructed and applied easily. This is the basis of the equivalent [12] TEBD [61] and tDMRG [62, 63] algorithms. Here $\hat{H}_n$ is the nearest-neighbor bond Hamiltonian acting on sites n and $n+1$. This approach is no longer viable when the Hamiltonian has longer-ranged terms, and attempts to accommodate such longer-ranged terms often exhibit poor scaling [64, 65] and require Hamiltonian-specialized implementation, resulting in inefficient, sometimes prohibitively inefficient code. Krylov-based time evolution, which will form the basis for our approach, has been considered in both DMRG [66] and MPS [13] variants for the time-independent case. We note that the latter approach has been used [13, 14] to study time dependent systems, but this necessitated very small time steps set by the rate of change of the Hamiltonian in order to provide accurate results. Our approach generalizes the latter method to the time-dependent case where the error is independent of the rate of change of the Hamiltonian and demonstrates how the algorithm can be formulated entirely in terms of FSA rules for MPOs.

7.7 Commutator-Free Magnus Expansions

The propagator of a general time-dependent Hamiltonian which does not commute with itself at different times is given as a time-ordered exponential

$$\hat{U}\left(t,t+\delta t\right)\equiv\mathcal{T}\left[\exp\left(\int_t^{t+\delta t}dt'\hat{H}\left(t'\right)\right)\right] \tag{7.73}$$

whose most well-known form is the Dyson series

$$\hat{U}\left(t,t+\delta t\right)=\hat{1}+\int_t^{t+\delta t}dt_1\hat{H}\left(t_1\right)+\int_t^{t+\delta t}dt_1\int_t^{t+\delta t}dt_2\hat{H}\left(t_1\right)\hat{H}\left(t_2\right)+\dots \tag{7.74}$$

This formulation of the propagator is not convenient numerically, as the Dyson series is an asymptotic series and so it can be difficult to determine an appropriate criteria for termination of the series. Furthermore, keeping only a finite number of terms in the Dyson series does not preserve the Lie group structure of the propagator; that is, the finite approximation is not unitary. An alternative approach which produces unitary approximations to the propagator was given by Magnus [67] who used the ansatz

$$\hat{U}\left(0,t\right)=\exp\left(-it\hat{\Omega}\left(t\right)\right) \tag{7.75}$$

to define the *Magnus series*

$$\hat{\Omega}\left(t\right)=\sum_{n=1}^{\infty}\hat{\Omega}_n\left(t\right) \tag{7.76}$$

where the n^{th} term is of order t^n in the sense that its power series in t starts with t^n. The term $\hat{\Omega}_n\left(t\right)$ involves n nested integrations over $n-1$ nested commutators of $\hat{H}\left(t\right)$ at different times. Explicitly, the first few terms are:

$$\begin{aligned}
\hat{\Omega}_1\left(t\right) &= \int_0^t dt_1\hat{H}\left(t_1\right)\,, && (7.77)\\
\hat{\Omega}_2\left(t\right) &= \frac{1}{2}\int_0^t dt_1\int_0^{t_1} dt_2\left[\hat{H}\left(t_1\right),\hat{H}\left(t_2\right)\right]\,, && (7.78)\\
\hat{\Omega}_3\left(t\right) &= \frac{1}{6}\int_0^t dt_1\int_0^{t_1} dt_2\int_0^{t_2} dt_3\Big(\left[\hat{H}\left(t_1\right),\left[\hat{H}\left(t_2\right),\hat{H}\left(t_3\right)\right]\right]\\
&\quad+\left[\left[\hat{H}\left(t_1\right),\hat{H}\left(t_2\right)\right],\hat{H}\left(t_3\right)\right]\Big)\,. && (7.79)
\end{aligned}$$

While approximations obtained from truncating the series yield unitary propagators, these expressions are still formidable numerically, involving nested commutators and multidimensional integrals. The commutators pose a special difficulty for MPOs, as exact multiplication of MPOs involves multiplication of the bond dimensions of the MPOs and hence the algorithm scales exponentially in the number of terms kept in the series. Optimization algorithms which attempt to variationally shrink the bond dimension of an MPO sum or product such as those proposed in Ref. [47] may also be used, but these become numerically unstable for large systems, and when MPOs are subtracted as in commutators large cancellations can cause these algorithms to become stuck far from the variational optimum.

Hence, rather than work directly with the Magnus series, Eq. (7.76), we start from ansätze of the form

$$\hat{U}\left(t,t+\delta t\right)=\prod_{i=1}^{s}\exp\left(-i\delta t\hat{\Omega}_i\right) \tag{7.80}$$

where each one of the $\hat{\Omega}_i$ is a linear combination of $\hat{H}$ at different times in the interval $[t,t+\delta t]$, and require that our ansatz matches the Magnus expansion (equivalently, the full propagator) up to order δt^{N+1}. We will call such an ansatz a commutator-free Magnus expansion (CFME) [68, 69]. This ansatz has a number of features which make it desirable for our purposes. It is exactly unitary and so the norm is conserved. Also, provided that we consider the case where only the Hamiltonian parameters change in time and the operators are time-independent, the sums of the Hamiltonian at different times can be represented exactly as an MPO using the rules of Sec. 7.2.2. Thus, the need for complex operations with MPOs vanishes. Finally, because the ansatz takes into account the time dependence of the Hamiltonian explicitly, the time step is not necessarily fixed by the rate of variation of the Hamiltonian, allowing for more coarse stepping in time with fixed error.

Following Ref. [70], the procedure for generating an N^{th}-order CFME is to expand the function $H\left(t\right)$ in terms of (shifted) Legendre polynomials P_n,

$$\hat{H}_n \;=\left(2n-1\right)\delta t\textstyle\int_0^1 dx\hat{H}\left(t+x\delta t\right)P_{n-1}\left(x\right)\,. \tag{7.81}$$

The orthogonality properties of the Legendre polynomials allow the nested integration to be done exactly, leaving a series of nested commutators of the $\hat{H}_n$. Working in a Hall basis [71], this series of commutators is matched with the original Magnus expansion to yield the order conditions $f_{i,n}$ such that

$$\hat{\Omega}_i = \sum_{n=1}^{N} f_{i,n} \hat{H}_n \,. \tag{7.82}$$

We note that these order conditions are independent of $\hat{H}(t)$ by construction, and so are set by the choice of CFME alone. As we only require the result to be valid to order δt^{N+1}, the integration required for the coefficients $\hat{H}_n$ may be performed using Gauss-Legendre quadrature of order $N/2+1$. The end result of the analysis is that an N^{th} order expansion with s exponentials may be written as

$$\hat{U}\left(t+\delta t, t\right) = e^{-i\delta\hat{\Omega}_1} \dots e^{-i\delta\hat{\Omega}_s} \tag{7.83}$$

$$\hat{\Omega}_i = \sum_{m=1}^{N/2+1} g_{i,m} \hat{H}\left(t + x_m \delta t\right) \tag{7.84}$$

$$g_{i,m} = w_m \sum_{n=1}^{N/2+1} \left(2n-1\right) P_{n-1}\left(x_m\right) f_{i,n} \tag{7.85}$$

where x_m and w_m are the points and weights for Gauss-Legendre quadrature [72]. In this work we use a fourth order expansion with three exponentials ($N=4$, $s=3$). The optimal order conditions for this case, obtained in Ref. [70], are

$$f_{1,1} = 11/40 \,, \quad f_{1,2} = 20/87 \,, \quad f_{1,3} = 7/50 \,, \quad f_{2,1} = 9/20 \,, \quad f_{2,3} = -7/25 \,, \tag{7.86}$$

with $f_{s-i+1,n} = \left(-1\right)^{n+1} f_{i,n}$. Order conditions for higher order expansions may also be found in Ref. [70].

We now consider that our time-dependent Hamiltonian MPO is constructed from a set of FSA rules $\left\{\mathcal{R}_p\left(\left\{\hat{O}^{p_1}, \dots, \hat{O}^{p_n}\right\}, \left\{h^p\left(t\right)\right\}, w^p\right)\right\}$ in which the OA and the weights are chosen without loss of generality to be time-independent. In this case, the expansion Eq. (7.81) is applied individually to each Hamiltonian parameter $h^p\left(t\right)$, resulting in the parameters $\{h^p_n\}$. Now, because of the canonical decomposition of Sec. 7.2.2, the MPO forms of $\hat{\Omega}_i$ from Eq. (7.84) at time t can be constructed exactly using the FSA rule set $\left\{\mathcal{R}_p\left(\left\{\hat{O}^{p_1},\dots,\hat{O}^{p_n}\right\},\left\{\sum_{m=1}^{N/2+1} g_{i,m}h^p\left(t+x_m\delta t\right)\right\},w^p\right)\right\}$. We note that each one of these operators has the same bond dimensions as the original Hamiltonian, and the updates of operators $\hat{\Omega}_i$ at each time step can be done in $\mathcal{O}\left(L\chi_O\right)$ time, which is essentially negligible.

The fact that we can construct the $\hat{\Omega}_i$ from the same FSA rules as the Hamiltonian implies that our CFME ansatz is equivalent to evolving our system according to piecewise constant Hamiltonians of the same form but with differing Hamiltonian parameters. Additionally, as also occurs in high-order Suzuki-Trotter expansions, evolution backwards in time may occur. Finally, we note that even terms in the Hamiltonian whose parameters do not vary in time have their magnitude altered by Eq. (7.84), as $\sum_m g_{i,m}\neq 1$ in general.

7.8 Krylov Subspace Propagation

Using the CFME Eq. (7.83) we never need to explicitly form an MPO representation of the propagator provided we can find an MPS representation of the exponential of an MPO applied to an MPS. We find such a representation from minimizing functionals of the form

$$\left||\phi\rangle-\exp\left(-i\delta t\hat{\Omega}\right)|\psi\rangle\right|^2\,, \tag{7.87}$$

where, importantly, $\hat{\Omega}$ has a known MPO representation. We do so by forming a Krylov subspace approximation to the exponential [73] in which the Krylov vectors are represented as MPSs. Specifically, we do so via the Lanczos algorithm for the matrix exponential, which can be stated as

1. *Input*: Input $\hat{\Omega}$ in MPO form, $|\psi\rangle$ in MPS form, and a tolerance ϵ for truncating the recursion.

2. *Initialize*: Set $\beta_0=\langle\psi|\psi\rangle=1$ and $|r\rangle=|\psi\rangle$.

3. *Iterate*: For $j = 1, 2, \ldots$ until convergence

    (a) $|v_j\rangle = |r\rangle/\beta_{j-1}$.

    (b) $|r\rangle = \hat{\Omega}|v_j\rangle$.

    (c) $\alpha_j = \langle v_j|r\rangle$.

    (d) Orthogonalize $|r\rangle$ against $|v_j\rangle$ and $|v_{j-1}\rangle$.

    (e) Re-orthogonalize $|r\rangle$ against all v_k, $k \leq j$ if necessary.

    (f) $\beta_j = \langle r|r\rangle$

    (g) Form the matrix exponential of $T^{(j)}$, $U^{(j)}$, and obtain $c^{(j)} = U^{(j)}_{1:j,1}$.

    (h) Test for convergence.

4. *Finalize:* Set $|\phi\rangle = \sum_{i=1}^{j} c_i^{(j)}|v_i\rangle$

Here $T^{(j)}$ is the symmetric tridiagonal matrix with the α_i, $1 \leq i \leq j$ on the diagonal and β_i, $1 \leq i \leq j-1$ on the superdiagonal. It is important to use a matrix exponentiation method which produces a unitary matrix to machine precision in order to not lose the Lie group structure. Because of the small linear dimensions of the matrix $T^{(j)}$, exponentiation by direct diagonalization is practical. An *a posteriori* estimate for convergence of the Lanczos recursion is that $\left|2\beta_{j-1}c_j^{(j)}\right| < \epsilon$, where ϵ is the tolerance [73]. This can be compared with residual estimates in the ordinary Lanczos algorithm for finding eigenvalues. A rigorous bound on the approximation $||\psi\rangle_{\mathrm{krylov}} - |\psi\rangle| \leq 12\exp\left\{-\frac{(\rho\delta t)^2}{16n}\right\}\left(\frac{e\rho\delta t}{4n}\right)^n$ can be established [74] when $n \geq \rho\delta t/2$ with n the number of Lanczos vectors and $\rho = |E_{\max} - E_{\min}|$ the spectral width of $\hat{\Omega}$. This estimate shows that for typical tolerances $\epsilon = 10^{-6}$ to 10^{-10}, 6 to 20 Lanczos vectors suffice.

As stated before, MPSs do not form a vector space and so the multiplication by $\hat{\Omega}$, the orthogonalization, and the final summation cannot be done exactly while keeping the bond dimension of our MPS fixed. However, just as with the eigenstate search, we can devise variational algorithms for these three operations which are iteratively performed until a desired tolerance is reached and use this tolerance to bound the bond dimension of our time-evolved MPS. We begin by briefly reviewing the standard algorithm [12] for finding the optimal MPS $|\phi\rangle$ representing a sum of MPSs $\sum_{k=1}^{N} c_k|\psi_k\rangle$ to a given tolerance, as the other algorithms are similar but more complex. In this case we have a set of $\mathcal{LR}$ overlaps defined between our variational state $\langle\phi|$ and the states $|\psi_k\rangle$ as in Eqs. (7.59) and (7.61). We now sweep through the lattice and make the replacement

$$A^{[j]}\dots A^{[j+s-1]} = \textstyle\sum_k c_k F^{[j]\{k\}}\,, \tag{7.88}$$

where the $F^{[j]\{k\}}$ are formed as in Eq. (7.62), see also Figure 7.5(b). The orthogonality center of $|\phi\rangle$ is then shifted, the $\mathcal{LR}$ overlaps updated, and sweeping continued. Convergence can be monitored via

$$\begin{aligned}\left||\phi\rangle-\sum_k c_k|\psi_k\rangle\right|^2 &= \mathrm{Tr}\left[\left(A^{[j]}\dots A^{[j+s-1]}\right)^{\dagger}A^{[j]}\dots A^{[j+s-1]}\right]+1 \\ &\quad -2\sum_k \mathrm{Re}\,\mathrm{Tr}\left[c_k\left(A^{[j]}\dots A^{[j+s-1]}\right)^{\dagger}F^{[j]\{k\}}\right]<\epsilon\,,\end{aligned} \tag{7.89}$$

with Re denoting the real part. Because we do not have to solve an eigenequation at each iteration, this algorithm is often much less costly than the iterative eigenstate search. Also, when we have that the coefficient vector c and all of the $\{|\psi_k\rangle\}$ have length 1, we can normalize the state $|\phi\rangle$ at the end of the calculation if required.

The algorithm to variationally fit an MPS to $\hat{\Omega}|\psi\rangle$ is similar. In this case we have a set of LR overlaps defined via the recursions

$$L^{[1]}_{\kappa\alpha\alpha'} = \delta_{\alpha,1}\delta_{\alpha',1}\delta_{\kappa,1}\,, \tag{7.90}$$

$$L^{[\ell+1]}_{\kappa\alpha\alpha'} = \left[\textstyle\sum_{\gamma' i'}\left[\sum_{i\kappa'}\left[\sum_{\gamma}A^{[\ell]i\star}_{\gamma\alpha}L^{[\ell]}_{\kappa'\gamma\gamma'}\right]W^{[\ell]ii'}_{\kappa'\kappa}\right]B^{[\ell]i'}_{\gamma'\alpha'}\right]\,, \tag{7.91}$$

$$R^{[L+1]}_{\kappa\beta'\beta} = \delta_{\beta,1}\delta_{\beta',1}\delta_{\kappa,1}\,, \tag{7.92}$$

$$R^{[\ell]}_{\kappa\beta'\beta} = \left[\sum\nolimits_{\gamma' i'} B^{[\ell]i'}_{\beta'\gamma'} \left[\sum\nolimits_{\kappa' i} W^{[\ell]ii'}_{\kappa\kappa'} \left[\sum\nolimits_{\gamma} R^{[\ell+1]}_{\kappa'\gamma'\gamma} A^{[\ell]i^\star}_{\beta\gamma}\right]\right]\right] , \tag{7.93}$$

where the MPS tensors of $|\phi\rangle$ are denoted by A and those of $|\psi\rangle$ denoted by B. We now sweep through the lattice and make the replacement

$$A^{[j]} \dots A^{[j+s-1]} = \hat{H}^{[j]}_{\mathrm{eff}} B^{[j]} \dots B^{[j+s-1]} \tag{7.94}$$

where the effective Hamiltonian is formed from the LR overlaps as in Eq. (7.41). Convergence can be monitored via

$$\begin{aligned} \left| |\phi\rangle - \hat{H}|\psi\rangle \right|^2 = {} & \mathrm{Tr}\left[\left(A^{[j]} \dots A^{[j+s-1]}\right)^\dagger A^{[j]} \dots A^{[j+s-1]}\right] + \langle\psi|\hat{H}^2|\psi\rangle \\ & -2\mathcal{R}\mathrm{Tr}\left[\left(A^{[j]} \dots A^{[j+s-1]}\right)^\dagger \hat{H}_{\mathrm{eff}} B^{[j]} \dots B^{[j+s-1]}\right] < \epsilon , \end{aligned} \tag{7.95}$$

Here, $\langle\psi|\hat{H}^2|\psi\rangle$ can be computed in a manner similar to the variance, and need only be computed once at the beginning of the calculation. We have also assumed that the block of tensors in Eq. (7.95) contains the orthogonality center. Again, this algorithm is often much less costly than the iterative eigenstate search.

We now turn to steps (iii)(d) and (iii)(e) of our algorithm. Step (iii)(d) is usually stated for ordinary vector spaces as

$$|r\rangle = |r\rangle - \alpha_j|v_j\rangle - \beta_{j-1}|v_{j-1}\rangle , \tag{7.96}$$

as $\alpha_j = \langle v_j|r\rangle$ and $\beta_{j-1} = \langle v_{j-1}|r\rangle$ and so this is equivalent to classical Gram-Schmidt orthogonalization. Hence, we could implement step (iii)(d) by using the fitting algorithm Eq. (7.88) to find the MPS closest to $|r\rangle - \alpha_j|v_j\rangle - \beta_{j-1}|v_{j-1}\rangle$. However, we have found that the following algorithm, which is closely related to eMPS, often converges more quickly and also is applicable to step (iii)(e). In our method we look for the optimal MPS $|\phi\rangle$ representing $|\psi\rangle$ but also subject to the constraints that $\langle\phi|\psi_k\rangle = 0$ for some set $\{|\psi_k\rangle\}$. We start by copying the state $|\psi\rangle$ to a variational ansatz $|\phi\rangle$. We then construct overlaps between the state $\langle\phi|$ and $|\psi\rangle$, which we call LR[57] and a set of overlaps between $\langle\phi|$ and $|\psi_k\rangle$, which we call $\mathcal{LR}$. We then sweep through the lattice and make the replacement

$$A^{[j]}\dots A^{[j+s-1]} \;=\sum\nolimits_{\gamma_1\dots\gamma_{s+1}} L^{[j]i_j}_{\alpha\gamma_1} B^{[j]i_j}_{\gamma_1\gamma_2}\dots B^{[j+s-1]i_j}_{\gamma_s\gamma_{s+1}} R^{[j+s]}_{\gamma_{s+1}\beta}\,, \tag{7.97}$$

with B the MPS tensors of $|\psi\rangle$ and A the MPS tensors of $|\phi\rangle$. We then apply the projector into the space orthogonal to the $|\psi_k\rangle$ by constructing the set $\left\{G^{[j]\{\mu\}}\right\}$, $\mu=1,\dots,p$, as in Eq. (7.55) and performing

$$A^{[j]}\dots A^{[j+s-1]} \;= A^{[j]}\dots A^{[j+s-1]} - \mathrm{Tr}\left(G^{[j]\{\mu\}\dagger}A^{[j]}\dots A^{[j+s-1]}\right)G^{[j]\{\mu\}}\,, \tag{7.98}$$

for $\mu=1,\dots,p$. Using the fitting algorithm of Eq. (7.88) corresponds to replacing the $\mathcal{LR}$ overlaps, which are between the *variational* state $\langle\phi|$ and the set $\{|\phi_k\rangle\}$, with a set of $\mathcal{LR}$ overlaps between the state $\langle\psi|$ and the set $\{|\phi_k\rangle\}$. Our algorithm, which amounts to fitting followed by modified Gram-Schmidt, uses information about the distance between the variational state and those to be orthogonalized against to determine operations, and hence often converges more quickly and is more stable. Convergence can be monitored by ensuring that $\langle\phi|\psi_k\rangle$ are orthogonal to a precision ϵ via

$$\left|\mathrm{Tr}\left(A^{[\ell]\dagger}F^{[\ell]\{k\}}\right)\right|\le\epsilon\,. \tag{7.99}$$

If one requires additional truncation of the bond dimension, one can switch to the ordinary fitting algorithm Eq. (7.88) at this point, using a new variational state $|\zeta\rangle$ to fit to $|\phi\rangle$.

We now pause to consider the sources of error in the time-propagation routine. First, because the CFME expansion Eq. (7.83) is only of order δt^{N+1}, the error in-

---

[57] In this initialization all of the LR are Kronecker deltas provided that $|\psi\rangle$ has an orthogonality center.

curred in using this form of the propagator is $\epsilon_{\mathrm{CFME}} = ct_{\mathrm{final}}\delta t^{N}$, where t_{final} is the final time reached. Thus, as the final time desired becomes longer, smaller time steps should be taken in order to keep the error fixed. The coefficient c can be determined by using this known scaling and decreasing the time step by a constant factor. Factors in the range $2^{1/N}$ to $3^{1/N}$ are practical. This strategy can also be used to devise adaptive time-step applications such as those used widely in ordinary differential equation solvers. Next, there is the error $\epsilon_{\mathrm{Krylov}}$ incurred in the Krylov subspace approximation to the exponential. As discussed above, this error can be minimized by adding more and more Lanczos vectors. This error also involves the time step δt, and so when determining the CFME expansion error constant c one should be careful that $\epsilon_{\mathrm{Krylov}} < \epsilon_{\mathrm{CFME}}$. Finally, there are errors resulting from the variational fitting of MPSs in steps (iii)(b), (iii)(d), and (iii)(e) of the Lanczos algorithm for the matrix exponential. These can be reduced by lowering the variational tolerances, but this is done typically at the expense of a larger bond dimension χ and hence a slower algorithm and more memory usage.

7.9 Simulation Protocol

We are now in a position to devise a complete, generic protocol for the time evolution of a 1D quantum system.

1. *Input*: Input an operator alphabet and a set of FSA rules defining the Hamiltonian MPO. Input the functional forms $\{h_p(t)\}$ of its Hamiltonian parameters, a final time desired t_{final} and a time-step δt. Input tolerances $\{\epsilon\}$ for variational ground state search and time evolution.

2. *Initial state preparation*: Find the ground state of the Hamiltonian using variational ground state search from Sec. 7.3. Alternatively, if a different initial state is desired, read in its MPS representation. Set $|\psi(t=0)\rangle$ to be this state.

3. *Measure*: Measure local observables and two-point correlation functions constructed from the OA as well as MPO observables constructed from their own FSA rule sets.

4. *eMPS*: Use eMPS (Sec. 7.4) to find excited states of the Hamiltonian, if desired. Measure properties of the excited states.

5. *tMPS*: Set $t = 0$.

   (a) For $i = s, \dots 1$, construct $\hat{\Omega}_i$ from Eq. (7.84) using the FSA rules and use the Krylov algorithm to apply the matrix exponentials of these operators in succession to $|\psi(t)\rangle$. Set $t = t + \delta t$.

   (b) Measure $|\psi(t)\rangle$ if desired.

   (c) Continue (a) and (b) until t_{final} is reached.

Starting from a base set of rules such as the site, bond, exponential, decaying function, and finite function rules of Sec. 7.2.2, a single implementation can accommodate a vast range of systems based on the particular OA, rules, and quench protocols $\{h_p(t)\}$ used.

7.10 Case Studies

7.10.1 Case Study: Ising model in a Transverse Field

For our first case study we choose the paradigmatic Ising model in a transverse field, with Hamiltonian

$$\hat{H} = -J \sum_{\langle i,j \rangle} \hat{\sigma}_i^z \hat{\sigma}_{i+1}^z - h \sum_i \hat{\sigma}_i^x \,. \tag{7.100}$$

Here J is the coupling energy, h is a transverse magnetic field, and the $\{\hat{\sigma}_i\}$ are the Pauli spin operators on site i. We choose this model because its dynamics are amenable to numerically exact study using the time-dependent Bogoliubov-de Gennes

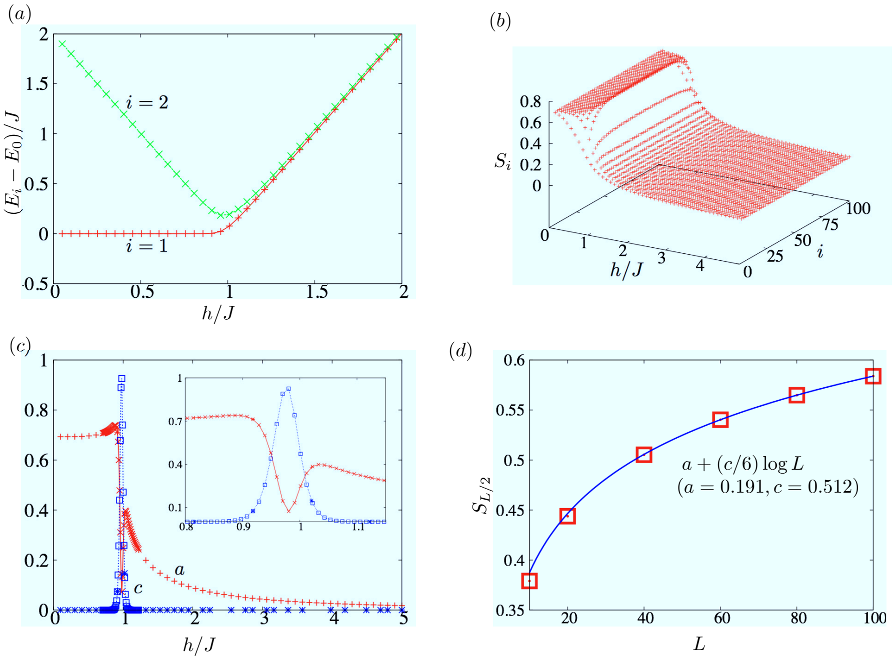


Figure 7.6: (color online) *Statics of the Ising model.* (a) The gaps to the two-lowest-lying eigenstates, computed using eMPS. (b) The bond entropy S_i vs. the site index i and magnetic field h for $L = 100$ sites. Note the increased curvature near the critical point $h = 1$. (c) The central charge c (blue line) and bulk entanglement a (red line) extracted from a fit to Eq. (7.101) vs. h for $L = 100$ sites, neglecting 30 sites at both boundaries. The inset is a close-up of the critical region. (d) The bond entropy of the central site $S_{L/2}$ vs. L at $h = 1.0$ together with a fit extracting the central charge.

formalism (see 7.12 for a review) and so we were able to carefully check convergence of our results. The statics have all been converged to eight digits, and the dynamics at all times to at least four digits. Here we refer to convergence of local quantities such as the energy or density of defects. Nonlocal quantities such as the bond entropy will not have this same accuracy, but numerical tests show that the qualitative behavior is unaffected. We begin with a discussion of the statics.

In Figure 7.6(a) we demonstrate the gaps from the ground state to the two lowest eigenstates, computed via eMPS. The variances are smaller than the point size in this case. The upper (lower) curve corresponds to even (odd) parity, while the ground state has even parity, where parity is defined as simultaneous inversion of all spins $P = \langle \prod_i \hat{\sigma}_i^x \rangle$. Hence, the relevant gap for discussing the quantum phase transition is in fact the gap to the second excited state, shown in green.[58] Both gaps close at criticality, and this can cause the first excited state returned by eMPS to be a mixture of these two states. This will not affect the energies so long as the variance remains small, but it can affect other observable properties of the states. There are two ways to remove this nuisance. The first, most complex, and most preferable is to use an MPS representation in which the state is explicitly $\mathbb{Z}_2$ invariant [40]. The second is to add a field coupling to the parity $-h_p \prod_i \hat{\sigma}_i^x$ to cause the relevant even-parity subspace to become lower in energy than the odd-parity subspace. This operator is an MPO with bond dimension 1, the MPO equivalent of a product state. The closing of the gap at the known critical point $h = 1$ is linear in $1/L$, indicating a conformally invariant critical theory with dynamical critical exponent $z = 1$.

We venture to determine the central charge of the critical theory by fitting to the Calabrese-Cardy formula in two ways. In the first, we fit to the finite-size formula

$$S_i \quad = \tfrac{c}{6} \log \left[\tfrac{2L}{\pi} \sin \left(\tfrac{\pi i}{L} \right) \right] + a \tag{7.101}$$

for fixed L and variable i, and in the second we fix i at $L/2$ and fit $S_{L/2}$ to this formula for various L. Near criticality, the presence of nonzero c indicates a curvature of S_i, while in the gapped phases S_i obeys an area law and is hence a constant apart from finite-size effects. The bond entropy in the bulk approaches the correct limiting values of $\log 2$ as $h \to 0$ and 0 as $h \to \infty$. The first fit, shown in Figure 7.6(a), provides us with a clear indicator of the critical region by the spike in the central charge c. However, the precise determination of c for a finite size system in this case is noisy, likely due to strong finite-size effects. Once we have narrowed down where the critical region is, the second fit, shown in Figure 7.6(b), allows us to extract the anticipated value $c = 1/2$ much more precisely. If the same scaling analysis is attempted at a point which is not the critical point, the bond entropy saturates and $c \to 0$ as $L \to \infty$.

---

[58] The even parity gap for $h > 1$ is in fact twice the demonstrated gap in green, but the essential piece is the closing of the gap at criticality.

We can understand this as a large but finite correlation length ξ. For $L/2 < \xi$, the system appears to be conformal and we see scaling of the bond entropy with L. For $L > \xi$ the bond entropy saturates and this scaling breaks down, indicating that the given region is not critical. We note that in this analysis we have used no properties which are specific to this system e.g. correlation functions of an order parameter to extract the critical behavior.

We now turn to the dynamics. The Ising model has also been a subject of interest for dynamics as a testbed for the Kibble-Zurek hypothesis (KZH) that equilibrium properties determine nonequilibrium properties following a quench across a quantum critical point. This was studied in Refs. [75–77] using the quench protocol

$$h\left(t\right)/J \quad = 5\left(1-\tfrac{t}{\tau}\right)\,,\quad 0\le t\le \tau\,. \tag{7.102}$$

A useful quantity for determining how non-adiabatic a particular quench is in this case is the density of defects

$$\rho=\frac{1}{2L}\sum_{i}\left(1-\hat{\sigma}_i^z\hat{\sigma}_{i+1}^z\right) \tag{7.103}$$

which is the density of Bogoliubov quasiparticles at zero magnetic field. In addition to the density of defects, universal scaling has also been predicted in the heat, or non-adiabatic part of the energy,

$$Q\left(t\right)=\langle\psi\left(t\right)|\hat{H}\left(t\right)|\psi\left(t\right)\rangle-\langle\psi_{\mathrm{g.s.}}\left(t\right)|\hat{H}\left(t\right)|\psi_{\mathrm{g.s.}}\left(t\right)\rangle \tag{7.104}$$

where $|\psi_{\mathrm{g.s.}}(t)\rangle$ represents the instantaneous ground state of $\hat{H}(t)$. In addition to these quantities, which are amenable to Bogoliubov-de Gennes analysis, we also compute the time-dependent bond entropy. Our results are shown in Figure 7.7.

We first discuss the heat, as shown in panels (a) and (d). Panel (a) displays the heat as a function of time, and demonstrates a sharp change in the behavior of the system as we pass through the critical point. This is especially true of the longest quenches. In panel (d) we investigate the heat as a function of the quench rate both at the time when h takes on its critical value, $t_c = 4\tau/5$, and at the final time when $h = 0$. The large difference indicates that non-adiabatic processes continue after we have passed from the critical region back into the gapped ferromagnetic region. Thus, the universal scaling of the heat may be difficult to determine if the critical point itself is not known sufficiently well. We now move on to the density of defects as shown in panels (b) and (e). In panel (b) we see that the density of defects at the final time decreases slowly to zero as $\tau \to \infty$; that is, when the quench becomes perfectly adiabatic. This is in accordance with the KZH prediction. Panel (e) demonstrates the large disparity between the density of defects at the critical time and the final time for all but the most rapid of quenches. Finally, in panels (c) and (f) we investigate the bond entropy. In panel (c) we see the bond entropy of the central bond as a function of time. As the quench becomes more adiabatic, the entropy increases more towards the ferromagnetic limiting value of $\log 2$. However, for very slow quenches, the bond entropy reaches this value before the end of the quench and then begins to oscillate. In panel (f) we show the bond entropy as a function of the bond index at the critical time. Bulk curvature such as that seen at criticality in Figure 7.6(d) is absent, indicating that the time-evolved state is still quite far from the conformal ground state.

7.10.2 Case Study: Dipolar Ising chain

In this section we go beyond exactly solvable models and study a dipolar Ising chain

$$\hat{H}_{\mathrm{dip}} = -J \sum_{i<j;|j-i|\leq 6} \frac{\hat{\sigma}_i^z \hat{\sigma}_j^z}{(j-i)^3} - h \sum_i \hat{\sigma}_i^x \,. \tag{7.105}$$

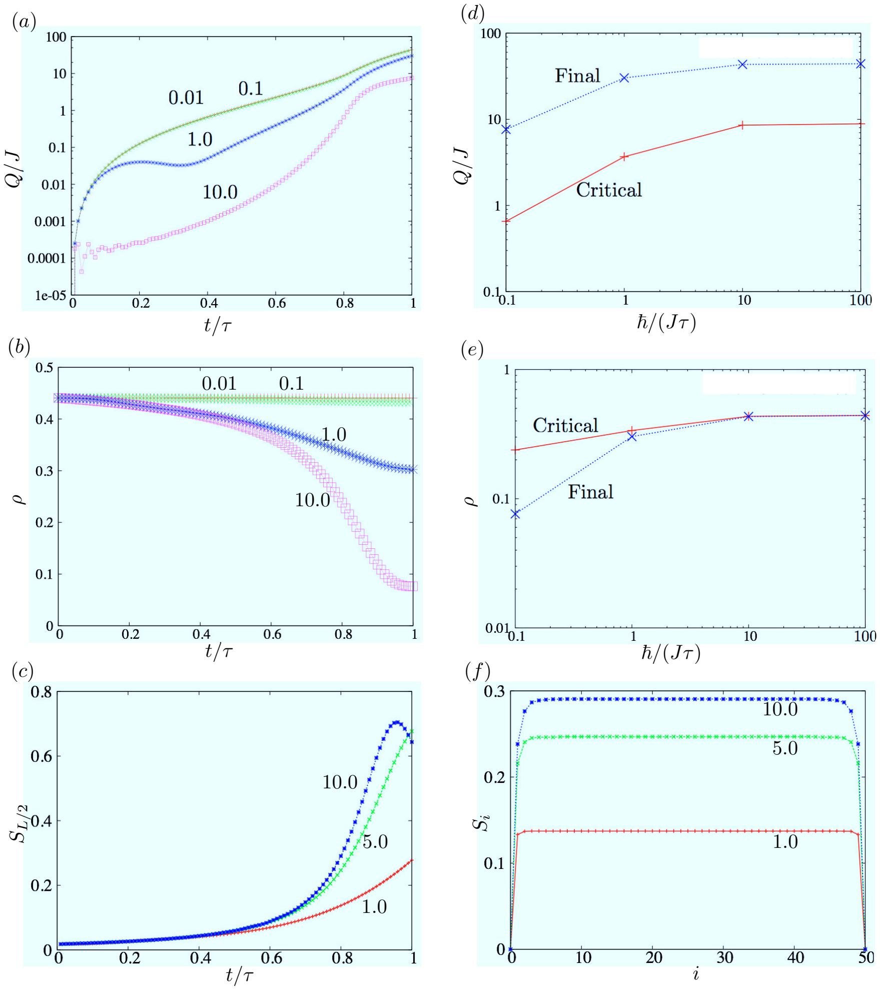


Figure 7.7: (color online) *Dynamics of the Ising model.* (a) The heat as a function of time shows a marked change in behavior as we transition past the critical time $t/\tau = 4/5$. The numbers indicate the value of $\tau J/\hbar$. $\tau J/\hbar = 0.1$ and 0.01 are indistinguishable on the scale of this plot. Both may be considered to be instantaneous. (b) The density of defects as a function of time scales to zero as $\tau J/\hbar \to \infty$, in accordance with the KZH. (c) The bond entropy of the central bond as a function of time approaches the limiting value $\log 2$ as the quench becomes more adiabatic. For nearly adiabatic quenches, the bond entropy oscillates after the critical point. (d) Scaling of the heat in the final and critical stages with the inverse quench time shows marked non-adiabatic processes occurring after passing the critical point. (e) Scaling of the density of defects in the final and critical stages with the inverse quench time shows non-adiabatic processes after passing the critical point only for slow quenches. (f) Snapshots of the bond entropy at the critical time demonstrate that the system is not generally close to its conformal ground state.

Such models are relevant to the study of ultracold molecules in optical lattices, where the dipole-dipole interaction falls away as $1/r^3$ with r the distance between dipoles [7–9, 48, 78]. Here the cutoff $|j-i| \leq 6$ represents a consistent order of approximation in going from a Hubbard-type model with dipolar interactions to a spin model. We stress that all results obtained in this section were obtained using the same implementation as the last section.

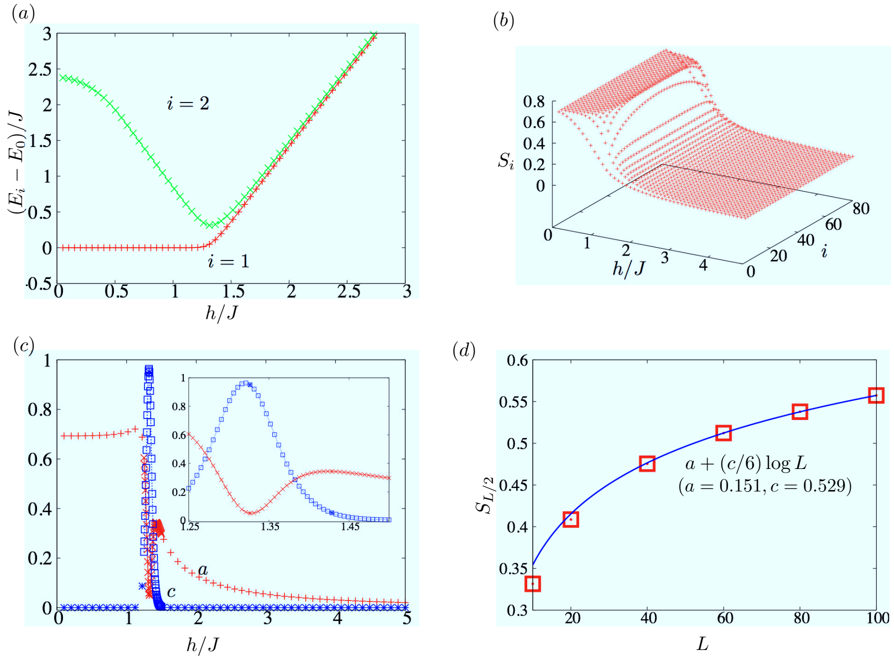


Figure 7.8: (color online) *Statics of the dipolar Ising model.* (a) The gaps to the two-lowest-lying eigenstates, computed using eMPS. Here we see a breakdown of the linear dispersion at small h, indicating interactions between quasiparticles. (b) The bond entropy S_i vs. the site index i and magnetic field h for $L = 80$ sites. The point of greatest curvature is shifted towards larger h with respect to the Ising model. (c) The central charge c (blue line) and bulk entanglement a (red line) extracted from a fit to Eq. (7.101) vs. h for $L = 80$ sites, neglecting 30 sites at both boundaries. The inset is a close-up of the critical region. (d) The bond entropy of the central site $S_{L/2}$ vs. L at $h = 1.362$ together with a fit extracting the central charge.

We first turn to the statics of this model, shown in Figure 7.8. Many of the features are similar to those of the nearest-neighbor Ising model. The most important differences are that the critical region is shifted towards larger h with respect to the nearest-neighbor Ising model as seen in panels b) and c). This indicates increased stability of the ferromagnetic phase, in accordance with expectations. Using these points as a guide, we determine the critical point to be $h_c = 1.362 \pm 0.01$, as shown by the scaling in panel d). Additionally, the energy of the first even parity excited state deviates from pure $z = 1$ linear behavior near $h = 0$, indicating interactions between quasiparticles which were noninteracting in the nearest-neighbor Ising model.

We now turn to dynamics, following the same quenching protocol Eq. (7.102) as above. The results are shown in Figure 7.9. We reiterate that the dynamics of this model cannot be handled by Bogoliubov-de Gennes methods, nor straightforwardly with standard tDMRG/TEBD. The density of defects no longer represents the density of quasiparticles at the critical point, but we compute it for comparison with the results of the nearest-neighbor Ising model. Because of the larger MPO bond dimensions and the more rapid growth of bond dimension for this model, we restrict our analysis to short times $J\tau/\hbar \le 5$, though an optimized implementation could reach longer times. The basic features are similar to the dynamics of the nearest-neighbor Ising model. One quantitative difference is that, because the critical point is reached at an earlier time than in the nearest-neighbor Ising model, oscillations in the central bond entropy occur for shorter quench times.

7.11 Conclusions

The power of matrix product state algorithms over DMRG-based algorithms is most readily apparent when multiple states are involved, as each state may be represented as a separate matrix product state in the former approach. Because matrix product states with a fixed bond dimension do not form a vector space, a set of matrix product states carries more information at smaller numerical cost than the same set represented as a multi-state targeted basis in DMRG. We have presented two algorithms, eMPS and tMPS, which use this property to find eigenstates and perform time evolution of strongly correlated 1D quantum systems.

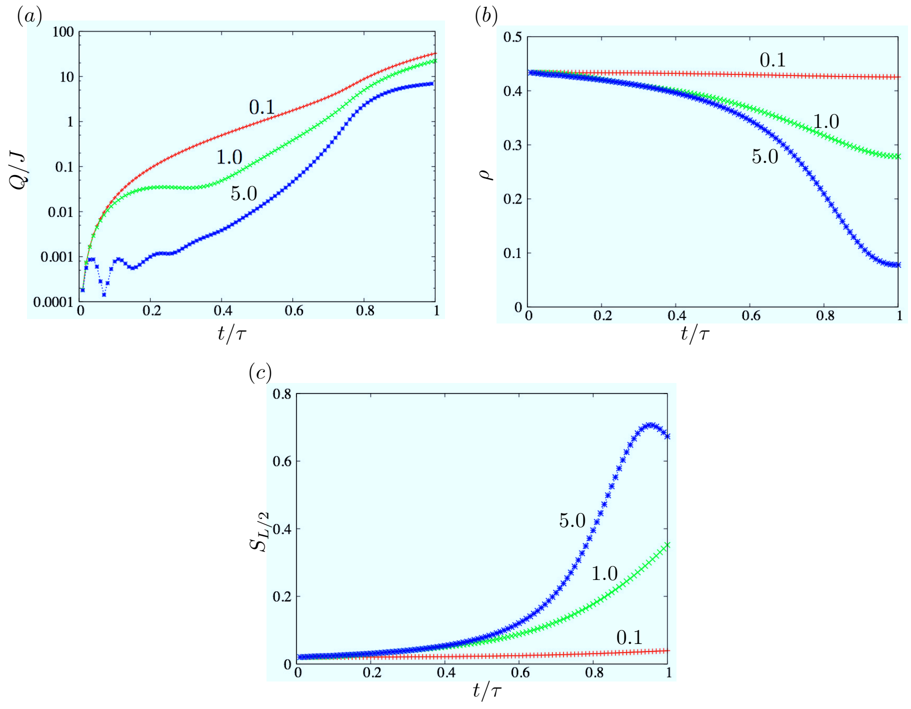


Figure 7.9: (color online) *Dynamics of the dipolar Ising model.* (a) The heat as a function of time displays a slower buildup in post-critical non-adiabatic effects for longer quenches, as in the nearest-neighbor case. (b) The density of domain walls as a function of time is comparable to that for the nearest-neighbor case, but no longer has the same interpretation in terms of quasiparticles. (c) The bond entropy of the central bond oscillates for the shorter quench time $J\tau/\hbar = 5$ due to the quench passing the critical point sooner than in the nearest-neighbor case.

eMPS uses a set of eigenstates stored as separate matrix product states to define a projector into the space orthogonal to this set. We use this projector to explicitly orthogonalize a variational state against previously determined eigenvectors in order to find excited states. The explicit matrix product state representation allows us to store the excited states much more accurately than with standard DMRG, and allows allows us to ensure global orthogonality between the eigenstates to a desired tolerance. The variance, which is computed exactly using the matrix product operator representation of the Hamiltonian, gives strict error bars on the energies obtained with this procedure.

tMPS avoids the need for an explicit representation of the propagator by using a commutator-free Magnus expansion and then building successive Krylov subspace approximations to the matrix exponentials which appear in the expansion. Each vector in the Krylov subspace is stored as a separate matrix product state to maximize efficiency. Furthermore, the operators $\hat{\Omega}_i$ have exact representations as matrix product operators with the same bond dimension as the Hamiltonian. Our algorithm eliminates the need for Hamiltonian-specialized implementation of dynamics. Additionally, by carefully accounting for the time dependence of the Hamiltonian with a commutator-free Magnus expansion, the error in our algorithm depends only on commutators of the Hamiltonian with itself at different times and not on its derivatives. As with eMPS, the errors are rigorously accounted for by considering distance functionals with the variational state.

The matrix product operator forms of 1D Hamiltonians can be obtained using a small set of finite state automaton rules such as the site, bond, and finite function rules. Using matrix product operator arithmetic, we can add together the various terms in a Hamiltonian from these rules to form a complete canonical MPO representation. This representation allows for templating of Hamiltonians which depend only on the type of interactions and not on the microscopic constituents of the lattice model. Furthermore, given the time-dependent form of the Hamiltonian parameters, one can use the same template to form the operators $\hat{\Omega}_i$ which appear in tMPS at negligible numerical cost.

We used our algorithms to study both the nearest-neighbor Ising model in a transverse field and a long-range Ising model in a transverse field. By the closing of the gap obtained with eMPS we determined that the critical points of these models were conformal, and so we used the Calabrese-Cardy formula for the bond entropy of conformal systems to locate the critical point and its associated central charge. The known result $h_c = 1$ was verified for the nearest-neighbor case, and the critical point was shifted deeper into the paramagnetic region $h_c = 1.362 \pm 0.01$ for the long-range case, indicating a stabilization of the ferromagnetic phase. We used tMPS to study the dynamics of these models following a linear quench of the transverse field from the paramagnetic phase through the critical point into the ferromagnetic phase. We saw strong non-adiabatic effects in the heat as quenching continued into the ferromagnetic region, scaling of the density of defects consistent with the Kibble-Zurek hypothesis, and the oscillation of the bond entropy near its limiting ferromagnetic phase value for nearly adiabatic quenches.

This work was supported by the Heidelberg Center for Quantum Dynamics, the Alexander von Humboldt Foundation and the National Science Foundation under Grant PHY-0903457. We also acknowledge the Golden Energy Computing Organization at the Colorado School of Mines for the use of resources acquired with financial assistance from the National Science Foundation and the National Renewable Energy Laboratories. We thank Erman Bekaroglu for discussions and a thorough reading of the manuscript.

7.12 Time Evolution of Exact Solution of 1D Transverse-Field Quantum Ising Model for Comparison with tMPS

The solution of the statics of the transverse-field quantum Ising model is covered in standard texts [52]. However, for comparison with tMPS we require a description of the dynamics, and so we present the dynamical case here. To find the exact solution of the Ising model,

$$\hat{H} = -J \sum_{\langle i,j \rangle} \hat{\sigma}_i^z \hat{\sigma}_j^z - h \sum_{i=1}^{L} \hat{\sigma}_i^x \,, \tag{7.106}$$

we affect the Jordan-Wigner transformation

$$\hat{\sigma}_i^x = 1 - 2\hat{c}_i^\dagger \hat{c}_i \,, \tag{7.107}$$

$$\hat{\sigma}_i^z = -\left(\hat{c}_i + \hat{c}_i^\dagger\right) \prod_{j<i} \left(1 - 2\hat{c}_j^\dagger \hat{c}_j\right) \,, \tag{7.108}$$

where the fermionic operators $\hat{c}_i$ satisfy the anticommutation relations $\{\hat{c}_i, \hat{c}_j^\dagger\} = \delta_{ij}$, $\{\hat{c}_i, \hat{c}_j\} = \{\hat{c}_i^\dagger, \hat{c}_j^\dagger\} = 0$. This transforms the Ising model into the fermion Hamiltonian

$$\hat{H} = -J \sum_{i=1}^{L-1} \left(\hat{c}_i^\dagger \hat{c}_{i+1}^\dagger + \hat{c}_i^\dagger \hat{c}_{i+1} + \text{h.c.}\right) + 2h \sum_{i=1}^{L} \hat{c}_i^\dagger \hat{c}_i - Lh \,. \tag{7.109}$$

As this is a quadratic form in fermion operators, it may be diagonalized by a canonical (Bogoliubov) transformation [79, 80]

$$\hat{c}_i = \sum_{k=1}^{L} \left(u_{ik} \hat{\gamma}_k + v_{ik}^\star \hat{\gamma}_k^\dagger\right) \tag{7.110}$$

which provides the set of Bogoliubov-de Gennes equations

$$\epsilon_k \mathbf{u}_k = A\mathbf{u}_k + B\mathbf{v}_k \tag{7.111}$$

$$\epsilon_k \mathbf{v}_k = -B\mathbf{u}_k - A\mathbf{v}_k \tag{7.112}$$

where $\mathbf{u}_k$ are the elements of $\{u_{ik}, i = 1, \ldots, L\}$ arranged as a vector and likewise for $\mathbf{v}_k$. The matrices A and B are real and tridiagonal, with the nonzero matrix elements $A_{i,i} = 2h$, $A_{i,j} = -J$, $|i-j| = 1$ and $B_{i,i+1} = -B_{i+1,i} = -J$. The pairs (u_{ik}, v_{ik}) with positive energy ϵ_k, $\epsilon_1 \leq \epsilon_2 \leq \cdots \leq \epsilon_L$, and the normalization $|\mathbf{u}_k|^2 + |\mathbf{v}_k|^2 = 1$ define quasiparticle operators

$$\hat{\gamma}_k = u^{\star}_{ik}\hat{c}_i + v_{ik}\hat{c}^{\dagger}_i \tag{7.113}$$

which bring the Hamiltonian into the diagonal form

$$\hat{H} = \sum_{k=1}^{N} \epsilon_k \left(\hat{\gamma}^{\dagger}_k \hat{\gamma}_k - \frac{1}{2}\right) . \tag{7.114}$$

Corresponding to every such pair is another pair $(\tilde{u}_{ik}, \tilde{v}_{ik}) = (v_{ik}, u_{ik})$ with $\tilde{\epsilon}_k = -\epsilon_k$ which defines the conjugate operator $\hat{\gamma}^{\dagger}_k$. Writing $\mathbf{u}_k$ and $\mathbf{v}_k$ together as a composite vector, the Bogoliubov-de Gennes equations take the form of a real symmetric eigenproblem of dimension $2L$:

$$\epsilon_k \begin{pmatrix} \mathbf{u}_k \\ \mathbf{v}_k \end{pmatrix} = \begin{pmatrix} A & B \\ -B & -A \end{pmatrix} \begin{pmatrix} \mathbf{u}_k \\ \mathbf{v}_k \end{pmatrix} \tag{7.115}$$

which can be readily solved using standard eigenvalue methods [81].

Evolution under the fermion Hamiltonian Eq. (7.109) does not preserve the number of fermions N_F but it does preserve the fermionic parity $(-1)^{N_F}$. Because the ground state is the Bogoliubov vacuum it contains no fermions, and so the first *accessible* excited state consists of two Bogoliubov excitations, one in each of the lowest two modes. The gap between the ground and first excited states is thus $\epsilon_1 + \epsilon_2$.

We now consider the Heisenberg equations of motion for the fermi operators

$$i\hbar \frac{d\hat{\mathbf{c}}(t)}{dt} = A(t)\,\hat{\mathbf{c}}(t) + B(t)\,\hat{\mathbf{c}}^{\dagger}(t) \ , \tag{7.116}$$

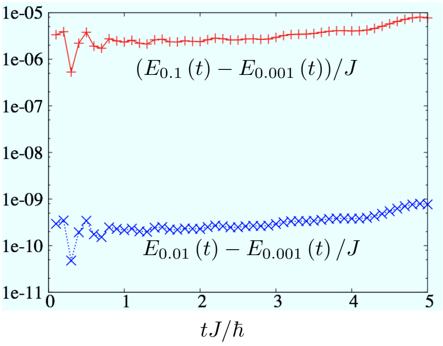


Figure 7.10: *Demonstration of Bogoliubov-de Gennes method convergence.* The errors in the energy computed with a given time step δt, $E_{\delta t}$, are shown as a function of time. The ratio of the errors as a function of time is roughly $(0.01/0.1)^4 = 0.0001$, as should be expected for our fourth-order CFME. Errors in the density of defects behave similarly.

where $A(t)$ and $B(t)$ are the time-dependent generalizations of A and B above and $\hat{\mathbf{c}}(t)$ and $\hat{\mathbf{c}}^{\dagger}(t)$ are the elements of $\{\hat{c}_i(t)\}$ and $\{\hat{c}_i^{\dagger}(t)\}$, respectively, arranged as vectors. Because this equation is linear in the Fermi operators it may be diagonalized with a time-dependent Bogoliubov transformation

$$\hat{c}_i(t) = \sum_{k=1}^{L} \left(u_{ik}(t)\,\gamma_k + v_{ik}^{\star}(t)\,\hat{\gamma}_k^{\dagger} \right) \tag{7.117}$$

where $\mathbf{u}_i(t)$ and $\mathbf{v}_i(t)$ subject to the time-dependent Bogoliubov-de Gennes equations

$$i\hbar\frac{d}{dt}\begin{pmatrix} \mathbf{u}_k(t) \\ \mathbf{v}_k(t) \end{pmatrix} = \begin{pmatrix} A(t) & B(t) \\ -B(t) & -A(t) \end{pmatrix}\begin{pmatrix} \mathbf{u}_k(t) \\ \mathbf{v}_k(t) \end{pmatrix} \equiv \mathcal{H}(t)\begin{pmatrix} \mathbf{u}_k(t) \\ \mathbf{v}_k(t) \end{pmatrix}, \tag{7.118}$$

and $\hat{\gamma}_k$ and $\hat{\gamma}_k^{\dagger}$ diagonalize the Hamiltonian at the initial time. Equivalently, u and v define time-dependent Bogoliubov operators

$$\hat{\gamma}_k(t) = u_{ik}^{\star}(t)\,\hat{c}_i + v_{ik}(t)\,\hat{c}_i^{\dagger} \tag{7.119}$$

such that the time-evolved state $|\psi(t)\rangle$ is the Bogoliubov vacuum of this set, i.e. $\hat{\gamma}_k(t)|\psi(t)\rangle = 0$. To compare with the MPS simulations, we note that the energy at time t is

$$-\frac{1}{2}\sum_k \epsilon_k\left(t\right) = -\frac{1}{2}\sum_k \left(\mathbf{u}_k\left(t\right)\mathbf{v}_k\left(t\right)\right)\mathcal{H}\left(t\right)\begin{pmatrix}\mathbf{u}_k\left(t\right)\\ \mathbf{v}_k\left(t\right)\end{pmatrix}. \tag{7.120}$$

Similarly, the density of defects is

$$\rho\left(t\right) = \frac{1}{2L}\sum_{i=1}^{L-1}\langle\psi\left(t\right)|\left(1-\hat{\sigma}_i^z\hat{\sigma}_{i+1}^z\right)|\psi\left(t\right)\rangle, \tag{7.121}$$

$$= \frac{1}{2L}\sum_{i=1}^{L-1}\left[1-\sum_{k=1}^{L}\left(v_{i,k}\left(t\right)-u_{i,k}\left(t\right)\right)\left(u_{i+1,k}^{\star}\left(t\right)+v_{i+1,k}^{\star}\left(t\right)\right)\right]. \tag{7.122}$$

Time evolution thus reduces to the solution of a $2L \times 2L$ time-dependent matrix differential equation which we solve using a CFME as in Sec. 7.7. Because the dimensions of the system are much smaller than those of the associated MPS problem we are able to take very small time steps, and so the results obtained in the method may be considered to be numerically exact, see Figure 7.10. Here we demonstrate that the error incurred in the energy as a function of time scales with the infinitesimal time step δt as δt^4 using our fourth-order CMFE. Hence, by decreasing δt, any desired degree of accuracy may be met.

7.13 References Cited

[1] T. Wenger T. Kinoshita and D. S. Weiss. A quantum Newton's cradle. *Nature*, 440:900–903, 2006.

[2] S. Trotzky, L. Pollet, F. Gerbier, U. Schnorrberger, I. Bloch, N. V. Prokof'ev, B. Svistunov, and M. Troyer. Suppression of the critical temperature for superfluidity near the Mott transition. *Nat Phys*, 6(12):998–1004, Dec 2010. ISSN 1745-2473. doi: 10.1038/nphys1799. URL http://dx.doi.org/10.1038/nphys1799.

[3] Stefan Trotzky, Yu-Ao Chen, Andreas Flesch, Ian P. McCulloch, Ulrich Schollwöck, Jens Eisert, and Immanuel Bloch. Probing the relaxation towards equilibrium in an isolated strongly correlated 1D Bose gas. *arXiv:1101.2659*, 2011.

[4] K.-K. Ni, S. Ospelkaus, M. H. G. de Miranda, A. Peér, B. Neyenhuis, J. J. Zirbel, S. Kotochigova, P. S. Julienne, D. S. Jin, and Jun Ye. A High Phase-Space-Density Gas of Polar Molecules. *Science*, 322:231–235, 2008.

[5] S. Ospelkaus, K. K. Ni, M. H. G. de Miranda, A. Pe'er, B. Nyenhuis, D. Wang, S. Kotochigova, P. S. Julienne, D. S. Jin, , and Jun Ye. Ultracold polar molecules near quantum degeneracy. *Faraday Discuss.*, 142:351–359, 2009.

[6] J. Deiglmayr, A. Grochola, M. Repp, K. Mörtlbauer, C. Glück, J. Lange, O. Dulieu, R. Wester, and M. Weidemüller. Formation of Ultracold Polar Molecules in the Rovibrational Ground State. *Phys. Rev. Lett.*, 101(13):133004, Sep 2008. doi: 10.1103/PhysRevLett.101.133004.

[7] M. L. Wall and L. D. Carr. Hyperfine molecular Hubbard Hamiltonian. *Phys. Rev. A*, 82(1):013611, Jul 2010. doi: 10.1103/PhysRevA.82.013611.

[8] Alexey V. Gorshkov, Salvatore R. Manmana, Gang Chen, Jun Ye, Eugene Demler, Mikhail D. Lukin, and Ana Maria Rey. Tunable Superfluidity and Quantum Magnetism with Ultracold Polar Molecules. *Phys. Rev. Lett.*, 107:115301, Sep 2011. doi: 10.1103/PhysRevLett.107.115301. URL `http://link.aps.org/doi/10.1103/PhysRevLett.107.115301`.

[9] Alexey V. Gorshkov, Salvatore R. Manmana, Gang Chen, Eugene Demler, Mikhail D. Lukin, and Ana Maria Rey. Quantum magnetism with polar alkali-metal dimers. *Phys. Rev. A*, 84:033619, Sep 2011. doi: 10.1103/PhysRevA.84.033619. URL `http://link.aps.org/doi/10.1103/PhysRevA.84.033619`.

[10] Anatoli Polkovnikov, Krishnendu Sengupta, Alessandro Silva, and Mukund Vengalattore. *Colloquium* : Nonequilibrium dynamics of closed interacting quantum systems. *Rev. Mod. Phys.*, 83:863–883, Aug 2011. doi: 10.1103/RevModPhys.83.863. URL `http://link.aps.org/doi/10.1103/RevModPhys.83.863`.

[11] David Poulin, Angie Qarry, Rolando Somma, and Frank Verstraete. Quantum Simulation of Time-Dependent Hamiltonians and the Convenient Illusion of Hilbert Space. *Phys. Rev. Lett.*, 106:170501, Apr 2011. doi: 10.1103/PhysRevLett.106.170501. URL `http://link.aps.org/doi/10.1103/PhysRevLett.106.170501`.

[12] Ulrich Schollwöck. The density-matrix renormalization group in the age of matrix product states. *Annals of Physics*, 326(1):96 – 192, 2011. ISSN 0003-4916. doi: 10.1016/j.aop.2010.09.012. URL `http://www.sciencedirect.com/science/article/pii/S0003491610001752`. <ce:title>January 2011 Special Issue</ce:title>.

[13] Juan José García-Ripoll. Time evolution of Matrix Product States. *New Journal of Physics*, 8(12):305, 2006. URL `http://stacks.iop.org/1367-2630/8/i=12/a=305`.

[14] Tassilo Keilmann and Juan José García-Ripoll. Dynamical Creation of Bosonic Cooper-Like Pairs. *Phys. Rev. Lett.*, 100:110406, Mar 2008. doi: 10.1103/PhysRevLett.100.110406. URL `http://link.aps.org/doi/10.1103/PhysRevLett.100.110406`.

[15] Jens Eisert and Hans J. Briegel. Schmidt measure as a tool for quantifying multiparticle entanglement. *Phys. Rev. A*, 64:022306, Jul 2001. doi: 10.1103/PhysRevA.64.022306. URL `http://link.aps.org/doi/10.1103/PhysRevA.64.022306`.

[16] D. Gross, S. T. Flammia, and J. Eisert. Most Quantum States Are Too Entangled To Be Useful As Computational Resources. *Phys. Rev. Lett.*, 102(19):190501, May 2009. doi: 10.1103/PhysRevLett.102.190501.

[17] Stellan Östlund and Stefan Rommer. Thermodynamic Limit of Density Matrix Renormalization. *Phys. Rev. Lett.*, 75(19):3537–3540, Nov 1995. doi: 10.1103/PhysRevLett.75.3537.

[18] Stefan Rommer and Stellan Östlund. Class of ansatz wave functions for one-dimensional spin systems and their relation to the density matrix renormalization group. *Phys. Rev. B*, 55(4):2164–2181, Jan 1997. doi: 10.1103/PhysRevB.55.2164.

[19] F. Verstraete, D. Porras, and J. I. Cirac. Density Matrix Renormalization Group and Periodic Boundary Conditions: A Quantum Information Perspective. *Phys. Rev. Lett.*, 93(22):227205, Nov 2004. doi: 10.1103/PhysRevLett.93.227205.

[20] Peter Pippan, Steven R. White, and Hans Gerd Evertz. Efficient matrix-product state method for periodic boundary conditions. *Phys. Rev. B*, 81(8):081103, Feb 2010. doi: 10.1103/PhysRevB.81.081103.

[21] Davide Rossini, Vittorio Giovannetti, and Rosario Fazio. Stiffness in 1D matrix product states with periodic boundary conditions. *Journal of Statistical Mechanics: Theory and Experiment*, 2011(10):P05021, 2011.

[22] D. Perez-Garcia, F. Verstraete, M. M. Wolf, and J. I. Cirac. Matrix Product State Representations. *Quantum Inf. Comput*, 7(401), 2007.

[23] A. Auerbach. *Interacting Electrons and Quantum Magnetism*. Springer, Berlin, 1994.

[24] Ian Affleck, Tom Kennedy, Elliott H. Lieb, and Hal Tasaki. Rigorous results on valence-bond ground states in antiferromagnets. *Phys. Rev. Lett.*, 59:799–802, Aug 1987. doi: 10.1103/PhysRevLett.59.799. URL http://link.aps.org/doi/10.1103/PhysRevLett.59.799.

[25] Hong-Hao Tu, Guang-Ming Zhang, and Tao Xiang. Class of exactly solvable SO(n) symmetric spin chains with matrix product ground states. *Phys. Rev. B*, 78:094404, Sep 2008. doi: 10.1103/PhysRevB.78.094404. URL http://link.aps.org/doi/10.1103/PhysRevB.78.094404.

[26] Steven R. White. Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.*, 69(19):2863–2866, Nov 1992. doi: 10.1103/PhysRevLett.69.2863.

[27] M B Hastings. An area law for one-dimensional quantum systems. *Journal of Statistical Mechanics: Theory and Experiment*, 2007(10):P08024, 2007.

[28] F. Verstraete and J. I. Cirac. Matrix product states represent ground states faithfully. *Phys. Rev. B*, 73(9):094423, Mar 2006. doi: 10.1103/PhysRevB.73.094423.

[29] J. Eisert, M. Cramer, and M. B. Plenio. *Colloquium* : Area laws for the entanglement entropy. *Rev. Mod. Phys.*, 82:277–306, Feb 2010. doi: 10.1103/RevModPhys.82.277. URL http://link.aps.org/doi/10.1103/RevModPhys.82.277.

[30] Christoph Holzhey, Finn Larsen, and Frank Wilczek. Geometric and renormalized entropy in conformal field theory. *Nuclear Physics B*, 424(3):443 – 467, 1994. ISSN 0550-3213. doi: 10.1016/0550-3213(94)90402-2. URL http://www.sciencedirect.com/science/article/pii/0550321394904022.

[31] Pasquale Calabrese and John Cardy. Entanglement entropy and quantum field theory. *Journal of Statistical Mechanics: Theory and Experiment*, 2004(06):P06002, 2004. URL http://stacks.iop.org/1742-5468/2004/i=06/a=P06002.

[32] Ian Affleck, Nicolas Laflorencie, and Erik S Sørensen. Entanglement entropy in quantum impurity systems and systems with boundaries. *Journal of Physics A: Mathematical and Theoretical*, 42(50):504009, 2009. URL http://stacks.iop.org/1751-8121/42/i=50/a=504009.

[33] Pasquale Calabrese and John Cardy. Entanglement entropy and conformal field theory. *Journal of Physics A: Mathematical and Theoretical*, 42(50):504005, 2009. URL http://stacks.iop.org/1751-8121/42/i=50/a=504005.

[34] John Cardy. *Scaling and Renormalization in Statistical Physics*. Cambridge University Press, Cambridge, 1996.

[35] Andreas M Läuchli and Corinna Kollath. Spreading of correlations and entanglement after a quench in the one-dimensional Bose-Hubbard model. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(05):P05018, 2008. URL `http://stacks.iop.org/1742-5468/2008/i=05/a=P05018`.

[36] Matthew S. Block, Ryan V. Mishmash, Ribhu K. Kaul, D. N. Sheng, Olexei I. Motrunich, and Matthew P. A. Fisher. Exotic Gapless Mott Insulators of Bosons on Multileg Ladders. *Phys. Rev. Lett.*, 106:046402, Jan 2011. doi: 10.1103/PhysRevLett.106.046402. URL `http://link.aps.org/doi/10.1103/PhysRevLett.106.046402`.

[37] Norbert Schuch, Ignacio Cirac, and Frank Verstraete. Computational Difficulty of Finding Matrix Product Ground States. *Phys. Rev. Lett.*, 100:250501, Jun 2008. doi: 10.1103/PhysRevLett.100.250501. URL `http://link.aps.org/doi/10.1103/PhysRevLett.100.250501`.

[38] Y.-Y. Shi, L.-M. Duan, and G. Vidal. Classical simulation of quantum many-body systems with a tree tensor network. *Physical Review A (Atomic, Molecular, and Optical Physics)*, 74(2):022320, 2006. doi: 10.1103/PhysRevA.74.022320. URL `http://link.aps.org/abstract/PRA/v74/e022320`.

[39] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins Studies in Mathematical Sciences. The Johns Hopkins University Press, Baltimore, 3 edition, 1996.

[40] Ian P. McCulloch. From density-matrix renormalization group to matrix product states. *Journal of Statistical Mechanics: Theory and Experiment*, 2007 (10):P10014, 2007. URL `http://stacks.iop.org/1742-5468/2007/i=10/a=P10014`.

[41] Tobias J. Osborne. Efficient Approximation of the Dynamics of One-Dimensional Quantum Spin Systems. *Phys. Rev. Lett.*, 97:157202, Oct 2006. doi: 10.1103/PhysRevLett.97.157202. URL `http://link.aps.org/doi/10.1103/PhysRevLett.97.157202`.

[42] Gregory M. Crosswhite and Dave Bacon. Finite automata for caching in matrix product algorithms. *Phys. Rev. A*, 78:012356, Jul 2008. doi: 10.1103/PhysRevA.78.012356. URL `http://link.aps.org/doi/10.1103/PhysRevA.78.012356`.

[43] M. P. A. Fisher, P. B. Weichman, G. Grinstein, and D. S. Fisher. Boson localization and the superfluid-insulator transition. *Phys. Rev. B*, 40:546–570, 1989.

[44] D. Jaksch, C. Bruder, J. I. Cirac, C. W. Gardiner, and P. Zoller. Cold Bosonic Atoms in Optical Lattices. *Phys. Rev. Lett.*, 81(15):3108–3111, Oct 1998. doi: 10.1103/PhysRevLett.81.3108.

[45] B Pirvu, V Murg, J I Cirac, and F Verstraete. Matrix product operator representations. *New Journal of Physics*, 12(2):025012, 2010. URL `http://stacks.iop.org/1367-2630/12/i=2/a=025012`.

[46] Gregory M. Crosswhite, A. C. Doherty, and Guifré Vidal. Applying matrix product operators to model systems with long-range interactions. *Phys. Rev. B*, 78:035116, Jul 2008. doi: 10.1103/PhysRevB.78.035116. URL `http://link.aps.org/doi/10.1103/PhysRevB.78.035116`.

[47] F. Fröwis, V. Nebendahl, and W. Dür. Tensor operators: Constructions and applications for long-range interaction systems. *Phys. Rev. A*, 81:062337, Jun 2010. doi: 10.1103/PhysRevA.81.062337. URL `http://link.aps.org/doi/10.1103/PhysRevA.81.062337`.

[48] M. L. Wall and L. D. Carr. In preparation.

[49] Ernest R. Davidson. The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices. *Journal of Computational Physics*, 17(1):87 – 94, 1975. ISSN 0021-9991. doi: 10.1016/0021-9991(75)90065-0. URL `http://www.sciencedirect.com/science/article/pii/0021999175900650`.

[50] D. Porras, F. Verstraete, and J. I. Cirac. Renormalization algorithm for the calculation of spectra of interacting quantum systems. *Phys. Rev. B*, 73:014410, Jan 2006. doi: 10.1103/PhysRevB.73.014410. URL `http://link.aps.org/doi/10.1103/PhysRevB.73.014410`.

[51] I P McCulloch. Infinite size density matrix renormalization group, revisited, 2008. URL `http://xxx.lanl.gov/abs/0804.2509`.

[52] S. Sachdev. *Quantum Phase Transitions*. Cambridge University Press, New York, 1999.

[53] Francisco Castilho Alcaraz, Miguel Ibáñez Berganza, and Germán Sierra. Entanglement of Low-Energy Excitations in Conformal Field Theory. *Phys. Rev. Lett.*, 106:201601, May 2011. doi: 10.1103/PhysRevLett.106.201601. URL `http://link.aps.org/doi/10.1103/PhysRevLett.106.201601`.

[54] M. Rigol, V. Dunjko, and M. Olshanii. Thermalization and its mechanism for generic isolated quantum systems. *Nature (London)*, 452:854, 2008.

[55] Marcos Rigol. Breakdown of Thermalization in Finite One-Dimensional Systems. *Phys. Rev. Lett.*, 103:100403, Sep 2009. doi: 10.1103/PhysRevLett.103.100403. URL `http://link.aps.org/doi/10.1103/PhysRevLett.103.100403`.

[56] Christian Gogolin, Markus P. Müller, and Jens Eisert. Absence of Thermalization in Nonintegrable Systems. *Phys. Rev. Lett.*, 106:040401, Jan 2011. doi: 10.1103/PhysRevLett.106.040401. URL `http://link.aps.org/doi/10.1103/PhysRevLett.106.040401`.

[57] Pasquale Calabrese and John Cardy. Evolution of entanglement entropy in one-dimensional systems. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(04):P04010, 2005. URL `http://stacks.iop.org/1742-5468/2005/i=04/a=P04010`.

[58] R. V. Mishmash and L. D. Carr. Quantum Entangled Dark Solitons formed by Ultracold Atoms in Optical Lattices. *Phys. Rev. Lett.*, 103:140403, 2009.

[59] R. V. Mishmash, I. Danshita, Charles W. Clark, and L. D. Carr. Quantum Many-Body Dynamics of Dark Solitons in Optical Lattices. *Phys. Rev. A*, 80: 053612, 2009.

[60] J. A. Glick and L. D. Carr. Macroscopic quantum tunneling of solitons in Bose-Einstein condensates. *Phys. Rev. Lett. under review, arXiv:1105.5164*, 2011.

[61] Guifré Vidal. Efficient Classical Simulation of Slightly Entangled Quantum Computations. *Phys. Rev. Lett.*, 91(14):147902, Oct 2003. doi: 10.1103/PhysRevLett.91.147902.

[62] Steven R. White and Adrian E. Feiguin. Real-Time Evolution Using the Density Matrix Renormalization Group. *Phys. Rev. Lett.*, 93:076401, 2004.

[63] A. J. Daley, C. Kollath, U. Schollwöck, and G. Vidal. Time-dependent density-matrix renormalization-group using adaptive effective Hilbert spaces. *Journal of Statistical Mechanics: Theory and Experiment*, 2004(04):P04005, 2004. URL `http://stacks.iop.org/1742-5468/2004/i=04/a=P04005`.

[64] E M Stoudenmire and Steven R White. Minimally entangled typical thermal state algorithms. *New Journal of Physics*, 12(5):055026, 2010. URL `http://stacks.iop.org/1367-2630/12/i=5/a=055026`.

[65] J Schachenmayer, I Lesanovsky, A Micheli, and A J Daley. Dynamical crystal creation with polar molecules or rydberg atoms in optical lattices. *New Journal of Physics*, 12(10):103044, 2010. URL `http://stacks.iop.org/1367-2630/12/i=10/a=103044`.

[66] Peter Schmitteckert. Nonequilibrium electron transport using the density matrix renormalization group method. *Phys. Rev. B*, 70:121302, Sep 2004. doi: 10.1103/PhysRevB.70.121302. URL http://link.aps.org/doi/10.1103/PhysRevB.70.121302.

[67] Wilhelm Magnus. On the exponential solution of differential equations for a linear operator. *Communications on Pure and Applied Mathematics*, 7(4):649–673, 1954. ISSN 1097-0312. doi: 10.1002/cpa.3160070404. URL http://dx.doi.org/10.1002/cpa.3160070404.

[68] S. Blanes and P.C. Moan. Fourth- and sixth-order commutator-free Magnus integrators for linear and non-linear dynamical systems. *Applied Numerical Mathematics*, 56(12):1519 – 1537, 2006. ISSN 0168-9274. doi: 10.1016/j.apnum.2005.11.004. URL http://www.sciencedirect.com/science/article/pii/S0168927405002163.

[69] S. Blanes, F. Casas, J.A. Oteo, and J. Ros. The Magnus expansion and some of its applications. *Physics Reports*, 470(56):151 – 238, 2009. ISSN 0370-1573. doi: 10.1016/j.physrep.2008.11.001. URL http://www.sciencedirect.com/science/article/pii/S0370157308004092.

[70] A. Alvermann and H. Fehske. High-order commutator-free exponential time-propagation of driven quantum systems. *Journal of Computational Physics*, 230(15):5930 – 5956, 2011. ISSN 0021-9991. doi: 10.1016/j.jcp.2011.04.006. URL http://www.sciencedirect.com/science/article/pii/S0021999111002300.

[71] Hans Munthe-Kaas and Brynjulf Owren. Computations in a free lie algebra. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 357(1754):957–981, 1999. doi: 10.1098/rsta.1999.0361. URL http://rsta.royalsocietypublishing.org/content/357/1754/957.abstract.

[72] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge Univ. Press, Cambridge, U.K., 1993.

[73] Y. Saad. Analysis of some Krylov subspace approximations to the matrix exponential operator. *SIAM J. Numer. Anal*, 29:209–228, 1992.

[74] Hochbruck M. and Lubich C. On Krylov Subspace Approximations to the Matrix Exponential Operator. *SIAM J. Numer. Anal.*, 34:1911–1925, 1997. doi: 10.1137/S0036142995280572. URL http://epubs.siam.org/sinum/resource/1/sjnaam/v34/i5/p1911_s1.

[75] Wojciech H. Zurek, Uwe Dorner, and Peter Zoller. Dynamics of a Quantum Phase Transition. *Phys. Rev. Lett.*, 95(10):105701, Sep 2005. doi: 10.1103/PhysRevLett.95.105701.

[76] Jacek Dziarmaga. Dynamics of a Quantum Phase Transition: Exact Solution of the Quantum Ising Model. *Phys. Rev. Lett.*, 95:245701, Dec 2005. doi: 10.1103/PhysRevLett.95.245701. URL `http://link.aps.org/doi/10.1103/PhysRevLett.95.245701`.

[77] Lukasz Cincio, Jacek Dziarmaga, Marek M. Rams, and Wojciech H. Zurek. Entropy of entanglement and correlations induced by a quench: Dynamics of a quantum phase transition in the quantum Ising model. *Phys. Rev. A*, 75(5): 052321, May 2007. doi: 10.1103/PhysRevA.75.052321.

[78] L. D. Carr, D. Demille, R. V. Krems, and Jun Ye. Cold and Ultracold Molecules: Science, Technology, and Applications. *New J. Phys.*, 11:055049, 2009.

[79] Elliott Lieb, Theodore Schultz, and Daniel Mattis. Two soluble models of an antiferromagnetic chain. *Annals of Physics*, 16(3):407 – 466, 1961. ISSN 0003-4916. doi: 10.1016/0003-4916(61)90115-4. URL `http://www.sciencedirect.com/science/article/pii/0003491661901154`.

[80] Pierre Pfeuty. The one-dimensional Ising model with a transverse field. *Annals of Physics*, 57(1):79 – 90, 1970. ISSN 0003-4916. doi: 10.1016/0003-4916(70)90270-8. URL `http://www.sciencedirect.com/science/article/pii/0003491670902708`.

[81] LAPACK package for solution of dense linear algebra problems: `http://www.netlib.org/lapack/f`.

CHAPTER 8

THE INFINITE SIZE VARIATIONAL MATRIX PRODUCT STATE ALGORITHM

Chapter 7 discusses algorithms using matrix product states (MPSs) as variational ansätze for finding eigenstates and time evolution of 1D quantum systems with a finite number of sites. In this chapter we instead focus on a system which has an infinite number of lattice sites[59] and whose Hamiltonian is spatially homogenous. Because of the translational invariance of the Hamiltonian, the eigenstates may also be chosen to be translationally invariant. However, in a many-body system invariance under translations of a single site may be spontaneously broken, resulting in a system which is only translationally invariant under shifts by q sites. Examples of this type of translational symmetry breaking arise in long-range interacting systems at strong coupling [1, 2], where crystalline states of any rational filling occupy finite regions of the phase diagram. As the broken symmetry is associated with a discrete group, the Mermin-Wagner theorem [3, 4] does not apply and so the system can display true long-range order. This has the unfortunate side effect of inducing very strong finite size effects in simulations with open boundary conditions, where MPS algorithms are most efficient. The algorithm presented in this chapter assumes that the system is invariant under translation by a user-specified number of sites q, and then optimizes this *unit cell* of q sites directly in the limit of an infinite system size. Performing the optimization in this limit removes spurious boundary effects which can completely obscure the properties of the system otherwise. A full set of examples are provided in Sec. 8.4 after the theory of the algorithm has been developed.

---

[59] Because we are considering spatially discrete systems, the infinities we encounter are always countable.

An algorithm for optimizing an infinite MPS was present already in the very first DMRG paper [5]. This algorithm, which is known as iDMRG,[60] begins by considering a two-site system. The energy is minimized in this configuration, resulting in the MPS[61]

$$|\psi\rangle = \sum_{i_1 j_1} A^{[1]i_1}\Lambda^{[1]}B^{[1]j_1}|i_1 j_1\rangle\,. \tag{8.1}$$

This minimization, and those that follow, can be performed using the ground state search method presented in Chapter 7. These two sites now form an environment into which two new sites are embedded. In order to maintain a consistent canonical form for the resulting four site MPS, the environment is formed from $A^{[1]}$ and $B^{[1]}$ and the two new sites replace $\Lambda^{[1]}$ as the orthogonality center. These two interior sites are optimized by minimizing the energy with the outer two sites held fixed, resulting in the four-site MPS

$$|\psi\rangle = \sum_{i_1 i_2 j_1 j_2} A^{[1]i_1}A^{[2]i_2}\Lambda^{[2]}B^{[2]j_2}B^{[1]j_1}|i_1 i_2 j_2 j_1\rangle\,. \tag{8.2}$$

The procedure of inserting two new sites and optimizing them with the others held fixed is repeated, leading at the n^{th} iteration to an MPS of the form

$$|\psi\rangle = \sum_{i_1\dots i_n j_1 \dots j_n} A^{[1]i_1}A^{[2]i_2}\dots A^{[n]i_n}\Lambda^{[n]}B^{[n]j_n}\dots B^{[2]j_2}B^{[1]j_1}|i_1\dots i_n j_n \dots j_1\rangle\,. \tag{8.3}$$

Using this method, White and Huse [6] found many of the properties of the spin-1 antiferromagnetic Heisenberg chain an unprecedented precision of twelve digits. In particular, they verified Haldane's conjecture [7] that this model has a finite gap to excitations, and demonstrated the presence of long range string order of the correlator

---

[60] Many MPS algorithms have infinite counterparts which are denoted by attaching an "i" to the beginning of the name, e.g. iTEBD. This naming scheme was developed well before Apple made it trendy to do so.

[61] Here and throughout this section the superscript index in brackets denotes from which iteration of iDMRG the particular tensor was obtained rather than its position in space.

$$g\left(\ell\right)=\langle \hat{S}_0^z \prod_{k=1}^{\ell-1}\left(-1\right)^{\hat{S}_k^z}\hat{S}_{\ell}^z\rangle\,, \tag{8.4}$$

corresponding to a hidden $\mathbb{Z}_2\times\mathbb{Z}_2$ order [8].[62]

While iDMRG produced very good results for this case, it is rarely used to study infinite systems because of strong setbacks in the general case. One of the main setbacks is that iDMRG does not produce a translationally invariant wavefunction. That is to say, after a large number of iterations the resulting state Eq. (8.3) is still an open boundary MPS with the central tensors being increasingly accurate representations of the bulk tensors which would be embedded in the center of an infinite chain. This makes it useful as a means to generate an initial ansatz for a finite-size simulation, but not to study the translationally invariant infinite system. As first realized by McCulloch [9], we can identify a translationally invariant unit cell from the tensors obtained by iDMRG by writing the infinite wavefunction in Vidal canonical form as

$$\begin{aligned}|\psi\rangle=&\sum_{\dots i_k i_{k+1} i_{k+2} i_{k+3},\dots}\dots\left(\Lambda^{[A]}\Gamma^{[A]i_k}\Lambda^{[B]}\Gamma^{[B]i_{k+1}}\right)\left(\Lambda^{[A]}\Gamma^{[A]i_{k+2}}\Lambda^{[B]}\Gamma^{[B]i_{k+3}}\right)\dots\\&\times|\dots i_k i_{k+1} i_{k+2} i_{k+3}\dots\rangle\,,\end{aligned} \tag{8.5}$$

where the repeat unit has been bracketed out. Translating this result into the A and B language using Eqs. (6.43)-(6.44), we have that the current best guess at the unit cell is

$$A^{[n]i}\Lambda^{[n]}B^{[n]j}\left(\Lambda^{[n-1]}\right)^{-1}\,. \tag{8.6}$$

---

[62] For a physical interpretation of string order and the order parameter $g\left(\ell\right)$, see the discussion following Eq. (6.74). While the results given there are for the AKLT state, the AKLT state and the ground state of the spin-1 antiferromagnetic Heisenberg model are in the same quantum phase.

With the realization of this repeat unit, we find that the algorithm can produce approximations to an infinite many-body state which is exactly translationally invariant. Explicitly, the infinite state is

$$
\begin{aligned}
|\psi\rangle = \sum_{\dots i_k j_k i_{k+1} j_{k+1} \dots} & \dots \left(A^{[n] i_k} \Lambda^{[n]} B^{[n] j_k} \left(\Lambda^{[n-1]} \right)^{-1} \right) \left(A^{[n] i_{k+1}} \Lambda^{[n]} B^{[n] j_{k+1}} \left(\Lambda^{[n-1]} \right)^{-1} \right) \dots \\
& \times \left| \dots i_k j_k i_{k+1} j_{k+1} \dots \right\rangle . \qquad (8.7)
\end{aligned}
$$

So far we have considered only unit cells which are two sites in length, as this is the configuration considered in the original iDMRG work. However, the procedure can be applied to unit cells of any length, as required for systems which spontaneously break single-site translational invariance. In order to keep the notation light, we will continue to use the two-site notation for the unit cell. Expressions for larger unit cells may be derived from these by treating i and j as multi-component site indices $\mathbf{i} = \left\{ i_1, \dots, i_{\lfloor q/2 \rfloor} \right\}$,[63] $\mathbf{j} = \left\{ j_{\lfloor q/2 \rfloor + 1}, \dots, j_q \right\}$ and then decomposing $A^{[n]\mathbf{i}}$ as a product of left-canonical matrices $A^{[n] i_1} \dots A^{[n] i_{\lfloor q/2 \rfloor}}$ and likewise for $B^{[n]}$. We will use the symbol q to denote the length of the unit cell in what follows. We also note that the computation time scales linearly in the length of the unit cell.

An important practical consequence of McCulloch's unit cell identification is that we can use the optimal tensors from the $(n-1)^{\mathrm{st}}$ and n^{th} iteration cycles to construct a guess at the optimal wavefunction of the $(n+1)^{\mathrm{st}}$ iteration. The optimal tensor at the n^{th} iteration Eq. (8.3) may be written as

$$
\begin{aligned}
& A^{[1] i_1} \dots A^{[n] i_n} \Lambda^{[n]} B^{[n] j_n} \dots B^{[1] j_1} \\
& = A^{[1] i_1} \dots \left(A^{[n] i_n} \Lambda^{[n]} B^{[n] j_n} \left(\Lambda^{[n-1]} \right)^{-1} \right) \Lambda^{[n-1]} \dots B^{[1] j_1} . \qquad (8.8)
\end{aligned}
$$

Identifying the unit cell, and recognizing that in the limit of an infinite system the presence of one additional unit cell is inconsequential, we can insert our best guess at the unit cell Eq. (8.6) to obtain a guess at the optimal tensor of the $n+1^{\mathrm{th}}$ iteration

---

[63] Here $\lfloor n \rfloor$ represents the smallest integer less than or equal to n.

$$A^{[1]i_1}\dots\left(A^{[n]i_n}\Lambda^{[n]}B^{[n]j_{n+1}}\left(\Lambda^{[n-1]}\right)^{-1}\right)\left(A^{[n]i_{n+1}}\Lambda^{[n]}B^{[n]j_n}\left(\Lambda^{[n-1]}\right)^{-1}\right)\Lambda^{[n-1]}\dots B^{[1]j_1}$$
$$= A^{[1]i_1}\dots A^{[n]i_n}\left(\Lambda^{[n]}B^{[n]j_{n+1}}\left(\Lambda^{[n-1]}\right)^{-1}A^{[n]i_{n+1}}\Lambda^{[n]}\right)B^{[n]j_n}\dots B^{[1]j_1}\,, \tag{8.9}$$

where the last line has bracketed out the guess. In this form it is also clear that, for a two-site unit cell, the optimization procedure optimizes the first site and the second site in the unit cell in alternate iterations. That is, the position of the beginning of the unit cell shifts each iteration. For unit cells of larger than two sites there is greater freedom in how to split the tensor for absorbing into the environment, but practical experience shows that cutting the unit cell in half as in the two-site case works best.[64] This is because this decomposition maintains that the two environments are equally valid representations of their infinite counterparts, avoiding “one-sided” errors.

With these notations, we may now state the infinite-size variational ground state search with MPS (iMPS) algorithm as formulated by McCulloch.

1. *Input*

   Input the matrix product operator representation of the Hamiltonian[65] and a sequence of bond dimensions $\{\chi_i\}$, $i = 1, 2, \dots$. Set $\chi = \chi_1$.

2. *Initialization*

   (a) Construct a finite-size, open boundary simulation on q sites and find the optimal state for the given χ. Bring into mixed canonical form $A^{[1]}\Lambda^{[1]}B^{[1]}$.

   (b) Absorb $A^{[1]}$ and $B^{[1]}$ into the environment using the caching recursions Eqs. (7.44)-(7.46) discussed in Sec. 7.3. Solve for the ground state of a new q-site cell with this environment, and bring into mixed canonical form $A^{[2]}\Lambda^{[2]}B^{[2]}$. Set $n = 2$.

---

[64] For a unit cell with an odd number of sites q, we alternately absorb $\lceil q/2\rceil$ and $\lfloor q/2\rfloor$ into the left environment.

[65] See Chapter 7 for a discussion of matrix product operators.

3. *Iteration*

   (a) Absorb $A^{[n]}$ and $B^{[n]}$ into the environment.

   (b) Initialize a trial guess at the ground state with the new environment $\left(\Lambda^{[n]} B^{[n] j_{n+1}} \left(\Lambda^{[n-1]}\right)^{-1} A^{[n] i_{n+1}} \Lambda^{[n]}\right)$.

   (c) Using the trial guess, find the ground state, and bring into mixed canonical form $A^{[n+1]} \Lambda^{[n+1]} B^{[n+1]}$.

   (d) Check for convergence. If converged, measure desired properties and increment to the next bond dimension χ. If the last χ was computed, exit, otherwise return to 3(a). If convergence is not reached, increment n by one and return to 3(a)

8.1 The Orthogonality Fidelity

If the ground state of the given Hamiltonian has the translational symmetry assumed by the iMPS unit cell ansatz, then it is expected that the iMPS iteration above converges to a fixed point in which the unit cells obtained from concurrent iterations are close in some sense. We can make this intuition precise by introducing the orthogonality fidelity

$$F_{\text{ortho}} = \operatorname{Tr} \sqrt{\sqrt{\rho_R^{[n]}} \rho^{[n-1]} \sqrt{\rho_R^{[n]}}}, \tag{8.10}$$

where the reduced density matrices $\rho_R^{[n]}$ and $\rho^{[n-1]}$ are obtained by tracing over everything to the right of the current unit cell at the n^{th} iteration and to the right of the orthogonality center at the $(n-1)^{\text{st}}$ iteration, respectively. For a mixed-canonical state, $\rho^{[n]} = \Lambda^{[n]\dagger} \Lambda^{[n]}$. We can compute $\rho_R^{[n]}$ by performing a singular value decomposition $\Lambda^{[n]} B^{[n]} = USV$. $\Lambda_R^{[n]}$ is SV, and $\rho_R^{[n]} = \Lambda_R^{[n]\dagger} \Lambda_R^{[n]}$. Now, using the cyclic properties of the trace for finite matrices, F_{ortho} is the sum of the singular values of $\Lambda_R^{[n]} \Lambda^{[n-1]\dagger}$, which is straightforwardly calculated.

8.2 Orthogonalization of Matrix Product States in the Thermodynamic Limit

Let us now turn to measurements using our fixed point MPS. First, we recall that the translationally invariant unit cell is represented most accurately at the n^{th} iteration as

$$A^{[n]i}\Lambda_n B^{[n]j}\left(\Lambda^{[n-1]}\right)^{-1}, \tag{8.11}$$

or, shifting the position of the beginning of the unit cell, as

$$\left(\Lambda^{[n-1]}\right)^{-1} A^{[n]i}\Lambda^{[n]}B^{[n]j}. \tag{8.12}$$

Considering Eq. (8.11) and shifting the orthogonality center to the edge of the unit cell, we find

$$A^{[n]i}A^{[n]j}\Lambda_R^{[n]}\left(\Lambda^{[n-1]}\right)^{-1}. \tag{8.13}$$

Here, the tensor describing the right half of the unit cell, $\Lambda^{[n]}B^{[n]j}$, has been converted to $A^{[n]j}\Lambda_R^{[n]}$, where $A^{[n]j}$ is left-canonical and different from the tensor $A^{[n]i}$ describing the left half of the unit cell. In order to keep the notation light, we will use A for both of these tensors, and they will be distinguished by the indices and the order in which they appear within the unit cell. The appearance of the matrix $P \equiv \Lambda_R^{[n]}\left(\Lambda^{[n-1]}\right)^{-1}$ indicates a deviation from orthonormality for finite F_{ortho}, which is always the case in numerics.[66] By a deviation from orthonormality, we mean that the right basis of $A^{[n]j}$ is not orthogonal, and also that the transfer matrix has a spectral radius different than 1. Similarly, by extracting $\Lambda_L^{[n]}$ from the SVD of $A^{[n]i}\Lambda^{[n]}$ in Eq. (8.12), one can show that the left basis of $A^{[n]i}$ is also not orthogonal. A means to measure the amount of (non)orthogonality in the bases of these MPS tensors is to measure the expectation of the unit operator between these states. This is readily done within the

---

[66] Except in extreme cases, for example if the ground state is a product state.

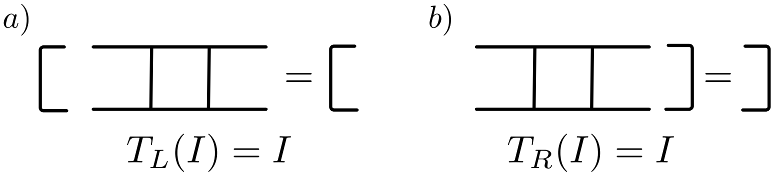


Figure 8.1: Normalization conditions for infinite MPSs. a) For a left-canonical unit cell, the transfer operator of the unit cell admits the identity as a left eigenmatrix with eigenvalue 1. b) For a right-canonical unit cell, the transfer operator of the unit cell admits the identity as a right eigenmatrix with eigenvalue 1.

transfer operator formalism introduced in Sec. 6.5.

We define the transfer operator of a left-canonical unit cell, $T_L(E)$ as

$$T_L(E) = \sum_{ij} P^\dagger {A^j}^\dagger {A^i}^\dagger E A^i A^j P\,, \tag{8.14}$$

where P was defined in the last paragraph. Here E is a $\chi \times \chi$ matrix, and the transfer operator T_L is an operator which takes $\chi \times \chi$ matrices to $\chi \times \chi$ matrices. Also, we note that T_L operates on the right of E. Hence, we use the notation $T_L(E)$ to avoid confusion with the direction of operation. Also, while naively the operation of the transfer operator would require $\mathcal{O}(\chi^4)$ operations, it can be done in $\mathcal{O}(\chi^3)$ by exploiting its tensor network structure.

The usefulness of the transfer operator is best seen by considering the norm of the translationally invariant wavefunction, which is the contraction of an infinite tensor network. If we assume that we have contracted this network from $-\infty$ to 0 and the result is stored in E, then $T_L(E)$ gives the result of this contraction being carried through one more unit cell. Hence, for an orthonormal state, we would have that $T_L(I) = I$, see Figure 8.1. Stated equivalently, the identity matrix is a left eigenmatrix of the transfer operator with eigenvalue 1. However, in the present case, $T_L(I) = P^\dagger P$. We can remedy this [10] by solving for the largest eigenmatrix of $T_L(I)$, call it V_L. Here it should be noted that $T_L(I)$ is not symmetric in general. Also, because of the tensor network structure of $T_L(E)$, a sparse eigensolver should be used to find the eigenmatrix corresponding to the largest eigenvalue. In practice we use the routine `dnaupd` from the Arnoldi-based ARPACK [11] package for non-symmetric sparse matrices. Because of the "quadratic" form of $T_L(I)$, V_L is Hermitian and positive definite when the largest eigenvalue is non-degenerate, and hence we can decompose it into $V_L = X^\dagger X$ where X is invertible. While this is suggestive of a Cholesky decomposition [12], Cholesky decomposition becomes unstable when the matrix is nearly singular [13] and so it is advisable to rather perform an eigenvalue decomposition $V_L = U\Lambda U^\dagger$ with U the matrix with eigenvectors of V_L as columns and then $X = \sqrt{\Lambda}U^\dagger$. We then transform the unit cell via a similarity transformation $XA^iA^jPX^{-1}$ whence $T_L(I) = I$, as desired. The case of degenerate maximal eigenvalue will be discussed at the end of this subsection.

Shifting the starting point of the unit cell as in Eq. (8.12) and moving the orthogonality center to the left, we have that the right-canonical unit cell is

$$QB^iB^j\,, \tag{8.15}$$

where $Q \equiv \left(\Lambda^{[n-1]}\right)^{-1}\Lambda_L^{[n]}$. The interpretations of B^i and Λ^L are parallel to that of A^j and Λ^R above. We stress that this unit cell is independent of X, and so all operations here are compatible with the change of basis of the unit cell such that the right basis is orthogonal. For the transfer operator of a right-canonical unit cell

$$T_R(E) = \sum_{ij} QB^iB^jEB^{j\dagger}B^{i\dagger}Q^\dagger\,, \tag{8.16}$$

we hence have that $T_R(I) = QQ^\dagger$. Proceeding as before, we find the (right) eigenmatrix V_R associated with the largest eigenvalue and decompose it as $V_R = YY^\dagger$ via an eigenvalue decomposition, $V_R = U\Lambda U^\dagger$, $Y = U\sqrt{\Lambda}$. We then transform the unit cell as $Y^{-1}QB^iB^jY$. If we reinsert all definitions, we find that the properly orthogonalized unit cell may be written as

$$X A^{[n]i} \Lambda^{[n]} B^{[n]j} Y Y^{-1} \left(\Lambda^{[n-1]}\right)^{-1} X^{-1} . \tag{8.17}$$

This may be brought into the original form of the unit cell

$$A^{[n]i} \Lambda^{[n]} B^{[n]j} \left(\Lambda^{[n-1]}\right)^{-1} , \tag{8.18}$$

by identifying the new set of tensors

$$A^{[n]i} = X A^{[n]i} , \tag{8.19}$$
$$B^{[n]j} = B^{[n]j} Y , \tag{8.20}$$
$$\Lambda^{[n-1]} = X \Lambda^{[n-1]} Y . \tag{8.21}$$

The identification of the tensors Eqs (8.19)-(8.21) has the benefit of not requiring the inverses of X and Y, but only of $X\Lambda^{[n-1]}Y$. This set of transformations gives us a translationally invariant unit cell with proper left and right orthonormalization ensured. In what follows, when we write e.g. A_n^i we refer to the orthonormalized tensor and not the raw output from the iMPS iteration.

We can now discuss canonical forms for the translationally invariant MPS state. By writing down a product of several unit cells

$$\dots A^{[n]i} \Lambda^{[n]} B^{[n]j} \left(\Lambda^{[n-1]}\right)^{-1} A^{[n]i} \Lambda^{[n]} B^{[n]j} \left(\Lambda^{[n-1]}\right)^{-1} A^{[n]i} \Lambda^{[n]} B^{[n]j} \left(\Lambda^{[n-1]}\right)^{-1} \dots , \tag{8.22}$$

we can identify three different canonical forms paralleling the canonical forms for finite MPSs. The first is fully left-canonical, and is obtained by $A^j = \Lambda^{[n]} B^{[n]j} \left(\Lambda^{[n-1]}\right)^{-1}$:

$$\dots \left(A^{i_1} A^{j_1}\right) \left(A^{i_2} A^{j_2}\right) \dots . \tag{8.23}$$

Here, the parentheses denote unit cells, and the ordering of the indices now denotes position in the lattice from left to right rather than the optimization cycle index, compare Eq. (8.3). By the properties of our orthonormalization procedure above, these tensors are truly left-canonical in the sense of Eq. (6.24). The second is fully right-canonical, and is obtained by $B^i = \left(\Lambda^{[n-1]}\right)^{-1} A^{[n]i}\Lambda^{[n]}$:

$$\dots\left(B^{i_1}B^{j_1}\right)\left(B^{i_2}B^{j_2}\right)\dots\,. \tag{8.24}$$

Again, these matrices are all right-canonical in the sense of Eq. (6.33). The mixed-canonical state is obtained by inserting the identity $I = \Lambda^{[n-1]}\left(\Lambda^{[n-1]}\right)^{-1}$ after one of the $\left(\Lambda^{[n-1]}\right)^{-1}$ and using the identifications of A^j and B^i to find

$$\dots\left(A^{i_1}A^{j_1}\right)\Lambda^{[n-1]}\left(B^{i_2}B^{j_2}\right)\dots\,. \tag{8.25}$$

This is the form most useful for expectation values. Here, the parentheses denote the left-canonical and right-canonical unit cells from Eq. (8.23) and Eq. (8.24). The left-canonical unit cell repeats infinitely to the left of $\Lambda^{[n-1]}$, and the right-canonical unit cell repeats infinitely to the right of $\Lambda^{[n-1]}$. We note that the compatibility of Eqs. (8.23), (8.24), and (8.25) implies that T_R admits $\Lambda^{[n-1]\dagger}\Lambda^{[n-1]}$ as a left eigenmatrix with eigenvalue 1 and similarly T_L admits $\Lambda^{[n-1]}\Lambda^{[n-1]\dagger}$ as a right eigenmatrix with eigenvalue 1. This can also be verified directly.

In the case of a degenerate maximal eigenvalue, the left and right eigenmatrices corresponding to the maximal eigenvalue we find are not unique, but form a basis for the degenerate eigenspace. Hence, the ordering of the numerically obtained eigenmatrices may be such that $\langle k|k\rangle$ is very close to zero, where $\langle k|$ is the k^{th} left eigenmatrix, $|k\rangle$ is the k^{th} right eigenmatrix, and we use the Frobenius inner product. This can cause severe instability in the formation of the properly orthogonalized tensors given in Eqs. (8.19)-(8.21). To avoid this difficulty, we form the Gram matrix of the left and right eigenmatrices

$$M_{kk'} = \langle k|k'\rangle\,, \tag{8.26}$$

and perform a singular value decomposition $M = USV$. We now unitarily transform the left-and right eigenmatrices as

$$|k\rangle = V_{kk'}|k'\rangle \,, \tag{8.27}$$
$$\langle k| = U_{k'k}\langle k'| \,, \tag{8.28}$$

such that their Gram matrix is now the positive diagonal matrix of singular values S. This is effectively a transformation of the left and right eigenmatrices to the well-conditioned subspace. In practice we perform this procedure on any set of Hermitian and positive definite eigenmatrices whose eigenvalues differ in a relative sense by 10^{-4},[67] as a quasi-degeneracy resulting in a large correlation length may be indicative of a true degeneracy that is not converged in χ.

8.3 Expectation Values

We now turn to expectation values. Here we will use the mixed-canonical form Eq. (8.25). Without loss of generality, we assume that the support[68] of our observable operator contains the orthogonality center. On the right of the rightmost operator is an infinite product of right canonical matrices B. Due to the normalization of the state and the right canonical condition shown graphically in Figure 8.1, the result of the infinite contraction of right-canonical matrices becomes $\lim_{N\to\infty} T_R^N(I) = I$. Similarly, on the left of the leftmost operator we have an infinite product of left-canonical matrices A, yielding $\lim_{N\to\infty} T_L^N(I) = I$. Thus, the calculation of expectation values of operators with finite support is obtained by contracting the finite network on which our expectation operators have support and tracing the boundaries, see Figure 8.2. We will turn to the question of operators with infinite support,[69] whose expectation

---

[67] This value corresponds to a correlation length of roughly 10^4 lattice sites. The reason for this choice is practical; current experimental setups using ultracold gases would have difficulty distinguishing a state with a correlation length this large from a state which is truly long-range ordered. Hence, explicitly breaking symmetries by mixing near-degenerate eigenstates for the benefit of numerical stability will not affect our ability to predict experimental outcomes.

[68] The support of an operator is the region of the lattice on which the operator is nonzero.

[69] That is, expectations of operators which act on all sites in the infinite lattice.

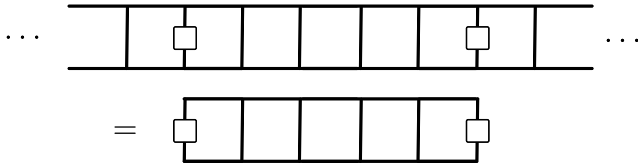


Figure 8.2: Expectation of a finitely-supported operator within an infinite MPS. Due to the left- and right-canonical conditions shown graphically in Figure 8.1, the expectation of an operator whose finite support contains the orthogonality center may be evaluated as a finite tensor network contraction using the methods of Sec. 7.5.

values diverge for the infinite state, in the next paragraph. When computing two-point correlation functions e.g. $\langle \hat{n}_0 \hat{n}_r \rangle$, we find that best results are obtained when the result is averaged over all possible separations r consistent with translation invariance. Namely, we compute

$$\langle \hat{n}_0 \hat{n}_r \rangle = \frac{1}{q} \sum_{i=0}^{q-1} \langle \hat{n}_i \hat{n}_{i+r} \rangle \, , \tag{8.29}$$

where q is the number of sites in the unit cell and the summation runs over all sites in the unit cell. This accounts for possible breaking of the translational symmetry within the unit cell itself.

We now turn our attention to finding the expectation value of an operator representing an extensive observable, taking the Hamiltonian as a paradigmatic example [14]. This amounts to finding the energy density of the infinite state. We make the assumption that the expectation of the observable grows at most extensively such that it has a finite density in the thermodynamic limit. This holds for all reasonable Hamiltonians, but not for example for the square of the Hamiltonian.[70] The Hamiltonian is assumed to be spatially homogenous, and so its matrix product operator (MPO) representation is specified by a single MPO matrix W. To be concrete, let us consider the MPO representation of the Ising model in a transverse field

---

[70] This can be seen by considering the expectation of $\hat{H}^2$ in the ground state, which is $e_0 L^2$ with e_0 the energy density and $L \to \infty$ the number of lattice sites.

$$\hat{H}_{\text{Ising}} = -J \sum_{\langle i,j \rangle} \hat{\sigma}_i^z \hat{\sigma}_j^z - h \sum_i \hat{\sigma}_i^x \, , \tag{8.30}$$

which is given by

$$W = \begin{pmatrix} \hat{I} & 0 & 0 \\ \hat{\sigma}^z & 0 & 0 \\ -h\hat{\sigma}^x & -J\hat{\sigma}^z & \hat{I} \end{pmatrix} . \tag{8.31}$$

Here $\hat{\sigma}^\nu$ denotes the Pauli matrix along the ν^{th} Cartesian direction. This MPO has a bond dimension $\chi_O = 3$.

Assuming the state to be in right-canonical form as in Eq. (8.24), the expectation $\langle \hat{H} \rangle$ becomes an infinitely large contraction of the form

$$\langle \hat{H} \rangle = \operatorname{Tr}\left[\dots \left(\sum_{ii'} B^{i\star} \otimes W^{ii'} \otimes B^{i'} \right) \left(\sum_{jj'} B^{j\star} \otimes W^{jj'} \otimes B^{j'} \right) \dots \right] . \tag{8.32}$$

Comparing with Eq. (6.65) suggests defining a generalized unit cell transfer operator as

$$T_{R;X}\left(E\right) = \sum_{iji'j'} X^{ii';jj'} B^{i'} B^{j'} E B^{j\dagger} B^{i\dagger} \, , \tag{8.33}$$

where $X^{ii';jj'}$ is an operator acting on the local indices of the unit cell. Defining the elements of the MPO representation of our operator acting on the unit cell as

$$\mathcal{W}_{\kappa\kappa'}^{ii'jj'} = \sum_{\kappa''} W_{\kappa\kappa''}^{ii'} W_{\kappa''\kappa'}^{jj'} \, , \tag{8.34}$$

we can write the infinite contraction in terms of the generalized transfer operator Eq. (8.33) as shown graphically in Figure 8.3.

The general procedure of evaluating the infinite product of transfer operators is similar to the procedure used to normalize infinite MPSs. In the normalization procedure, the restriction that the state have a norm of one implied that the transfer

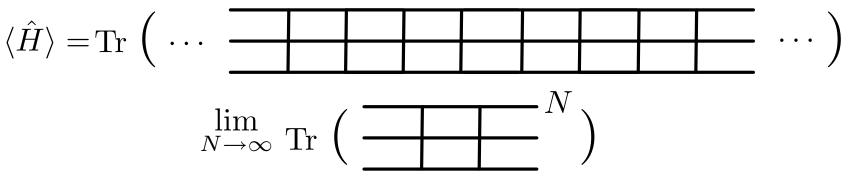


Figure 8.3: Expectation of an infinitely-supported operator within an infinite MPS. Because of the repeating structure of the unit cell and the homogenous MPO representation, we can identify a repeating tensor network structure which is the generalized transfer operator $T_{R;\mathcal{W}_{\kappa\kappa'}}$. The uncontracted bonds in the center of the expectation correspond to the bond indices of the MPO.

operator had a dominant eigenmatrix with eigenvalue 1. The left- or right-canonical form of the unit cell then allowed us to identify that this dominant eigenmatrix was in fact the identity operator. In the present case, the dominant matrix will have χ_O components, $(E_1, \dots, E_{\chi_O})$, where χ_O is the bond dimension of the MPO. To outline the general structure of the dominant matrices, let us consider multiplying[71] the matrix W in Eq. (8.31) by itself L times to find

$$W^L = \begin{pmatrix} \hat{I}^{\otimes L} & 0 & 0 \\ \hat{I}^{\otimes(L-1)} \otimes \hat{\sigma}^z & 0 & 0 \\ \hat{H}_{\mathrm{Ising}} & -J\hat{\sigma}^z \otimes \hat{I}^{\otimes(L-1)} & \hat{I}^{\otimes L} \end{pmatrix} . \tag{8.35}$$

This is indicative of the general structure of an MPO. In particular, the two halves of the bond term $\hat{\sigma}^z\hat{\sigma}^z$ have a string of identities appended to them to become $\hat{I}^{\otimes(L-1)} \otimes \hat{\sigma}^z$ and $\hat{\sigma}^z \otimes \hat{I}^{\otimes(L-1)}$. In DMRG, these operator strings are referred to as *connection operators*, as they specify how the Hamiltonian connects the L sites acted on by this MPO product to the rest of the system. Furthermore, the lower left element of the MPO is the Hamiltonian on L sites, and the upper left and lower right components are the identity on L sites. Because the only term in the first row is the identity, E_1 is an eigenmatrix of T_R with eigenvalue 1 provided that the unit cell is in right-canonical form. Hence, the first component of the dominant eigenmatrix is the identity matrix.

---

[71] By multiplication, we mean contraction over the bond indices with the matrix-valued elements combined by tensor products, see Sec. 7.2.2.

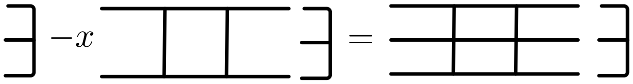


Figure 8.4: Fixed point relations for the dominant matrices of the Hamiltonian MPO transfer matrix, Eq. (8.37).

To find the other components of the dominant matrix, let us consider that we have performed the expectation Eq. (8.32) over n unit cells starting with some boundary matrices $\{E_\kappa(0)\}$ and have stored the result in the matrices $\{E_\kappa(n)\}$. Here n denotes how over how many unit cells the contraction has been performed. Carrying out the contraction over one further unit cell produces the matrix

$$E_\kappa(n+1)=\sum_{\kappa'}T_{R;\mathcal{W}_{\kappa\kappa'}}\left(E_{\kappa'}(n)\right)\,. \tag{8.36}$$

We now focus on the case where all infinite-ranged terms in the Hamiltonian are free of Fermi phases.[72] In these cases any operator appearing on the diagonal of the MPO matrix $\mathcal{W}$ is proportional to the identity. Separating out this part of the MPO explicitly, we have the fixed point relations

$$\left(I-xT_R\right)\left(E_\kappa\right)=\sum_{\kappa'<\kappa}T_{R;\mathcal{W}_{\kappa\kappa'}}\left(E_{\kappa'}\right)\,. \tag{8.37}$$

In Eq. (8.37) we have dropped the arguments n with the understanding that the dominant matrices are fixed points of Eq. (8.37) and are hence independent of n.

The crucial component in solving Eq. (8.37) is that, because of the lower triangular structure and the fact that E_1 is known a priori, the right hand side of Eq. (8.37) is a known matrix, and hence Eq. (8.37) represents a non-symmetric system of linear equations for E_κ, see Figure 8.4. For $\kappa\neq\chi_O$, there are two possibilities. The first is that there are no long range terms, and so $x=0$. In this case there is no linear system of equations, but rather an equality for the unknown matrix E_κ. The second is that

---

[72] This assumption does not hold for long-range tunneling of fermions, for example. Such cases can be dealt with, but are not relevant for this thesis.

there is a long-range interaction with $x < 1$.[73] In this case we do have to solve a linear system of equations, but the matrix on the left hand side is nonsingular due to the fact that the spectral radius of T_R is 1. We solve these using the GMRES method [15], a Krylov-subspace based sparse linear solver for non-symmetric matrices. This allows us to take advantage of the tensor network structure when applying T_R so that the solution is $\mathcal{O}\left(\chi^3\right)$. In the case of the Ising model with MPO given by Eq. (8.31), all elements with $1 < \kappa < \chi_O$ have $x = 0$, and so the associated $\{E_\kappa(n)\}$ can be solved for explicitly. These matrices do not depend on the initial values $E_\kappa(0)$ provided that $E_1(0) = I$, and so are the fixed points $\{E_\kappa\}$ of Eq. (8.37).

Finally, we turn to $\kappa = \chi_O$. Here we have $x = 1$, and so the left-hand matrix in Eq. (8.37) is right-singular for components of E_{χ_O} along the identity. However, the lower-left element of a product of L MPO matrices is the Hamiltonian on L sites, see Eq. (8.35), and so $E_{\chi_O}(n+1)$ represents the overlap of the Hamiltonian on n unit cells in the basis states of the unit cell MPS decomposition. Hence, the singularity arises from the fact that this expectation can take on any value $e_0 q n$, where e_0 is the energy density, q is the number of lattice sites, and n is a positive integer. That is to say, the result of carrying the contraction through one more unit cell is

$$E_\kappa(n+1) = -q e_0 I \delta_{\kappa,\chi_O} + \sum_{\kappa'} T_{R;\mathcal{W}_{\kappa\kappa'}}\left(E_{\kappa'}(n)\right)\,, \tag{8.38}$$

where the constant piece is the energy expectation of the unit cell. We can remove this ambiguity by using the fact that $\Lambda^{[n-1]^\dagger}\Lambda^{[n-1]}$ spans the left null space of the matrix $(I - T_R)$ to find

$$q e_0 = \mathrm{Tr}\left[\Lambda^{[n-1]^\dagger}\Lambda^{[n-1]} \sum_{\kappa' < \chi_O} T_{R;\mathcal{W}_{\kappa\kappa'}}\left(E_{\kappa'}\right)\right]\,. \tag{8.39}$$

---

[73] x=1 corresponds to uniform long-range interactions, and we know of no physical real-space Hamiltonians involving such terms.

This provides us with the energy density. Hence, the relevant fixed-point condition for the dominant matrices is not Eq. (8.37) but

$$(I - xT_R)(E_\kappa) = -qe_0 I \delta_{\kappa,\chi_O} + \sum_{\kappa' < \kappa} T_{R;\mathcal{W}_{\kappa\kappa'}}(E_{\kappa'}) \,, \tag{8.40}$$

which differs from Eq. (8.37) only for $\kappa = \chi_O$. The fact that the energy density takes the form Eq. (8.39) also implies that the right hand side of Eq. (8.40) is orthogonal to the left null space of the left hand side matrix. By the fundamental theorem of linear algebra [16], the right hand side lies completely within the image of the operator on the left-hand side, and so this equation has a consistent solution and not merely a least-squares or pseudo-inverse solution in spite of the fact that the operator on the left hand side is singular. This construction contrasts with the method taken in Ref. [14], in which the extensive part of the observable is not included in Eq. (8.40) and a projection procedure within the GMRES method is used instead. The fact that we can add any multiple of the identity matrix to the dominant matrix E_{χ_O} which solves Eq. (8.40) and still have a solution to Eq. (8.40) corresponds to the freedom in choosing a zero of energy for the infinite system. Using Eq. (8.39) we get the energy per site several orders of magnitude more accurately than we obtain from raw output of the effective Hamiltonian eigenequation used to optimize the iMPS.

8.4 Examples

To demonstrate the iMPS method, we will study the spin-1/2 and spin-1 antiferromagnetic Heisenberg models. Antiferromagnetic Heisenberg models have long been of interest in studies of quantum magnetism, where they represent a minimal Hamiltonian describing particles whose spins wish to be anti-aligned with their neighbors [17]. The spin-1/2 model is exactly solvable via Bethe ansatz [18, 19], and is known to be critical with central charge 1 and z-z correlations decaying asymptotically as

$$\langle \hat{S}_0^z \hat{S}_r^z \rangle \sim (-1)^r \, \frac{\sqrt{\ln |r|}}{|r|} \, . \tag{8.41}$$

The z-z correlation function describes how strongly the z-component of a spin at position 0 is correlated with the z-component of a spin r sites away. Experimentally, this information is probed via the dynamical structure factor

$$S\left(\mathbf{q}, \omega\right) = L^{-1} \int_{-\infty}^{\infty} dt e^{i\omega t} \sum_{i,j=1}^{L} e^{i\mathbf{q}\cdot\left(\mathbf{r}_i - \mathbf{r}_j\right)} \langle \hat{S}_i^z\left(t\right) \hat{S}_j^z\left(0\right) \rangle \, , \tag{8.42}$$

where $\hat{S}_i^z\left(t\right)$ is a Heisenberg picture operator and L is the number of sites [17]. In one dimension, true long-range ferromagnetic order where $\langle \hat{S}_0^z \hat{S}_r^z \rangle$ approaches a constant as $r \to \infty$ is destroyed by strong fluctuations according to the Mermin-Wagner theorem [3, 4], and so power law decay represents the strongest allowed degree of correlation. Furthermore, its nearest-neighbor and next-nearest neighbor z-z correlations are known from the Bethe ansatz solution to be $\frac{1}{12}\left(1 - 4\log 2\right) \approx -0.14771573$ [19] and $\frac{1}{12}\left(1 - 16\log 2 + 9\,\zeta\left(3\right)\right) \approx 0.06067977$ [20]. The spin-1 model has no properties which are amenable to exact computation. However, its properties were of great interest after Haldane [7] conjectured a major difference between integer-spin and half integer-spin antiferromagnetic Heisenberg models. In particular, as opposed to the half-integer case where the system is critical and correlations decay algebraically, integer-spin chains are gapped and non-critical, and z-z correlations decay exponentially as

$$\langle \hat{S}_0^z \hat{S}_r^z \rangle \sim (-1)^r \, \frac{\exp\left(-\left|r\right|/\xi\right)}{|r|} \, . \tag{8.43}$$

DMRG found one of its important early applications for this system in which Haldane's conjecture was verified, and the gap and correlation length ξ were both calculated to high precision [6].

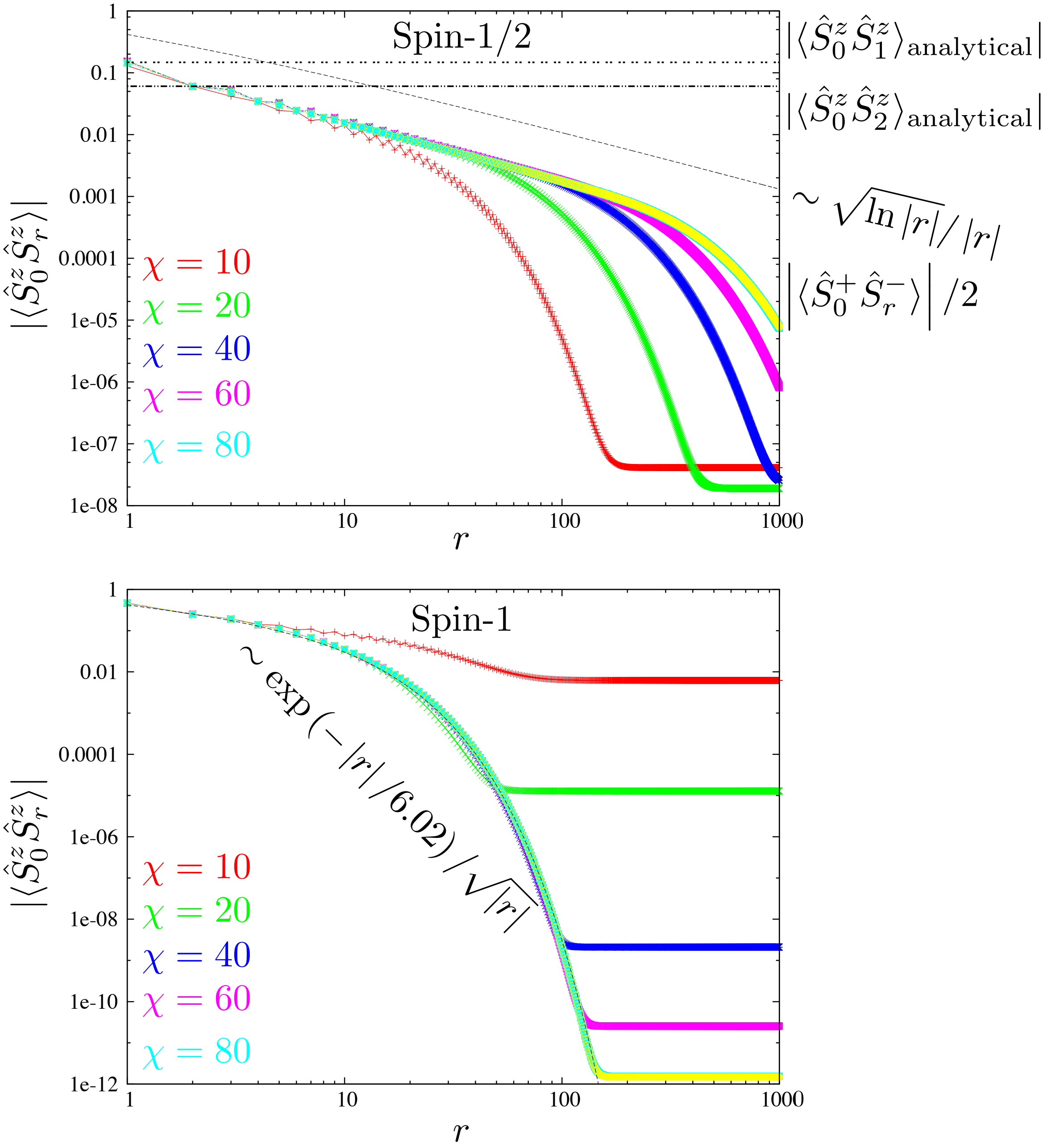


Figure 8.5: Correlation functions for the spin-1/2 and spin-1 Heisenberg models computed with iMPS. Top panel: The z-z correlation functions for the spin-1/2 antiferromagnetic Heisenberg model is compared with exactly known results. As the bond dimension is increased, the range over which this function follows the true algebraic decay increases, compare Figure 6.5. Bottom panel: The z-z correlation function for the spin-1 antiferromagnetic Heisenberg model is shown together with a fit demonstrating its exponential decay.

In Figure 8.5, we show the z-z correlation function of these two models computed from converged iMPS simulations with a range of bond dimensions χ. In the upper plot for the spin-1/2 model, we also plot the nearest-neighbor and next-nearest-neighbor correlations as horizontal lines to accentuate that even MPSs with small bond dimension $\chi = 10$ capture the short-distance physics well. The tilted dashed line is a function with the same scaling as the Bethe ansatz result Eq. (8.41). As the bond dimension increases, the power law behavior of the correlator persists to further distances. Hence, what is more important than the value of the correlator for a single χ is the scaling behavior of the correlator as χ is increased. Plotted in yellow, which lies on top of the light blue line, is the $\langle \hat{S}_0^+ \hat{S}_r^- \rangle/2$ correlator to demonstrate that spin-rotational symmetry is preserved by the simulation. As is always the case for MPSs with a finite bond dimension that do not have long-range order, at large enough distances the behavior of the correlator always becomes exponential with a correlation length equal to the largest correlation length of the transfer operator. This behavior may also be compared with the approximation of a power law by a sum of exponentials shown in Figure 6.5. In the bottom panel of Figure 8.5 we show the analogous plot for the spin-1 chain. Here the correlator is exponential, as shown via the fit through the correlator with the largest bond dimension.

In Figure 8.6, we display the scaling of the largest correlation length obtained from the sub-leading eigenvalue of the transfer operator with the bond dimension. The critical spin-1/2 state shows an increase in the correlation length as the bond dimension increases, and a fit to the prediction Eq. (6.75) yields a central charge in good agreement with the known value. Also shown in this plot is the behavior of the spin-1 chain, in which the saturation of the correlation length to a finite value implies the absence of criticality. Typical values of χ for MPS/DMRG studies on finite lattices are a few hundreds to a few thousands, with values of tens of thousands being reported for quasi-2D systems [21, 22]. In finite-size MPS studies one must first

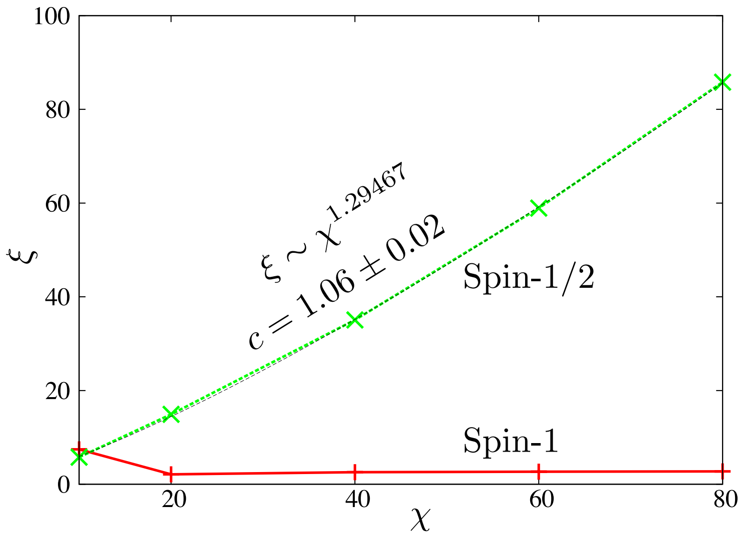


Figure 8.6: Scaling of the correlation length with the bond dimension χ in iMPS. The spin-1/2 chain obeys the scaling relation Eq. (6.75) between the correlation length and the bond dimension, yielding a central charge in agreement with the analytical prediction. The spin-1 chain shows a saturation of the correlation length with the entanglement cutoff, and is hence non-critical.

extrapolate $\chi \to \infty$ before one can perform reliable finite-size scaling $L \to \infty$. By instead taking the limit $L \to \infty$ first and performing finite-entanglement scaling, we can obtain very good results using comparatively rather modest values of the bond dimension χ.

8.5 References Cited

[1] M. Dalmonte, G. Pupillo, and P. Zoller. One-Dimensional Quantum Liquids with Power-Law Interactions: The Luttinger Staircase. *Phys. Rev. Lett.*, 105:140401, Sep 2010. doi: 10.1103/PhysRevLett.105.140401. URL http://link.aps.org/doi/10.1103/PhysRevLett.105.140401.

[2] B. Capogrosso-Sansone, C. Trefzger, M. Lewenstein, P. Zoller, and G. Pupillo. Quantum Phases of Cold Polar Molecules in 2D Optical Lattices. *Phys. Rev. Lett.*, 104:125301, 2010.

[3] N. D. Mermin and H. Wagner. Absence of Ferromagnetism or Antiferromagnetism in One- or Two-Dimensional Isotropic Heisenberg Models. *Phys. Rev. Lett.*, 17:1133–1136, Nov 1966. doi: 10.1103/PhysRevLett.17.1133. URL http://link.aps.org/doi/10.1103/PhysRevLett.17.1133.

[4] P. C. Hohenberg. Existence of Long-Range Order in One and Two Dimensions. *Phys. Rev.*, 158:383–386, Jun 1967. doi: 10.1103/PhysRev.158.383. URL `http://link.aps.org/doi/10.1103/PhysRev.158.383`.

[5] Steven R. White. Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.*, 69(19):2863–2866, Nov 1992. doi: 10.1103/PhysRevLett.69.2863.

[6] Steven R. White and David A. Huse. Numerical renormalization-group study of low-lying eigenstates of the antiferromagnetic S =1 Heisenberg chain. *Phys. Rev. B*, 48:3844–3852, Aug 1993. doi: 10.1103/PhysRevB.48.3844. URL `http://link.aps.org/doi/10.1103/PhysRevB.48.3844`.

[7] F. D. M. Haldane. Nonlinear Field Theory of Large-Spin Heisenberg Antiferromagnets: Semiclassically Quantized Solitons of the One-Dimensional Easy-Axis Néel State. *Phys. Rev. Lett.*, 50:1153–1156, Apr 1983. doi: 10.1103/PhysRevLett.50.1153. URL `http://link.aps.org/doi/10.1103/PhysRevLett.50.1153`.

[8] Tom Kennedy and Hal Tasaki. Hidden $Z_2 \times Z_2$ symmetry breaking in Haldane-gap antiferromagnets. *Phys. Rev. B*, 45:304–307, Jan 1992. doi: 10.1103/PhysRevB.45.304. URL `http://link.aps.org/doi/10.1103/PhysRevB.45.304`.

[9] I P McCulloch. Infinite size density matrix renormalization group, revisited, 2008. URL `http://xxx.lanl.gov/abs/0804.2509`.

[10] R. Orús and G. Vidal. Infinite time-evolving block decimation algorithm beyond unitary evolution. *Phys. Rev. B*, 78:155117, Oct 2008. doi: 10.1103/PhysRevB.78.155117. URL `http://link.aps.org/doi/10.1103/PhysRevB.78.155117`.

[11] ARPACK package for sparse solution of eigenvalue problems: `http://www.caam.rice.edu/software/ARPACK/`.

[12] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins Studies in Mathematical Sciences. The Johns Hopkins University Press, Baltimore, 3 edition, 1996.

[13] J. H. Wilkinson. *Rounding errors in algebraic processes*. Prentice-Hall, Englewood Cliffs, N.J., 1964.

[14] I. P. McCulloch L. Michel. Schur forms of matrix product operators in the infinite limit, 2010. URL `http://arxiv.org/abs/1008.4667`.

[15] Y. Saad and M. Schultz. GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986. doi: 10.1137/0907058. URL `http://epubs.siam.org/doi/abs/10.1137/0907058`.

[16] Gilbert Strang. The Fundamental Theorem of Linear Algebra. *The American Mathematical Monthly*, 100(9):pp. 848–855, 1993. ISSN 00029890. URL `http://www.jstor.org/stable/2324660`.

[17] A. Auerbach. *Interacting Electrons and Quantum Magnetism*. Springer, Berlin, 1994.

[18] H. Bethe. Zur Theorie der Metalle. *Zeitschrift für Physik A*, 71:205–226, 1931. doi: 10.1007/BF01341708.

[19] L. Hulthén. Über das Austauchsproblem eines Kristalles. *Ark. Mat. Astron. Fys. A*, 26:1, 1938.

[20] Minoru Takahashi. *Thermodynamics of One-Dimensional Solvable Models*. Cambridge University Press, Cambridge, 1999.

[21] Hong-Chen Jiang, Matthew S. Block, Ryan V. Mishmash, James R. Garrison, D. N. Sheng, Olexei I. Motrunich, and Matthew P. A. Fisher. Non-Fermi liquid d-wave metal phase of strongly interacting electrons. *http://arxiv.org/abs/1207.6608*, 2012.

[22] Stefan Depenbrock, Ian P. McCulloch, and Ulrich Schollwöck. Nature of the Spin-Liquid Ground State of the $S = 1/2$ Heisenberg Model on the Kagome Lattice. *Phys. Rev. Lett.*, 109:067201, Aug 2012. doi: 10.1103/PhysRevLett.109.067201. URL `http://link.aps.org/doi/10.1103/PhysRevLett.109.067201`.

CHAPTER 9

FINITE TEMPERATURE MATRIX PRODUCT STATE ALGORITHMS AND APPLICATIONS

Abstract:[74] We review the basic theory of matrix product states (MPS) as a numerical variational ansatz for time evolution, and present two methods to simulate finite temperature systems with MPS: the ancilla method and the minimally entangled typical thermal state method. A sample calculation with the Bose-Hubbard model is provided.

9.1 Introduction

The dimension of the Hilbert space for a general many-body system increases exponentially with the system size, severely restricting the system sizes amenable to straightforward numerical study. Several techniques have been developed to deal with this fact, such as the stochastic sampling of the Hilbert space in quantum Monte Carlo techniques and the judicious use of symmetries and sparse matrix structures in exact diagonalizations. The most successful approximate method for 1d systems is the density matrix renormalization group (DMRG) method first pioneered by White [1] (see Chapter 24 for a methodology tailored towards higher-dimensional lattice configurations). Soon after, the theory of matrix product states[2, 3] (MPS) was used to shed light on the amazing success of DMRG[4, 5]. Ideas from quantum information theory, most notably the idea of bipartite entanglement, have led to the development of MPS algorithms which generalize DMRG to time evolution[6, 7], periodic boundary conditions [8], and finite temperature[9, 10]. In this chapter we

---

[74] Published previously as *Finite Temperature Matrix Product State Algorithms and Applications*, M. L. Wall and L. D. Carr, Chapter in ”Quantum Gases: Finite Temperature and Non-Equilibrium Dynamics” (Vol. 1 Cold Atoms Series), N. P. Proukakis, S. A. Gardiner, M. J. Davis and M. H. Szymanska, eds. (Imperial College Press, 2012).

review algorithms based on MPS for finite temperature simulations and discuss their relevance to studying finite temperature superfluid systems.

9.2 Methodology

9.2.1 Matrix Product States

A *matrix product state*[75] (MPS) on a lattice with periodic boundary conditions is defined as

$$|\Psi_{\mathrm{mps}}\rangle = \sum_{i_1,i_2,\dots i_L=1}^{d} \mathrm{Tr}\left(\mathbf{A}^{[1]i_1}\cdots\mathbf{A}^{[L]i_L}\right)|i_1,\cdots,i_L\rangle, \tag{9.1}$$

where the $\mathbf{A}^{[k]i_k}$ are square matrices[76] of dimension χ (the bond dimension), d is the dimension of the Hilbert space spanned by the $\{|i_k\rangle\}$, and L is the number of lattice sites. Let us refer to the set of all MPSs with bond dimension χ as $\mathcal{M}_\chi$. An MPS in $\mathcal{M}_\chi$ contains $Ld\chi^2$ parameters, and so it is clear that any state on a finite lattice can be written as an MPS provided we take the bond dimension to be $\chi_{\max} = d^{\lfloor L/2\rfloor}$. However, the great utility of MPSs is that an MPS with bond dimension $\chi \ll \chi_{\max}$ often provides an excellent approximation to the true state[12] and allows for exponentially more efficient manipulation and calculation of observables than an exact representation.

To visualize MPSs and operations with them, it is useful to introduce the notion of a *tensor network diagram* as in Figure 9.1. In such a diagram a box represents a tensor, free lines are uncontracted indices and closed lines are contracted indices. Figure 9.1(a) shows the state of a many-body system expressed in the basis of the full Hilbert space as an L-index tensor, and Figure 9.1(b) shows the same state written as an MPS. The advantage of the MPS representation becomes clear when we compute scalar products such as $\langle\psi|\hat{O}|\phi\rangle$.

---

[75] An MPS is a vector in Hilbert space. The qualifier matrix product refers to the fact that the expansion coefficients in the Fock basis are expressed as products of matrices.

[76] These matrices can be taken to have the same symmetry as the state they represent, e.g., if the state has real coefficients in some basis then the MPS matrices can be taken to be real. See [11] and references therein for more details.

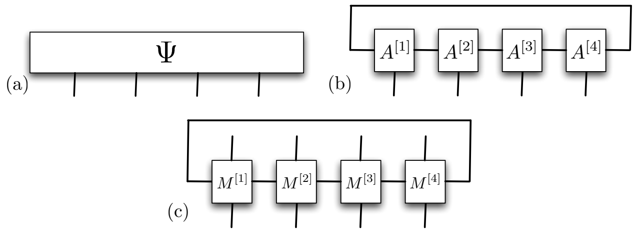


Figure 9.1: (a) Tensor network representation of full 4 site wavefunction. (b) Tensor network representation of an MPS on 4 sites. (c) Tensor network representation of an MPO on 4 sites.

Before we discuss how scalar products are efficiently computed, it is advantageous to introduce a *matrix product operator* (MPO) as

$$\hat{O} = \sum_{i_1,\ldots,i_L=1}^{d} \sum_{i'_1,\ldots,i'_L=1}^{d} \mathrm{Tr}\left(\mathbf{M}^{[1]i_1 i'_1} \cdots \mathbf{M}^{[L]i_L i'_L}\right) |i_1, \cdots, i_L\rangle\langle i'_1, \cdots, i'_L|, \tag{9.2}$$

where each of the $\mathbf{M}^{[k]i_k i'_k}$ is a matrix, the dimensions of which are bounded by a fixed number (bond dimension) χ_O. The tensor network representation of an MPO is similar to that of an MPS, but there are two uncontracted indices per tensor corresponding to the bra and ket indices; see Figure 9.1(c). Equivalently, one can think of each element of the matrix $\mathbf{M}^{[k]}$ as being operator valued, where the operator acts on the space spanned by $\{|i_k\rangle\}$.

Let us now see how to evaluate the scalar product of an operator $\hat{O}$ represented as an MPO between two states $|\psi\rangle$ and $|\phi\rangle$ represented as MPSs. Let us denote the MPO matrices of $\hat{O}$ as $\mathbf{M}$ and the MPS matrices of $|\psi\rangle$ and $|\phi\rangle$ as $\mathbf{A}$ and $\mathbf{B}$, respectively. Then, we have

$$\begin{aligned}\langle\psi|\hat{O}|\phi\rangle = &\sum_{i_1,\ldots,i_L=1}^{d} \sum_{i'_1,\ldots,i'_L=1}^{d} \mathrm{Tr}\left(\mathbf{A}^{[1]i_1\star} \cdots \mathbf{A}^{[L]i_L\star}\right) \\ &\times \mathrm{Tr}\left(\mathbf{M}^{[1]i_1 i'_1} \cdots \mathbf{M}^{[L]i_L i'_L}\right) \mathrm{Tr}\left(\mathbf{B}^{[1]i_1} \cdots \mathbf{B}^{[L]i_L}\right)\end{aligned} \tag{9.3}$$

$$\begin{aligned}
=&\mathrm{Tr}\left(\left[\sum_{i_1,i'_1=1}^{d}\mathbf{A}^{[1]i_1\star}\otimes\mathbf{M}^{[1]i_1i'_1}\otimes\mathbf{B}^{[1]i'_1}\right]\times\cdots\right.\\
&\left.\times\left[\sum_{i_L,i'_L=1}^{d}\mathbf{A}^{[L]i_L\star}\otimes\mathbf{M}^{[L]i_Li'_L}\otimes\mathbf{B}^{[L]i'_L}\right]\right)\\
\equiv&\mathrm{Tr}\left(\mathbf{E}_{\mathbf{M}}^{[1]}\left(\mathbf{A},\mathbf{B}\right)\cdots\mathbf{E}_{\mathbf{M}}^{[L]}\left(\mathbf{A},\mathbf{B}\right)\right),
\end{aligned}\tag{9.4}$$

where the last line defines the generalized transfer matrix $\mathbf{E}_{\mathbf{M}}^{[k]}(\mathbf{A},\mathbf{B})\equiv\sum_{i_k,i'_k=1}^{d}\mathbf{A}^{[k]i_k\star}\otimes\mathbf{M}^{[k]i_ki'_k}\otimes\mathbf{B}^{[k]i'_k}$, which is a $\chi^2\chi_O\times\chi^2\chi_O$ matrix. Naively we would expect that the multiplication of two transfer matrices would require $\mathcal{O}(\chi^6\chi_O^3)$ operations, but the special structure of the transfer matrices allows us to perform such a multiplication in $\mathcal{O}(\chi^5\chi_O^2d^2)$[77] as

$$\begin{aligned}
&\left[\mathbf{E}_{\mathbf{M}}^{[k]}\left(\mathbf{A},\mathbf{B}\right)\mathbf{E}_{\mathbf{M}}^{[k+1]}\left(\mathbf{A},\mathbf{B}\right)\right]_{[\alpha\gamma\beta],[\alpha'\gamma'\beta']}\\
&\quad=\sum_{i'=1}^{d}\sum_{\beta=1}^{\chi}\left(\sum_{i=1}^{d}\sum_{\gamma''=1}^{\chi_O}\left(\left[\mathbf{G}_{\mathbf{M}}^{[k]}\left(\mathbf{A},\mathbf{B}\right)\right]_{[\alpha\gamma\beta],[\alpha'\gamma''\beta']}\right)\mathbf{M}_{\gamma''\gamma'}^{[k+1]ii'}\right)\mathbf{B}_{\beta''\beta'}^{[k+1]i'},
\end{aligned}\tag{9.5}$$

where

$$\left[\mathbf{G}_{\mathbf{M}}^{[k]}\left(\mathbf{A},\mathbf{B}\right)\right]_{[\alpha\gamma\beta],[\alpha'\gamma''\beta']}\equiv\sum_{\alpha''=1}^{\chi}\left[\mathbf{E}_{\mathbf{M}}^{[k]}\left(\mathbf{A},\mathbf{B}\right)\right]_{[\alpha\gamma\beta],[\alpha''\gamma''\beta'']}\mathbf{A}_{\alpha''\alpha'}^{[k+1]i\star}.\tag{9.6}$$

Here the square brackets around indices denote a composite index in the Kronecker representation and parentheses give the order in which the contraction should be performed to ensure the best scaling. In particular, it is essential not to sum over the α'' and β'' indices simultaneously.[78] The tensor network representation of the scalar product procedure is given in Figure 9.2.

---

[77] The fact that the boundary matrices of MPSs with open boundary conditions have bond dimension 1 allows us to perform this contraction in $\mathcal{O}(\chi^3\chi_O^2d^2)$, and recent developments for periodic boundary conditions have reduced the scaling to $\mathcal{O}(\chi^3\chi_O^2d^2)$ for large systems with only a few relevant correlation lengths[13, 14].

[78] Here and throughout we use greek indices to denote bond indices and roman indices to denote physical indices.

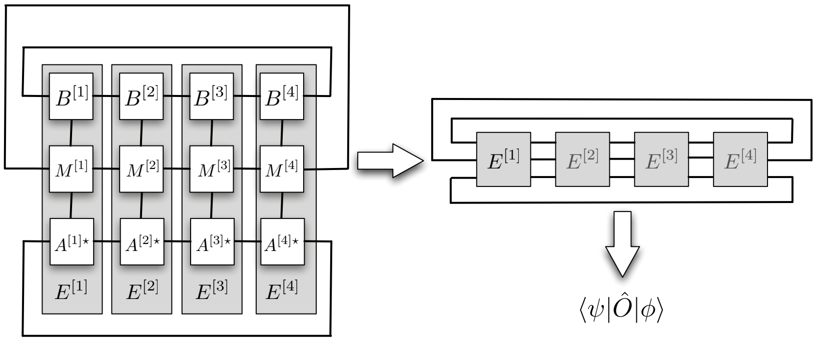


Figure 9.2: Tensor network representation of the scalar product procedure of Eq. (9.4). The transfer matrices $\mathbf{E}^{[k]}_{\mathbf{M}}(\mathbf{A},\mathbf{B})$ have been abbreviated as $E^{[k]}$ for succinctness.

Many operators of interest, such as translationally invariant 1d Hamiltonians, can be easily represented as MPOs with small bond dimension $\chi_O \sim 4-10$ [14, 15], and the MPO representations of more complex operators can be constructed using simple MPO arithmetic [15, 16]. That the MPO form of an operator is optimal for MPS algorithms can be straightforwardly deduced using the tensor network formalism, as the scalar product of an MPO between two MPSs is the most general 1d tensor network that can be efficiently contracted; see Figure 9.2.

We now turn to the simulation of time evolution using MPSs. The main difficulty of using MPSs is that $\mathcal{M}_\chi$ is not a vector space.[79] Thus, when operators such as the propagator are applied to an MPS we must find the optimal[80] projection into $\mathcal{M}_\chi$ to keep the algorithm efficient. We denote this projection as $\mathcal{P}_\chi$. The optimal MPS $|\psi\rangle \in \mathcal{M}_\chi$ representing the MPS $\hat{U}|\phi\rangle$ is

$$\begin{aligned}\mathcal{P}_\chi\left[\hat{U}|\phi\rangle\right] &= \min_{|\psi\rangle\in\mathcal{M}_\chi}\left||\psi\rangle - \hat{U}|\phi\rangle\right|^2 \\ &= \min_{|\psi\rangle\in\mathcal{M}_\chi}\left[\langle\psi|\psi\rangle + \langle\phi|\hat{U}^\dagger\hat{U}|\phi\rangle - 2\mathcal{R}\left(\langle\psi|\hat{U}|\phi\rangle\right)\right],\end{aligned} \tag{9.7}$$

---

[79] This can be seen from the fact that the addition of two MPSs is given by the direct sum of their matrices: $|\psi_C\rangle = |\psi_A\rangle + |\psi_B\rangle \Rightarrow \mathbf{C}^{[\mathrm{k}]} = \mathbf{A}^{[\mathrm{k}]} \oplus \mathbf{B}^{[\mathrm{k}]}$. If the matrices $\mathbf{A}^{[k]}$ and $\mathbf{B}^{[k]}$ have orthogonal bases then $\dim(\mathbf{C}^{[\mathrm{k}]}) = \dim(\mathbf{A}^{[\mathrm{k}]}) + \dim(\mathbf{B}^{[\mathrm{k}]})$.

[80] By optimal we mean that the overlap is maximal in the 2-norm. Although MPSs do not form a vector space, they are embedded in a larger Hilbert space and so this norm is well-defined.

where $\mathcal{R}(\bullet)$ denotes the real part. Each of the scalar products in Eq. (9.7) may be written as a quadratic form in each of the matrices $\mathbf{A}^{[k]i_k}$, as is demonstrated in the tensor network diagram Figure 9.3.

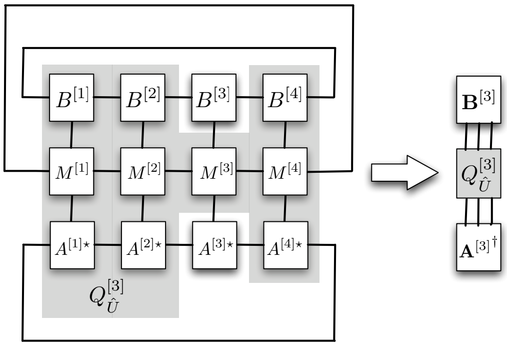


Figure 9.3: Tensor network representation of the quadratic form representing $\langle\psi|\hat{U}|\phi\rangle$ in Eq. (9.7).

Again denoting the matrices in the MPS representation of $|\psi\rangle$ by $\mathbf{A}$ and those of $|\phi\rangle$ by $\mathbf{B}$, the quadratic form of the k^{th} site may be written as

$$\mathcal{Q}^{[k]} = \mathbf{A}^{[k]^\dagger}\mathbf{Q}^{[\mathrm{k}]}_{\hat{1}}\mathbf{A}^{[\mathrm{k}]} + \mathbf{B}^{[\mathrm{k}]^\dagger}\mathbf{Q}^{[\mathrm{k}]}_{\hat{\mathrm{U}}^\dagger\hat{\mathrm{U}}}\mathbf{B}^{[\mathrm{k}]} - 2\mathcal{R}\left(\mathbf{A}^{[\mathrm{k}]^\dagger}\mathbf{Q}^{[\mathrm{k}]}_{\hat{\mathrm{U}}}\mathbf{B}^{[\mathrm{k}]}\right), \tag{9.8}$$

where $\mathbf{A}^{[k]}$ represents the $d\chi^2$ elements of the $\{\mathbf{A}^{[k]i_k}\}$, arranged as a vector, and the matrices $\mathbf{Q}_{\hat{\mathrm{O}}}$ are defined as

$$\left[\mathbf{Q}^{[\mathrm{k}]}_{\hat{\mathrm{O}}}\right]_{[\alpha i_k \alpha'][\beta i'_k \beta']} = \sum_{\gamma,\gamma'=1}^{\chi_O} \mathbf{M}^{[k]i_k i'_k}_{\gamma\gamma'} \left[\prod_{j\neq k} \mathbf{E}^{[j]}_{\mathbf{M}}\left(\mathbf{C},\mathbf{D}\right)\right]_{[\alpha\gamma\beta],[\alpha'\gamma'\beta']}, \tag{9.9}$$

where $\mathbf{C}$ and $\mathbf{D}$ are either $\mathbf{A}$ or $\mathbf{B}$ depending on the quadratic form. The $\mathbf{M}^{i_k i'_k}$ in this final expression are the matrices in the MPO representation of $\hat{O}$. The stationary points of the quadratic form Eq. (9.8) are given by the solution of the linear system[81][8]

---

[81] It is important to note that while $\mathbf{Q}_{\hat{1}}$ is the quadratic form representing the scalar product $\langle\psi|\psi\rangle$ it can not in general be made equal to the identity. The numerical conditioning of this matrix and of the linear system Eq. (9.10) can be improved by suitable choice of 'gauge conditions' on the matrices A; see[8].

$$\mathbf{Q}_{\hat{1}}\mathbf{A}^{[k]} = \mathbf{Q}_{\hat{U}}\mathbf{B}^{[k]}. \tag{9.10}$$

The algorithmic procedure for time evolution is to sweep back and forth through the lattice, solving Eq. (9.10) at each site until convergence is reached. In practice, it is essential for efficiency not to explicitly form the matrices $\mathbf{Q}_\bullet$, but rather to use iterative methods which require only multiplication by the $\mathbf{Q}_\bullet$ to solve Eq. (9.10). Details on the form of the propagator $\hat{U}$ can be found in [16, 17].

9.2.2 The Ancilla Method

At finite temperature, the state of a quantum system is given by the thermal density matrix $\hat{\rho} = e^{-\beta\hat{H}}/Z$. The ancilla method[9, 18] relies on the notion of purification[19] to represent the thermal density matrix as a pure state in an enlarged Hilbert space. Each physical site is augmented with an ancilla which has the same Hilbert space dimension as the physical site. The MPS representation of such a system is

$$|\psi\rangle = \sum_{i_1,\ldots,i_L=1}^{d}\sum_{a_1,\ldots,a_L=1}^{d} \mathrm{Tr}\left(\mathbf{A}^{[1]i_1a_1}\cdots\mathbf{A}^{[L]i_La_L}\right)|i_1a_1\cdots i_La_L\rangle. \tag{9.11}$$

One can think of the combined system as a two-legged ladder, with the physical sites on the lower leg and the ancillae on the upper leg. The purpose of the ancillae is to act as a perfect heat bath which, when traced out, provides the proper thermal density matrix for the physical system. The choice of ancilla for infinite temperature ($\beta = 0$) is simply the normalized purification of the identity

$$|\psi(0)\rangle = \frac{1}{\sqrt{d^L}}\prod_{k=1}^{L}\sum_{i_k,a_k=1}^{d}\delta_{i_ka_k}|i_ka_k\rangle, \tag{9.12}$$

which represents a product of maximally entangled site-ancilla pairs. This state has an MPS representation with bond dimension 1 generated by taking all matrices to be $\mathbf{A}^{[k]i_ka_k}_{\alpha\beta} = \delta_{\alpha,1}\delta_{\beta,1}\delta_{i_ka_k}/\sqrt{d}$. The extension to finite inverse temperature β is provided by evolving only the physical sites[82] in imaginary time up to $\beta/2$,

$$|\psi(\beta)\rangle = e^{-\beta \hat{H}/2} |\psi(0)\rangle \, . \tag{9.13}$$

This time evolution can be efficiently performed using the methods of section 9.2.1. Observables are calculated using transfer matrices as above with the additional requirement that the ancilla degrees of freedom are traced over.

The ancilla method is conceptually very simple, and becomes numerically exact for large enough bond dimension. However, because the MPS [Eq. (9.11)] must encode the information of both the system and the bath, it requires a bond dimension $\sim \chi_{\text{gs}}^2$ at low temperatures, where χ_{gs} is the bond dimension required to accurately represent the ground state. Typical values of χ_{gs} range from 50 – 5000, making the ancilla method impractical for many systems at very low temperatures.

We conclude this section by remarking that the ancilla method represents a highly idealized heat bath chosen to reproduce the exact thermal density matrix. Many of the current examples of strongly correlated many-body systems, e.g. cold atoms, are very mesoscopic and are in contact with thermal reservoirs which are also mesoscopic. A modification of the ancilla method where the perfect entanglement at infinite temperature is replaced with ancilla-ancilla and ancilla-system couplings in the Hamiltonian can be devised. Alternatively, one can directly simulate master equations by considering matrix product density operators with optimal projections based on the Hilbert–Schmidt distance[8] or matrix product decompositions of superkets with local projections[10].

9.2.3 Minimally Entangled Typical Thermal States

A new method for finite temperature MPS simulations has recently been proposed by White [20]. The idea stems from the question 'What is a typical wave function of a quantum system at finite temperature?' That is, if we are to measure a quantum

---

[82] That is, the Hamiltonian only couples physical sites to physical sites, and not ancillae to ancillae or physical sites to ancillae.

system at finite temperature, what ‘typical’ pure states would we find, and with what probabilities? It is clear from the basic tenets of statistical mechanics that any set of typical states $\{|\phi\,(i)\rangle\}$ must satisfy

$$\sum_i P\,(i)\,|\phi\,(i)\rangle\langle\phi\,(i)\,| = e^{-\beta\hat{H}}, \tag{9.14}$$

where $P(i)$ is the probability of measuring the system to be in state $|\phi(i)\rangle$, and so the expectation of an operator $\hat{A}$ at finite temperature may be written as

$$\langle\hat{A}\rangle = \sum_i \frac{P(i)}{Z}\langle\phi(i)|\hat{A}|\phi(i)\rangle, \tag{9.15}$$

with Z the partition function. From Eq. (9.15), we see that we can calculate observables using an unweighted average of $\langle\phi(i)|\hat{A}|\phi(i)\rangle$, if we choose the $|\phi(i)\rangle$ at random according to their probabilities of being measured $P(i)/Z$. It is easy to generate states satisfying the typicality condition [Eq. (9.14)] simply by taking any orthonormal basis $\{|i\rangle\}$ and defining

$$|\phi\,(i)\rangle = [P\,(i)]^{-1/2}\exp\left(-\beta\hat{H}/2\right)|i\rangle, \quad P\,(i) = \langle i|\exp\left(-\beta\hat{H}\right)|i\rangle. \tag{9.16}$$

We now use the freedom in the choice of the orthonormal basis $\{|i\rangle\}$ to generate typical states with the least amount of spatial entanglement, as these are the states which can be most efficiently represented as MPSs[6, 21]. This amounts to taking the $\{|i\rangle\}$ to be classical product states (CPSs), $|i\rangle = \prod_{k=1}^{L}|i_k\rangle$, where i_k labels the state of site k. The set of $|\phi(i)\rangle$ obtained from this choice of $\{|i\rangle\}$ are called *minimally entangled typical thermal states* (METTS).

The most efficient algorithmic procedure for generating thermal averages using METTS is as follows:

1. Choose a CPS $|i\rangle$ at random.

2. Evolve in imaginary time using the methods of section 9.2.1 to generate the METTS $|\phi(i)\rangle = [P(i)t]^{-1/2}\exp(-\beta\hat{H}/2)|i\rangle$.

3. Compute observables of interest using this METTS and add to the running averages.

4. Randomly select a new CPS $|i'\rangle$ according to the probability $|\langle i'|\phi(i)\rangle|^2$.

5. Repeat from step 2 until converged.

We see that the main loop of this algorithm closely resembles a Monte Carlo iteration with measurement taking the place of the usual configuration updates. However, it does not rely on a rejection method to perform sampling, and so each METTS that is generated can be used to generate statistics. In practice very few (~ 100) METTS suffice to obtain the total energy to a relative accuracy of 10^{-5}. For algorithmic details on how to perform the CPS selection to minimize correlations between successive METTS we refer the reader to [16].

This METTS algorithm has many advantages over the ancilla method of the previous section. As we do not have to encode the bath degrees of freedom in our MPS, the bond dimension required to accurately represent each METTS ranges from 1 at infinite temperature to χ_{gs} at very low temperatures. This makes the METTS method more efficient than the ancilla method by a factor of $10^3 - 10^{10}$ for typical systems at very low temperatures. Additionally, if the Hamiltonian of interest has a global symmetry then we can use the fact that the MPS matrices must transform irreducibly to speed up the calculation[15] or find the thermal ensemble corresponding to a fixed quantum number (canonical ensemble). This latter point is relevant to cold atom systems where the total number of atoms is held fixed.[83]

---

[83] The ancilla method can also be used to simulate systems in the canonical ensemble, but the process is complicated by the fact that we need the purification of the constrained infinite temperature density matrix. This purification can be generated using a ground state DMRG-type calculation with a suitably chosen Hamiltonian[22]. The Hamiltonian will contain artificial ancilla-ancilla and ancilla-physical site couplings which are typically highly nonlocal.

9.3 Validity Issues

It has been shown that MPSs can faithfully represent ground states of 1d gapped Hamiltonians with at most nearest neighbor interactions with a bond dimension which grows only polynomially in the system size[12]. In higher dimensions this polynomial scaling gives way to an exponential scaling[23], but calculations on 2D systems of width 8 – 12 are still feasible[24]. Generalizations of MPSs to higher dimensions exist, but are so far limited by poor polynomial scaling of tensor contractions[25–27]. Perhaps the most important quality of MPS methods as compared to other efficient many body methods, such as quantum Monte Carlo, is that MPS methods work equally well for fermionic or frustrated systems. All of the methods presented here will work equally well for any 1d or quasi-1d physical system.

9.4 Application: Specific Heat of the Hard-Core Extended Bose-Hubbard Model

As an example of how the above methods may be applied to study the behavior of a finite temperature superfluid system, we study the properties of the hard-core extended Bose-Hubbard model

$$\hat{H} = -J \sum_{\langle i,j \rangle} \left(\hat{b}_i^{\dagger} \hat{b}_j + \text{H.c.} \right) + V \sum_{\langle i,j \rangle} \hat{n}_i \hat{n}_j \tag{9.17}$$

at half filling. This model is known to have a superfluid phase in the XY universality class for $V < 2J$. In the below figure we show a typical thermodynamic quantity, the specific heat $C_V = \beta^2(\langle \hat{H}^2 \rangle - \langle \hat{H} \rangle^2)/L$, as a function of temperature and the nearest-neighbor repulsion. Note that computation of $\langle \hat{H}^2 \rangle$ is easily performed when the MPO representation of $\hat{H}$ is known.

9.5 Acknowledgments

We acknowledge useful discussions with Juan José García Ripoll and Miles Stoudenmire. This work was supported by the National Science Foundation under Grant

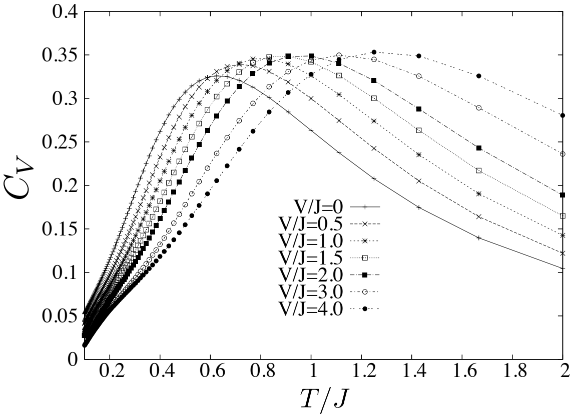


Figure 9.4: Specific heat of the hard-core extended Bose-Hubbard model on 34 sites for repulsive nearest neighbor interaction $V = 0, 0.5, 1.0, 1.5, 2.0, 3.0, 4.0$. The qualitative behavior of the low temperature specific heat changes as V becomes larger than $2J$ because the system transitions from a gapless superfluid phase into a gapped insulating phase.

PHY-0903457. MLW thanks the Boulder Summer School for Condensed Matter for stimulating discussions. We also acknowledge the Golden Energy Computing Organization at the Colorado School of Mines for the use of resources acquired with financial assistance from the National Science Foundation and the National Renewable Energy Laboratories.

9.6 References Cited

[1] Steven R. White. Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.*, 69:2863, 1992.

[2] A Klumper, A Schadschneider, and J Zittartz. Equivalence and solution of anisotropic spin-1 models and generalized t-J fermion models in one dimension. *J. Phys. A*, 24:L955, 1991.

[3] M. Fannes, B. Nachtergaele, and R. F. Werner. Finitely correlated states on quantum spin chains. *Commun. Math. Phys.*, 144:443, 1992.

[4] Stellan Östlund and Stefan Rommer. Thermodynamic limit of density matrix renormalization. *Phys. Rev. Lett.*, 75:3537, 1995.

[5] Stefan Rommer and Stellan Östlund. Class of ansatz wave functions for one-dimensional spin systems and their relation to the density matrix renormalization group. *Phys. Rev. B*, 55:2164, 1997.

[6] Guifré Vidal. Efficient Simulation of One-Dimensional Quantum Many-Body Systems. *Phys. Rev. Lett.*, 93:040502, 2004.

[7] A J Daley, C Kollath, U Schollwöck, and G Vidal. Time-dependent density-matrix renormalization-group using adaptive effective Hilbert spaces. *J. Stat. Mech.*, 2004:P04005, 2004.

[8] F. 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.

[9] F. Verstraete, J. J. García-Ripoll, and J. I. Cirac. Matrix Product Density Operators: Simulation of Finite-Temperature and Dissipative Systems. *Phys. Rev. Lett.*, 93:207204, 2004.

[10] Michael Zwolak and Guifré Vidal. Mixed-State Dynamics in One-Dimensional Quantum Lattice Systems: A Time-Dependent Superoperator Renormalization Algorithm. *Phys. Rev. Lett.*, 93:207205, 2004.

[11] D. Perez-Garcia, F. Verstraete, M. M. WolF, and J. I. Cirac. Matrix Product State Representations. *Quantum Inf. Comput.*, 7(401), 2007.

[12] F. Verstraete and J. I. Cirac. Matrix product states represent ground states faithfully. *Phys. Rev. B*, 73:094423, 2006.

[13] Peter Pippan, Steven R. White, and Hans Gerd Evertz. Efficient matrix-product state method for periodic boundary conditions. *Phys. Rev. B*, 81:081103, 2010.

[14] B Pirvu, V Murg, J I Cirac, and F Verstraete. Matrix product operator representations. *New J. Phys.*, 12:025012, 2010.

[15] Ian P McCulloch. From density-matrix renormalization group to matrix product states. *J. Stat. Mech.*, 2007:P10014, 2007.

[16] E M Stoudenmire and Steven R White. Minimally entangled typical thermal state algorithms. *New J. Phys.*, 12:055026, 2010.

[17] Juan Josè García-Ripoll. Time evolution of Matrix Product States. *New J. Phys.*, 8:305, 2006.

[18] Adrian E. Feiguin and Steven R. White. Finite-temperature density matrix renormalization using an enlarged Hilbert space. *Phys. Rev. B*, 72:220401, 2005.

[19] M. Nielsen and I. Chuang. *Quantum Computation and Quantum Information.* Cambridge University Press, Cambridge, 2000.

[20] Steven R. White. Minimally Entangled Typical Quantum States at Finite Temperature. *Phys. Rev. Lett.*, 102:190601, 2009.

[21] Guifré Vidal. Efficient Classical Simulation of Slightly Entangled Quantum Computations. *Phys. Rev. Lett.*, 91:147902, 2003.

[22] Adrian E. Feiguin and Gregory A. Fiete. Spectral properties of a spin-incoherent Luttinger liquid. *Phys. Rev. B*, 81:075108, 2010.

[23] Shoudan Liang and Hanbin Pang. Approximate diagonalization using the density matrix renormalization-group method: A two-dimensional-systems perspective. *Phys. Rev. B*, 49:9214, 1994.

[24] Steven R. White and D. J. Scalapino. Pairing on striped $t-t'-J$ lattices. *Phys. Rev. B*, 79:220504, 2009.

[25] F. Verstraete and J. I. Cirac. Renormalization algorithms for Quantum Many-Body systems in two and higher dimensions. *arXiv:cond-mat/0407066v1*, 2004.

[26] Norbert Schuch, Michael M. Wolf, Frank Verstraete, and J. Ignacio Cirac. Simulation of Quantum Many-Body Systems with Strings of Operators and Monte Carlo Tensor Contractions. *Phys. Rev. Lett.*, 100:040501, 2008.

[27] G. Evenbly and G. Vidal. Entanglement Renormalization in Two Spatial Dimensions. *Phys. Rev. Lett.*, 102:180406, 2009.

PART V

OPEN SOURCE CODE AND EDUCATIONAL MATERIALS

CHAPTER 10

OPEN SOURCE CODE DEVELOPMENT

The purpose of this chapter is to describe two open source code projects in which the author has been involved. The first, open source time-evolving block decimation (OSTEBD) [1], is a package which was maintained by the Carr theoretical physics research group and heavily modified for flexibility and efficiency by the author. OSTEBD had two releases, one in February of 2009 and the other in October of 2009. The OSTEBD dedicated blog has been viewed uniquely over 2,100 times to date, and OSTEBD has been used in 8 publications within the Carr research group [2–9] and 6 known publications outside of this group [10–15]. The most recent blog statistics, as gathered by http://wordpress.com, are shown in Figure 10.1. An extensive user's guide was distributed with OSTEBD describing the background of time-evolving block decimation (TEBD), providing exercises to acquaint the user with the package, and manual pages for all routines in the code. The most recent version of the manual is included in Appendix B, and the most recent release of OSTEBD, v2.0, is contained on the source code CD accompanying this thesis.

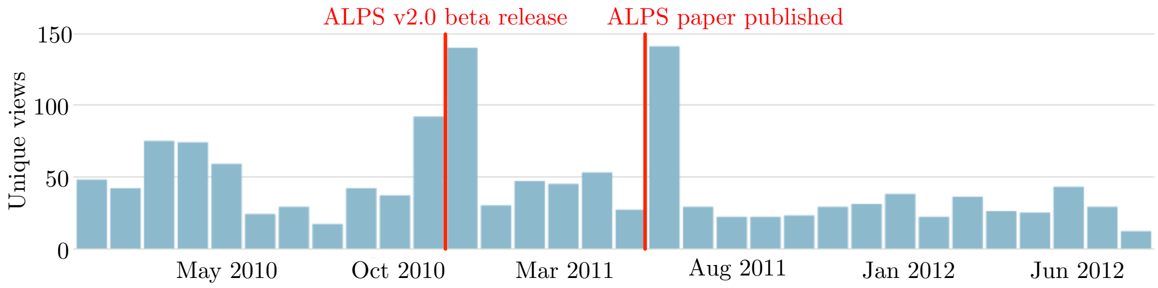


Figure 10.1: Open source time-evolving block decimation blog statistics. The red lines indicate releases of ALPS code and documentation.

In 2010, the code used for the OSTEBD project was significantly altered for stability and speed and merged into the algorithms and libraries for physics simulations

(ALPS) package [16], the premier resource for numerical methods for strongly correlated many-body problems. This code was formally included in the v2.0 release of the ALPS package in November 2010. ALPS contains open source implementations of nearly all widely used numerical methods for strongly correlated many-body systems, including exact diagonalization; worm, stochastic series expansion, and quantum Wang-Landau quantum Monte Carlo; dynamical mean field theory; and a static density-matrix renormalization group code. The TEBD software described in this thesis was the first component of the ALPS package to enable the study of dynamics. The ALPS software package is maintained by the ALPS international collaboration presently consisting of 28 researchers. In addition to being used for cutting-edge research world-wide with nearly 250 citations for the first two versions of ALPS at the time of writing [8, 17], ALPS is also used as a pedagogical tool for summer schools. The ALPS collaboration maintains other pedagogical resources such as tutorials and a user's forum. The most recent major release of ALPS at the time of writing of this thesis, ALPS 2.1, is included with the source code CD. Additionally, all documentation for the ALPS TEBD routines which were written solely by the present author, including tutorial exercises, are reprinted in Appendix C.

The OSTEBD manual (see Sec. B) covers the vast majority of that project. Hence, in Sec. 10.1 we provide a brief overview of the package and its capabilities, referring the reader to the manual for more detail. Sec. 10.1.1 contains information about parallel extensions to OSTEBD which were in alpha version at the time of the v2.0 release. The documentation for the ALPS code is given in Appendix C. It is much less detailed than the documentation for the OSTEBD project. Hence, in Sec. 10.2 we discuss the differences in the ALPS and OSTEBD codes, as well as other parts of the ALPS project which required development by the present author including a Python [18] front end and integration with the VisTrails workflow provenance system [19].

10.1 Open Source Time-Evolving Block Decimation Overview.

OSTEBD [1] is a software package, written in Fortran 90, implementing the TEBD algorithm for one-dimensional systems with nearest-neighbor interactions [20, 21]. In addition to real-time propagation, ground states may be found using imaginary time propagation. OSTEBD supports systems of bosons with or without spins, fermions with or without spins, spin systems, and open or periodic boundary conditions [22]. The resulting matrix product state (MPS) representations of the wavefunction are explicitly stored, allowing for checkpointing of long simulations as well as wavefunction-based techniques for detecting quantum phase transitions such as fidelity analysis [6]. Arbitrary one and two-point correlation functions can be specified by the user, and routines to obtain one and two-body density matrices are included for more complex or quantum-information based measures. Number conservation for particle models or magnetization conservation for spin models is supported [23, 24]. The irreps of the associated on-site Hilbert space are allowed to be degenerate. Both data parallelism and intrinsic parallelism are included in alpha version with the 2.0 release of the code, and are discussed in Sec. 10.1.1. OSTEBD v2.0 contains $21,822$ lines of code in its core, i.e., code which does not comprise main files or inputs.

Inputs are provided using the Fortran NAMELIST syntax, which allows for input files to be generated using scripts. For example, the inputs for the `BoseHubbard_ITP.f90` case study are given in the file `BH_ITP.nml` as

```
&SystemSettings
 systemSize=30, maxFilling=5, totNum=30, BoundaryCond='O',
     TrotterOrder=5
&end

&BHParams
 jTunn=1.0, U0=20.0, V0=0.0, mu0=0.0
&end

&ITPParams
```

```
  chiMin=15, chiMax=20, convCriterion1=0.00001, convCriterion2
     =0.0000001, stepsForJudge=100, dtITP=0.05, maxITPsteps=4000,
     itpDir='ITPDATA/'
&end
```

The top line gives Hilbert space metadata, including the number of lattice sites, the maximum number of particles allowed on any site, the number of particles, the boundary conditions, and the order of Trotter expansion used for the propagator, see Appendix B.3.5. The second line gives the parameters of the Bose-Hubbard model, Eq. (B.151). Finally, the last line gives convergence criteria for imaginary time propagation, such as the maximum number of iterations, the imaginary time step, and the maximal bond dimensions. In addition to supporting a wide array of models, the package was designed to be easily modified to suit users' individual needs. The comprehensive manual pages starting with Appendix B.7 of the OSTEBD manual were part of this approach, as were exercises using the OSTEBD routines in Appendix B.4 which require users to write their own code. For more detail on the educational component of the manual, we refer the reader to Chapter 11.

10.1.1 Parallel Extensions in Open Source Time-Evolving Block Decimation

Data-parallelism is supported in alpha version in OSTEBD v2.0 via the `PD_Extension`. Here a phase diagram is defined by a range of chemical potentials $\mu_{\mathrm{min}} \leq \mu \leq \mu_{\mathrm{max}}$ with μ_{res} points or total numbers $N_{\mathrm{min}} \leq N \leq N_{\mathrm{max}}$ with N_{res} points and a range of tunnelings $J_{\mathrm{min}} \leq J \leq J_{\mathrm{max}}$ with J_{res} points. A master node sends points in this phase diagram to worker nodes one at a time, and the worker nodes send observables to the master node when the computation has finished. This automatically scheduled data-parallel paradigm is more efficient than assigning the tasks evenly between all nodes, as points in more highly entangled regions of the phase diagram take more time to compute than less entangled regions, and so nodes which are given only lowly entangled regions may sit idle while others compute highly entangled regions. The code in this extension was used in Ref. [6]. For large numbers of tasks, i.e., large $\mu_{\mathrm{res}} J_{\mathrm{res}}$ or $N_{\mathrm{res}} J_{\mathrm{res}}$, the code is $P-1$ times faster than a single core within a few percent, where P is the number of processors.

`Sites_Parallel_Extension`, also included in alpha version with OSTEBD, is an intrinsically parallel TEBD code using MPI [25]. In the sense of balancing the load evenly and having large granularity, the most naturally parallelizable portion of the TEBD algorithm is the loop over spatial positions during time evolution. The ability to intrinsically parallelize the time evolution sweep in TEBD stems from the fact that in the Vidal canonical form Eq, (6.39), any bipartite splitting can be chosen to be the orthogonality center. Hence, when we use Vidal's suggested Trotterization of the propagator:

$$\begin{aligned} e^{-i\hat{H}\delta t} &= e^{-i\hat{H}_{\mathrm{odd}}\delta t/2\hbar} e^{-i\hat{H}_{\mathrm{even}}\delta t/\hbar} e^{-i\hat{H}_{\mathrm{odd}}\delta t/2\hbar} + \mathcal{O}\left(\delta t^3\right)\,, \\ e^{-i\hat{H}_{\mathrm{odd}}\delta t/2\hbar} &= \prod_{\mathrm{odd}\ l} e^{-i\hat{H}_l \delta t/2\hbar}\,, \\ e^{-i\hat{H}_{\mathrm{even}}\delta t/\hbar} &= \prod_{\mathrm{even}\ l} e^{-i\hat{H}_l \delta t/\hbar}\,, \end{aligned} \tag{10.1}$$

we can apply the propagation over even bonds and odd bonds simultaneously at all bipartite splittings. A slab decomposition of the application of propagators across all similar parity bonds would ideally reduce the scaling of the most expensive step by a factor of P, where P is the number of processors.

A two site operation performed on the two sites l and $l+1$ involves $\lambda^{[l]}$, $\Gamma^{[l]}$, $\lambda^{[l+1]}$, $\Gamma^{[l+1]}$, and $\lambda^{[l+2]}$, see Appendix B.3.4.2. Thus, for a given processor to time evolve site l it needs to also own[84] the local tensors of site $l+1$. This means that, in a distributed memory paradigm, the last site that can be time evolved by a given processor is the penultimate one. In order for all processors to possess a current[85] copy of all owned local tensors, they must receive at least one and possibly two sets of

---

[84] In this section we use the word own to denote that a processor holds this object in its memory.

[85] By current, we mean time-evolved to the most recent time.

local tensors from neighboring sites. Thus, each processor overlaps one Γ tensor and two λ tensors with each of its neighbors. To evenly divide the memory load among all processors, we give each processor the same number of sites to optimize. When the number of processors does not evenly divide the number of sites, we give the first and last processors more sites, as the sites at the end of the chain are more weakly entangled and hence we expect to have better load balancing in this configuration. In addition to the number of Γ tensors, which we denote in the code as `my_local_dim`, we also give each processor a two-component array `my_bounds` which contains the absolute site indices of the first and last Γ tensor that the processor owns.

Initialization of the tensors in the absence of conserved quantities may be done without communication. Similarly, the Hamiltonian, propagators, and all other operators may be initialized independently. In the presence of conserved quantities, the `LabelLeft` and `LabelRight` structures depend on the state of the wavefunction to the left of the given site. Hence, we initialize the state sequentially in processor rank, with each processor sending its penultimate value of cumulative conserved quantity to the next processor.

The parity of the local site indices may not be the same as the parity of the global site indices. That is to say, the first site a processor owns might be the sixth site in the actual chain, and so one must be careful when applying propagators over even or odd bonds to apply them to globally even or globally odd sites. Because updating a local tensor at site l requires owning the local tensors of site $l+1$ and the processors do not own all of the local tensors, after each Trotter sweep we must pass local tensors between processors in order that all processors are properly updated. If the last site that a given processor π updates is one less than the total number of sites that the processor owns, then the next processor will not have updated its first tensor in the same Trotter sweep due to the parity difference. Thus, processor π should send its last updated local tensors to processor $(\pi+1)$, where they will become the first local tensors. In a similar manner, if processor π begins on a site with the same parity as the current Trotter sweep,[86] it should send its first local tensors to processor $(\pi - 1)$ where they will become the new last local tensors. Identical analysis applies to the even time step, which completes a Trotter sweep.

We now pause to consider the asymptotic scaling of our algorithm's computation time and communication time. The computational scaling of TEBD is $\mathcal{O}\left[L\frac{t_f}{\delta t}\max\left(d^3\chi^3, d^4\chi^2\right)\right]$, where χ is the bond dimension, d the local dimension, δt the infinitesimal time step, L the number of sites, and t_f the final time. The communication time is proportional to the size of the sent data, which is $\mathcal{O}\left(d\chi^2 + \chi\right)$ per time step. In essentially all cases of interest, $\chi \gg d$, and so we will assume that the $\mathcal{O}\left(\chi^3\right)$ scaling dominates. If we now divide the computation time among P processors and have all P processors communicate, we have a computational scaling of $\mathcal{O}\left[\frac{L}{P}\frac{t_f}{\delta t}d^3\chi^3\right]$ and a communication scaling of $\mathcal{O}\left[\frac{t_f}{\delta t}d\chi^2\right]$. We thus expect the best performance when $L \gg P$, $\chi \gg P$, or both, as the computation time is larger than the communication time in these instances. Additionally, parallelization is more efficient for increasing d. The $L \gg P$ condition states that each processor should update several local tensors before communicating, and the conditions involving χ and d state that it is less computationally intensive to send a copy of a tensor than to perform a tensor operation such as contraction or decomposition. For large L, the first condition is naturally enforced by the condition $P \leq \lfloor \frac{L}{2} \rfloor$, for if P is larger than this some processors will own less than two local tensors and so two site operations cannot be performed.

Not all of the expensive operations in TEBD can be parallelized using the above decomposition. The two most expensive non-parallel operations are restoration of canonical form and the computation of two-point correlation functions. The restoration of canonical form must be done in a sweeping fashion for neighboring sites. If

---

[86] Equivalently, processor $(\pi - 1)$ ends with parity opposite to the Trotter sweep.

canonical form is not restored after every time step but only before measuring observables, then this does not slow down the execution considerably. However, not using a strict canonical form at every time step degrades the quality of a TEBD simulation, as will be discussed in Sec. 10.2.1. Two-point correlation functions $\langle \hat{O}_i \hat{O}_j \rangle$ require communication of a partial overlap between processors when i and j are not both owned by the same processor. However, the amount of information which must be sent is less than in a typical Trotter sweep, and so these operations do not slow down parallel execution considerably.

We quantify the performance of our parallel implementation by three quantities: the speedup, defined as

$$S_P \equiv \frac{T_1}{T_P}, \tag{10.2}$$

where T_P denotes the time it takes the code to run on P processors, the efficiency

$$E_P \equiv \frac{T_1}{PT_P}, \tag{10.3}$$

and the experimentally determined serial fraction

$$\exp_f = \frac{\frac{1}{S_P} - \frac{1}{P}}{1 - \frac{1}{P}}. \tag{10.4}$$

The speedup is said to be ideal if $S_P \approx P$. Likewise, optimal efficiency is $E = 1$. The experimentally determined serial fraction, in an ideal case, should be as small as possible, as its inverse limits the speedup for fixed data as the number of processors increases (by Amdahl's law [26]).

We show these measures for a variety of system sizes L and bond dimensions χ in Figure 10.2. As expected from the above scaling arguments, simulations with larger L and larger χ perform better in all three measures.

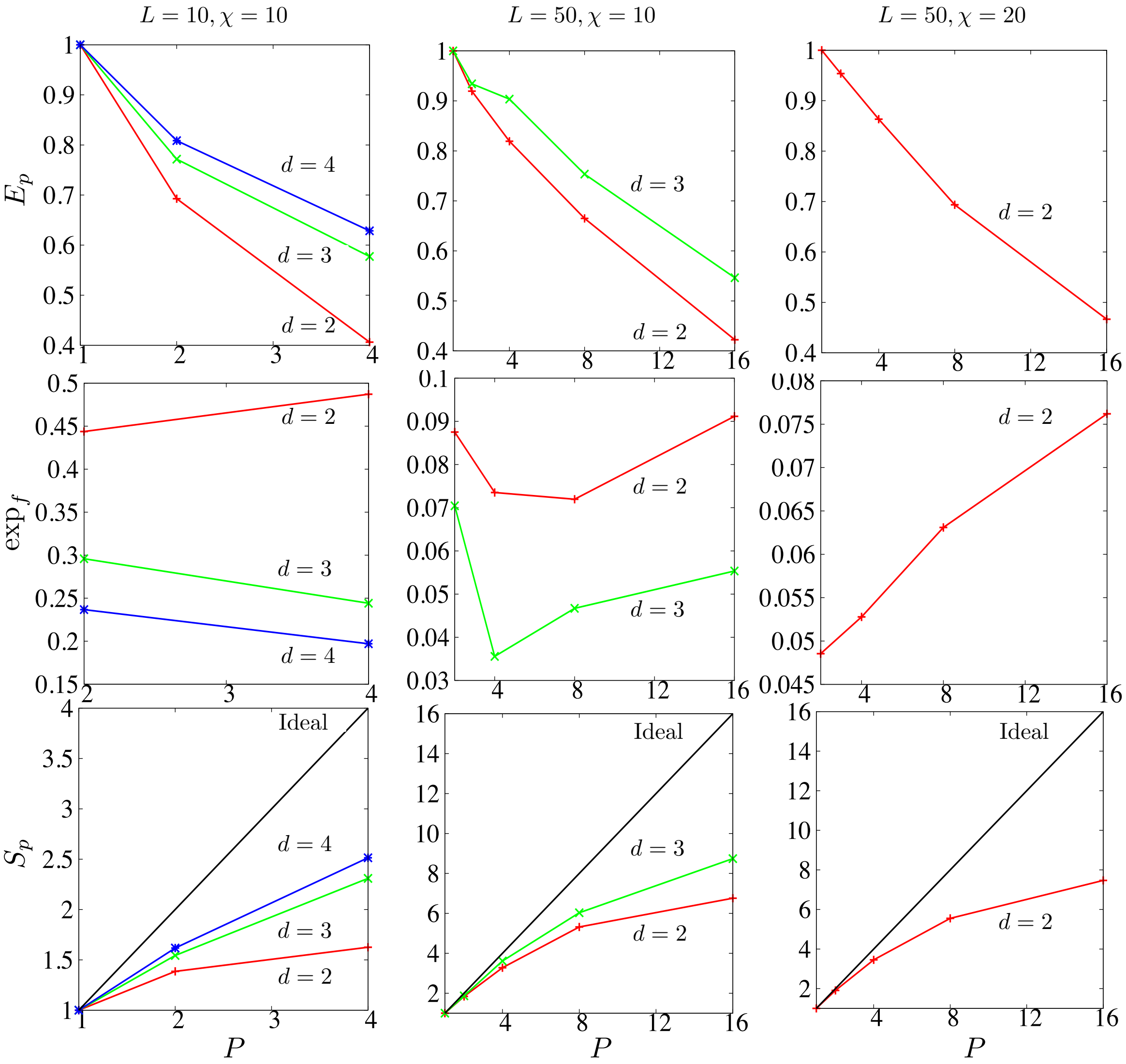


Figure 10.2: Performance of intrinsically parallel OSTEBD routines. The parallel performance as determined by all three measures increases as L, χ, and d increase, in accordance with the scaling expectations given in the text.

10.2 The Algorithms and Libraries for Physics Simulations Time-Evolving Block Decimation Routines.

The ALPS TEBD routines perform similar functions to the OSTEBD routines, performing real or imaginary time evolution of systems of bosons, fermions, or spins. Rather than using different main files or the NAMELIST system of input as in OSTEBD, the ALPS routines instead rely on a Python interface which writes appropriate input files, calls the Fortran routines, and the post-processes the output from the Fortran routines. The parts of this front end which are relevant to TEBD are covered in Sec. 10.2.2. The ALPS code is less easily modified to suit user needs than the OSTEBD code due to the requirement of compatibility with the front end, but the author attempted to counteract this by including a wide array of models, measures, and time evolution protocols accessible by the Python interface. The Python interface for the general MPS routines discussed in Appendix A increases the capacity for models and measures enormously, and will be included in a later release of ALPS. In addition to the different interface, several improvements of the core routines of OSTEBD regarding speed and efficiency were implemented, and are discussed in Sec. 10.2.1. The ALPS TEBD routines can also be utilized via the integration of ALPS with the VisTrails workflow provenance system in part due to Python ports written by the author. Discussions of scientific workflows, provenance, and the ALPS VisTrails interface are relegated to Sec. 10.2.3.

The ALPS TEBD code consists of 10,610 lines of Fortran and on the order of 700 lines of Python written by the author, not including tutorials and other main files.

10.2.1 Numerical Optimizations of the Algorithms and Libraries for Physics Simulations Code

In two-site operations with TEBD, it is always assumed that the MPS is in the Vidal canonical form, in particular that the Schmidt vectors $\left\{|\phi_{\alpha}^{[1\dots l-1]}\rangle\right\}$ and $\left\{|\phi_{\beta}^{[l+2\dots L]}\rangle\right\}$ are orthonormal bases. When this is true, then the tensors on which

TEBD operates contain the orthogonality center, and any truncation which occurs represents an optimal truncation of the wavefunction in that it minimizes the 2-norm distance between the truncated state and the true state at that bipartite splitting. When these bases are not orthogonal, the truncation represents an optimal truncation of the particular tensor, but there is not generally any relation between the tensor and the actual wavefunction. After a two-site operation, the Schmidt vectors $\left\{|\phi_{\gamma}^{[1\dots l]}\rangle\right\}$ and $\left\{|\phi_{\gamma}^{[l+1\dots L]}\rangle\right\}$ will be orthogonal by construction, but the same cannot be said of other sets of vectors which change under the course of the operation, for example the set $\left\{|\phi_{\gamma}^{[1\dots l+1]}\rangle\right\}$. This destruction of canonical form always arises from non-unitary operations such as application of an imaginary time propagator, but it also arises from truncation of the bond dimension of a state following application of a unitary operator. The way that OSTEBD accounted for these errors was to explicitly return the state to its canonical form periodically during real and imaginary time evolution. Additionally, in the case that the time step is small, imaginary time propagation is close to unitary, and when truncation is also small the effects of being away from canonical form are not extreme and can be accounted for by extrapolating the time step to zero [27].

A better way to account for the loss of canonical form is to re-order the application of the propagator such that all of the bases which we assume to be orthonormal throughout the course of an update sweep have been made orthonormal by the last operation on them. This can be done by applying operations to successive sites in a directed sweeping motion, e.g. $(l, l+1)$, $(l+1, l+2)$, etc. or $(l+1, l+2)$, $(l, l+1)$, etc.. Before computing observables, in which all sites are assumed to be in canonical form, we can return the state to a fully canonical representation by applying the identity operator to two sites at a time in the same forwards and backwards sweeping motion.

In order to apply the propagators in the sweeping fashion as just described, we abandon the use of the Trotter expansion discussed in Appendices B.3.5.1 and B.3.5.2, and instead use the expansion [28]

$$\exp\left(-i\hat{H}\delta t/\hbar\right)=\prod_{i=1}^{L-1}\exp\left(-i\hat{H}_{i,i+1}\delta t/2\hbar\right)\prod_{i=L-1}^{1}\exp\left(-i\hat{H}_{i,i+1}\delta t/2\hbar\right)+\mathcal{O}\left(\delta t^3\right)\,, \tag{10.5}$$

where $\hat{H}_{i,i+1}$ is the two-site Hamiltonian acting on sites i and $i+1$. The use of this decomposition to explicitly maintain local canonical form is the first of the optimizations implemented in the ALPS code for stability.

Another source of possible numerical error in TEBD as implemented in the OSTEBD package is in the decomposition of the object $\Theta^{ij}_{\alpha\beta}$ representing the two-site wavefunction following application of some operation into the Γ tensors and λ tensors as[87]

$$\Theta_{(\alpha i)(j\beta)}\overrightarrow{\text{SVD}}\,USV\,, \tag{10.6}$$

$$\tilde{\lambda}^{[l+1]}_{\alpha_l}=\frac{S_{\alpha_l}}{\sqrt{\sum_{\alpha=1}^{\chi}\left(S_\alpha\right)^2}}\,, \tag{10.7}$$

$$\tilde{\Gamma}^{[l]i_l}_{\alpha_{l-1}\alpha_l}=U_{(i_l-1)\chi+\alpha_{l-1},\alpha_l}/\lambda^{[l]}_{\alpha_{l-1}}\,, \tag{10.8}$$

$$\tilde{\Gamma}^{[l+1]i_{l+1}}_{\alpha_l\alpha_{l+1}}=V_{\alpha_l,(i_{l+1}-1)\chi+\alpha_{l+1}}/\lambda^{[l+2]}_{\alpha_{l+1}}\,. \tag{10.9}$$

Here $\overrightarrow{\text{SVD}}$ indicates performing a singular value decomposition on the left hand side to yield the right hand side. In particular, the elements of the tensor λ appearing in Eq. (10.8) and Eq. (10.9) can be very small, of order the machine precision, resulting in loss of precision in the elements of the Γ tensors. To remedy this, we do not store the tensors Γ explicitly, but rather only store products such as $\lambda\Gamma$ or $\Gamma\lambda$ to create left- and right-canonical MPS tensors according to the translation rules Eq. (6.43) and (6.44). If a single λ tensor is left uncontracted from the remainder of the tensors

---

[87] See also Eqs. (B.80)-(B.83) of the OSTEBD manual.

of the MPS at any time, the resulting MPS decomposition is in mixed canonical form rather than the Vidal canonical form,[88] with the uncontracted λ tensor being the orthogonality center. The splitting of Θ now becomes

$$\Theta_{(\alpha i)(j\beta)} \overrightarrow{\text{SVD}}\, USV\,, \tag{10.10}$$

$$\tilde{\lambda}^{[l+1]}_{\alpha_l} = \frac{S_{\alpha_l}}{\sqrt{\sum_{\alpha=1}^{\chi}\left(S_\alpha\right)^2}}\,, \tag{10.11}$$

$$(\lambda\Gamma)^{[l]i_l}_{\alpha_{l-1}\alpha_l} = U_{(i_l-1)\chi+\alpha_{l-1},\alpha_l}\,, \tag{10.12}$$

$$(\Gamma\lambda)^{[l+1]i_{l+1}}_{\alpha_l\alpha_{l+1}} = V_{\alpha_l,(i_{l+1}-1)\chi+\alpha_{l+1}}\,. \tag{10.13}$$

The bond-centered orthogonality center can now be shifted in the direction of the propagator sweep as discussed in Sec. 6.3. it is remarkable that the mixed canonical form arises naturally when repairing numerical instabilities that arise from the more theoretically appealing Vidal canonical form. This demonstrates the general fact that the theoretically most appealing methods are often not the most numerically stable in practice.

A third optimization is an improved implementation of symmetry-adapted MPSs (see Sec. 6.6) over OSTEBD in the case that all of the on-site irreps are non-degenerate. In this case, the Γ tensors may be written as

$$\Gamma^{i}_{\alpha\beta} = \Gamma_{\alpha\beta}\delta_{i,q_\beta-q_\alpha+1-q_{i,\min}}\,, \tag{10.14}$$

where q_i is the quantum number on site i, q_γ is the cumulative quantum number to the left of bond γ, and $q_{i,\min}$ is the minimum allowed value of q_i. Thus, we can store only the matrices $\Gamma_{\alpha\beta}$ rather than the full tensors. In addition to a reduction in storage requirements, this also avoids sums over the local dimension which leads to significant speedup.

---

[88] See Sec. 6.3 for a discussion of canonical forms for MPSs.

Fourth, OpenMP threading [29] over computationally expensive portions of the code is implemented. In particular, the greatest speedup occurs by threading over the SVD of Θ when Abelian symmetries are used. Here the threading is over blocks corresponding to a fixed total charge to the left of the bond on which Θ is centered. This threading displays nearly perfect parallel efficiency because threads operate on blocks independently. Threading is also performed over the formation of Γs from the SVD matrices, the formation of Θ, and propagation of the partial overlap G used for two-point correlation functions, see Appendix B.3.6.4 of the OSTEBD manual.

Fifth, in OSTEBD the bond dimension is used as the main truncation parameter, and is fixed by the user without any input from the simulation. A better means to control the entanglement cutoff is to define a tolerance ϵ on the percentage of the singular value norm which can be discarded at a particular two-site operation, and then let this tolerance dynamically define the bond dimension. That is, χ is taken to be the smallest integer satisfying

$$1-\frac{\sum_{\alpha=1}^{\chi} S_{\alpha}^{2}}{\sum_{\alpha} S_{\alpha}^{2}} \leq \epsilon\,. \tag{10.15}$$

Here, the sum in the denominator runs over all the singular values. The parameter ϵ is represented by `TRUNC_LIMIT` and a safeguard bond dimension is provided by `CHI_LIMIT`. The parameter `ITP_CHIS` is also able to be specified separately for imaginary time propagation. The value of χ now represents a safeguard value to avoid running out of memory, and may be set very large. Provided that this safeguard value of χ is not reached, the simulation remains quasi-exact in the sense that the distance between the state returned by TEBD and the true time evolution obtained from the given initial state is bounded by a known constant.

Finally, it is described in the OSTEBD manual how computing expectations of nonlocal observables expressed as tensor products can be performed more efficiently than a general two-site operator due to the tensor network structure of the contraction. The one place where this was not taken advantage of in OSTEBD was in computing the energy, as the expansions of the two-site Hamiltonians in terms of a separable basis are not generally known. The ALPS routines account for the non-separability of the Hamiltonian $\hat{H}$ which acts on two neighboring sites $|i\rangle|j\rangle$ via the following transformation

$$\left[\hat{H}\right]_{(ij),(i'j')} = \left[\hat{H}_{i,i+1}\right]_{(ii'),(j,j')} , \tag{10.16}$$

$$\left[\hat{H}\right]_{(ii'),(j,j')} \xrightarrow{\text{SVD}} \sum_{\gamma} U_{(i,i')\gamma}\Sigma_{\gamma}V_{\gamma(j,j')} , \tag{10.17}$$

which implies that we may write the operator as a sum of separable operators as

$$\Rightarrow \left[\hat{H}\right]_{(ij),(i'j')} = \sum_{\gamma} H^{L}_{(i,i')\gamma}H^{R}_{\gamma,(j,j')} , \tag{10.18}$$

where

$$H^{L}_{(i,i')\gamma} = U_{(i,i')\gamma}\Sigma_{\gamma} , \tag{10.19}$$

$$H^{R}_{\gamma,(j,j')} = V_{\gamma(j,j')} . \tag{10.20}$$

Using this decomposition, we can convert any two-site operator into a sum of tensor products of local operators, where the number of terms in the sum is equal to the number of nonzero singular values. The speedup of using this method together with the routines for computing expectations of tensor products versus using the two-site density matrix is striking. This is especially true in systems of bosons with large local dimension where the computation of the energy using the non-tensor-product methods can be the dominant scaling operation. It should also be noted that what we have affected is a decomposition of a rank-4 tensor into a matrix product operator. This same basic decomposition can be applied to turn an operator of any rank into a matrix product operator, see Sec. 7.2.2.

10.2.2 The Python Front End

To discuss the Python front end for the ALPS TEBD code, called *PyALPS*, it is simplest to start with an example, here provided by the TEBD tutorial `tutorial1a.py`. The file begins by loading the required Python modules, here PyALPS and routines for plotting output.

```
import pyalps
import matplotlib.pyplot as plt
import pyalps.plot
```

We now proceed to prepare all of the metadata defining our simulations using an array of Python dictionaries called `parms`

```
parms=[]
count=0
for A in [5.0, 10.0, 15.0, 25.0, 50.0]:
        count+=1
        parms.append({
                  'L'                          : 10,
                  'MODEL'                      : 'hardcore␣boson',
                  'CONSERVED_QUANTUMNUMBERS'   : 'N',
                  'N' : 5,
                  't'                          : 1.0,
                  'V'                          : 10.0,
                  'ITP_CHIS' : [20, 30, 35],
                  'ITP_DTS' : [0.05, 0.05,0.025],
                  'ITP_CONVS' : [1E-8, 1E-8, 1E-9],
                  'INITIAL_STATE' : 'ground',
                  'CHI_LIMIT' : 40,
                  'TRUNC_LIMIT' : 1E-12,
                  'NUM_THREADS' : 1,
                  'TAUS' : [A,  A],
                  'POWS' : [ 1.0, 1.0],
                  'GS' : ['V',  'V'],
                  'GIS' : [10.0,  0.0],
                  'GFS' : [0.0,  10.0],
                  'NUMSTEPS' : [500,  500],
                  'STEPSFORSTORE' : [5, 3],
                  'SIMID' : count
                })
```

The parameters which are included in **parms** are discussed in detail in the ALPS documentation, see Appendices C.1 and C.2. Here we only note that what results from these lines is a length-5 array **parms**. Each element of parms is a Python dictionary, which is an unordered set of key:value pairs. The keys in this instance are the strings on the left column, e.g. 'L' and 'MODEL', and the values form the right column. Thus, `parms[0]['L']` would return 10. Each element of **parms** defines a TEBD simulation.

We now write Fortran-readable files from this simulation metadata with **writeTEBDfiles** and run simulations using this input with **runTEBD** via

```
baseName='tutorial_1a'
#write output files
nmlnameList=pyalps.writeTEBDfiles(parms, baseName)
#run the application
res=pyalps.runTEBD(nmlnameList)
```

We extract information from the outputs of the Fortran code using `loadTimeEvolution` as

```
LEdata=pyalps.load.loadTimeEvolution(pyalps.getResultFiles(prefix
    ='tutorial_1a'), measurements=['Loschmidt Echo', 'V'])
```

The **measurements** tag specifies which outputs to load. **LEdata** is an array with the same number of elements as **parms**, and contains all of the same simulation metadata together with the specified measurements. This allows for complex evaluations to be performed involving parameters in the Hamiltonian, the system size, or any other input parameters. In the present case, we are interested only in plotting the Loschmidt echo and the parameter V vs. time. We do so by turning these values into ordered (x, y) pairs suitable for a 2D plot with `collectXY`, and then passing these pairs to a plotting front end

```
LE=pyalps.collectXY(LEdata, x='Time', y='Loschmidt Echo',foreach
    =['SIMID'])
for q in LE:
```

```
    q.props['label']=r'$\tau=$'+str(q.props['TAUS'][0])

plt.figure()
pyalps.plot.plot(LE)
plt.xlabel('Time␣$t$')
plt.ylabel('Loschmidt␣Echo␣$|<␣\psi(0)|\psi(t)␣>␣|^2$')
plt.title('Loschmidt␣Echo␣vs.␣Time')
plt.legend(loc='lower␣right')

Ufig=pyalps.collectXY(LEdata, x='Time', y='V',foreach=['SIMID'])
for q in Ufig:
    q.props['label']=r'$\tau=$'+str(q.props['TAUS'][0])

plt.figure()
pyalps.plot.plot(Ufig)
plt.xlabel('Time␣$t$')
plt.ylabel('V')
plt.title('Interaction␣parameter␣$V$␣vs.␣Time')
plt.legend(loc='lower␣right')
plt.show()
```

The resulting plots are shown as Figure 11.2 in Chapter 11, where the physical content of these tutorials are discussed in greater detail.

While the actual NAMELIST files which are passed to the Fortran routines are similar to those of OSTEBD above, the Python front end provides a powerful way to automate the writing of these inputs, allows for much easier post-processing of outputs, and also unifies the preparation, execution, and analysis of simulations.

The contributions of the present author to the Python front end were to write the Python routines to write Fortran-readable files for TEBD, run TEBD simulations, and process the outputs of TEBD as a function of time. Since TEBD was the first of the ALPS applications to simulate dynamics, the author also modified several of the other existing ALPS processing routines to accept data which depends on time.

10.2.3 Integration with the VisTrails Workflow Provenance System

The Python front end for ALPS provides a unified way of defining, executing, and analyzing simulations. ALPS v2.0 is also integrated with VisTrails [19], which is an open-source scientific workflow and provenance management system written

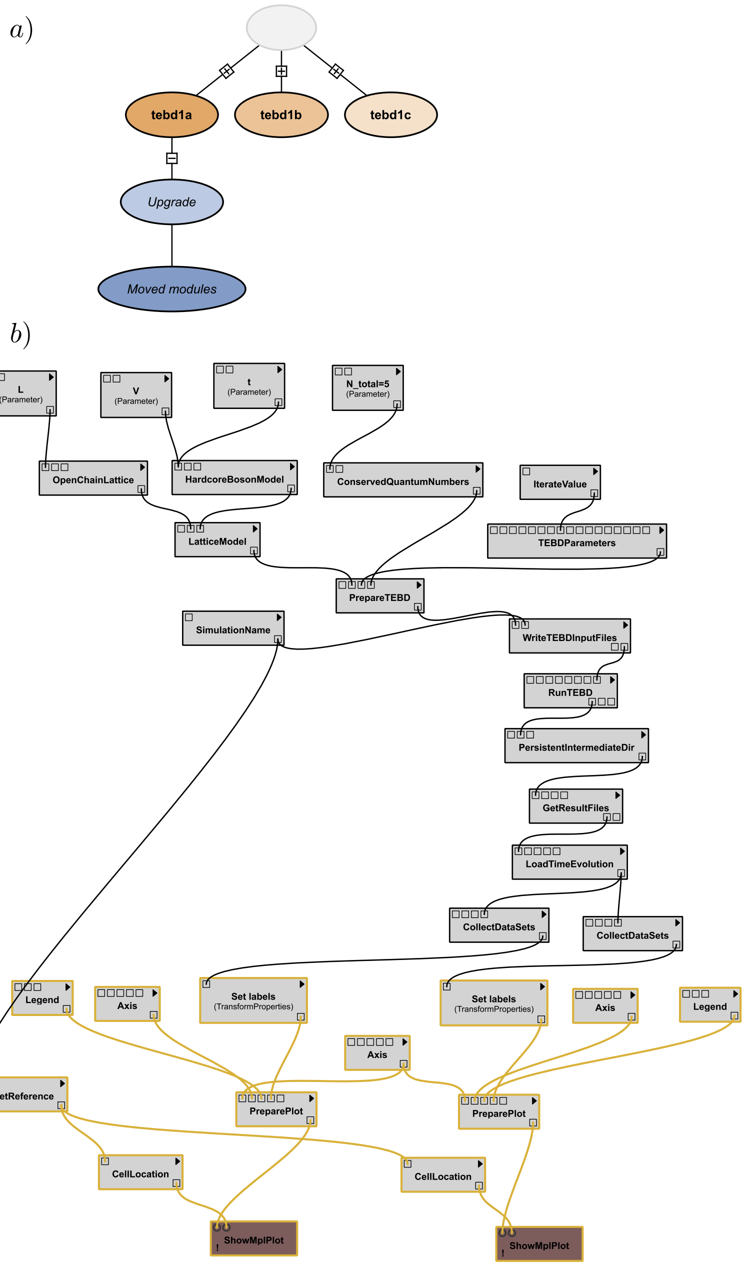


Figure 10.3: Example VisTrail for ALPS TEBD tutorial 1. a) The workflows for tutorial 1 are represented as the beige circles. Changes to the workflow of tebd1a are represented as blue ovals below the beige one. b) The contents of the workflow tebd1a. Here each large box represents a task performed by PyALPS or an internal VisTrails task.

in Python. By provenance, we refer to both the steps that need to be followed to produce a specific result (*Prospective provenance*) as well as the steps which were actually taken (*Retrospective provenance*) [30, 31]. Hence, the simulation metadata above, which user ran the simulation, at what time, etc. are all data relevant to the provenance of a particular result. VisTrails caches all of this data in persistent storage such that results are always reproducible without the need to repeat the calculation.

The same tutorial which was used to exemplify the Python front end is shown using VisTrails in Figure 10.3. In panel (a), the three parts to tutorial 1 are shown as three different workflows, the beige ovals. Under tebd1a, the blue boxes capture provenance information about changes made to the workflow itself. The actual workflow contents of tebd1a are shown in panel (b). Here the large boxes, called *modules* represent subtasks to be performed, with the small boxes on the top corresponding to inputs for a particular task and the boxes on the bottom corresponding to outputs from the task. The overall simulation flow is similar to that using the Python front end: parameters are specified or looped over, Fortran-readable files are written, the simulation is performed, data is extracted and post-processed, and plots are made. With VisTrails, the full provenance information of both how a simulation is to be performed as well as what the actual values used to achieve a particular result are permanently captured. The contributions of the present author to the VisTrails functionality of ALPS were modules to write files for and run TEBD simulations as well as the post-processing of time series.

10.3 References Cited

[1] Open source time-evolving block decimation. `http://physics.mines.edu/downloads/software/tebd/`.

[2] M. L. Wall and L. D. Carr. Emergent timescales in entangled quantum dynamics of ultracold molecules in optical lattices. *New J. Phys.*, 11:055027, 2009.

[3] L. D. Carr and M. K. Oberthaler. Spatial dependence of entropy in trapped ultracold Bose gases. *e-print arXiv:0906.4708*, 2009.

[4] R. V. Mishmash and L. D. Carr. Quantum Entangled Dark Solitons formed by Ultracold Atoms in Optical Lattices. *Phys. Rev. Lett.*, 103:140403, 2009.

[5] R. V. Mishmash, I. Danshita, Charles W. Clark, and L. D. Carr. Quantum Many-Body Dynamics of Dark Solitons in Optical Lattices. *Phys. Rev. A*, 80: 053612, 2009.

[6] L. D. Carr, M. L. Wall, D. G. Schirmer, R. C. Brown, J. E. Williams, and C. W. Clark. Mesoscopic Effects in Quantum Phases of Ultracold Quantum Gases in Optical Lattices. *Phys. Rev. A*, 81:013613, 2010.

[7] M. L. Wall and L. D. Carr. Hyperfine molecular Hubbard Hamiltonian. *Phys. Rev. A*, 82:013611, 2010.

[8] B. Bauer, L. D. Carr, H. G. Evertz, A. Feiguin, J. Freire, S. Fuchs, L. Gamper, J. Gukelberger, E. Gull, S. Guertler, A. Hehn, R. Igarashi, S. V. Isakov, D. Koop, P. N. Ma, P. Mates, H. Matsuo, O. Parcollet, G. Pawlowski, J. D. Picon, L. Pollet, E. Santos, V. W. Scarola, U. Schollwöck, C. Silva, B. Surer, S. Todo, S. Trebst, M. Troyer, M. L. Wall, P. Werner, and S. Wessel. The ALPS project release 2.0: Open source software for strongly correlated systems. *J. Stat. Mech.: Theor. Exp.*, page P05001, 2011.

[9] J. A. Glick and L. D. Carr. Macroscopic quantum tunneling of solitons in Bose-Einstein condensates. *Phys. Rev. Lett. under review, arXiv:1105.5164*, 2011.

[10] Giuseppe Carleo, Frederico Becca, Marco Schiró, and Michele Fabrizo. Localization and Glassy Dynamics Of Many-Body Quantum Systems. *Sci. Rep.*, 2(243), 2011.

[11] Yinyin Qian, Ming Gong, and Chuanwei Zhang. Quantum transport of bosonic cold atoms in double-well optical lattices. *Phys. Rev. A*, 84:013608, Jul 2011. doi: 10.1103/PhysRevA.84.013608. URL http://link.aps.org/doi/10.1103/PhysRevA.84.013608.

[12] Zheng-Wei Zhou, Shao-Liang Zhang, Xiang-Fa Zhou, Guang-Can Guo, Xingxiang Zhou, and Han Pu. Quantum phase transition of Bose-Einstein condensates on a nonlinear ring lattice. *Phys. Rev. A*, 83:043626, Apr 2011. doi: 10.1103/PhysRevA.83.043626. URL http://link.aps.org/doi/10.1103/PhysRevA.83.043626.

[13] Hong Lu, L. O. Baksmaty, C. J. Bolech, and Han Pu. Expansion of 1D Polarized Superfluids: The Fulde-Ferrell-Larkin-Ovchinnikov State Reveals Itself. *Phys. Rev. Lett.*, 108:225302, May 2012. doi: 10.1103/PhysRevLett.108.225302. URL http://link.aps.org/doi/10.1103/PhysRevLett.108.225302.

[14] Yinyin Qian, Ming Gong, and Chuanwei Zhang. Many-body Landau-Zener Transition in Cold Atom Double Well Optical Lattices. *http://arxiv.org/abs/1110.1653*, 2012.

[15] Shuai Yin, Xizhou Qin, Chaohong Lee, and Fan Zhong. Finite-time scaling of dynamic quantum criticality. *http://arxiv.org/abs/1207.1602*, 2012.

[16] ALPS. http://alps.comp-phys.org.

[17] A. F. Albuquerque, F. Alet, P. Corboz, P. Dayal, A. Feiguin, S. Fuchs, L. Gamper, E. Gull, S. Gurtler, A. Honecker, R. Igarashi, M. Korner, A. Kozhevnikov, A. Lauchli, S. R. Manmana, M. Matsumoto, I. P. McCulloch, F. Michel, R. M. Noack, G. Pawlowski, L. Pollet, T. Pruschke, U. Schollwock, S. Todo, S. Trebst, M. Troyer, P. Werner, and S. Wessel. The ALPS project release 1.3: Open-source software for strongly correlated systems. *Journal of Magnetism and Magnetic Materials*, 310:1187–1193, 2007.

[18] http://www.python.org/.

[19] http://www.vistrails.org/.

[20] G. Vidal. Efficient Classical Simulation of Slightly Entangled Quantum Computations. *Phys. Rev. Lett.*, 91:147902–1–4, 2003.

[21] G. Vidal. Efficient Simulation of One-Dimensional Quantum Many-Body Systems. *Phys. Rev. Lett.*, 93:040502, 2004.

[22] Ippei Danshita and Pascal Naidon. Bose-Hubbard ground state: Extended Bogoliubov and variational methods compared with time-evolving block decimation. *Phys. Rev. A*, 79:043601, Apr 2009. doi: 10.1103/PhysRevA.79.043601. URL http://link.aps.org/doi/10.1103/PhysRevA.79.043601.

[23] Andrew John Daley. *Manipulation and Simulation of Cold Atoms in Optical Lattices*. PhD thesis, Leopold-Franzens-Universität Innsbruck, 2005.

[24] Ian P. McCulloch. From density-matrix renormalization group to matrix product states. *J. Stat. Mech.*, 2007(10):P10014, 2007. URL http://stacks.iop.org/1742-5468/2007/i=10/a=P10014.

[25] Message Passing Interface Forum. http://www.mpi-forum.org/.

[26] L. Scott, Terry Clark, and Babak Bagheri. *Scientific Parallel Computing*. Princeton University Press, Princeton, NJ, 2005.

[27] R. Orús and G. Vidal. Infinite time-evolving block decimation algorithm beyond unitary evolution. *Phys. Rev. B*, 78:155117, Oct 2008. doi: 10.1103/PhysRevB.78.155117. URL http://link.aps.org/doi/10.1103/PhysRevB.78.155117.

[28] A. T. Sornborger and E. D. Stewart. Higher-order methods for simulations on quantum computers. *Phys. Rev. A*, 60:1956–1965, Sep 1999. doi: 10.1103/PhysRevA.60.1956. URL http://link.aps.org/doi/10.1103/PhysRevA.60.1956.

[29] Open Multiprocessing. http://openmp.org/wp/.

[30] C.T. Silva, J. Freire, and S.P. Callahan. Provenance for visualizations: Reproducibility and beyond. *IEEE Computing in Science and Engineering*, 9(5):82, 2007. doi: 10.1109/MCSE.2007.106.

[31] Juliana Freire, David Koop, Emanuele Santos, and Cláudio T. Silva. Provenance for computational tasks: A survey. *Computing in Science and Engg.*, 10(3): 11–21, May 2008. ISSN 1521-9615. doi: 10.1109/MCSE.2008.79. URL http://dx.doi.org/10.1109/MCSE.2008.79.

CHAPTER 11
EDUCATIONAL MATERIALS

Chapter 10 discusses the author's involvement in the development of open source code implementing variational matrix product state (MPS) algorithms. The present chapter covers educational materials which were distributed with these codes, either as part of the open source packages or within the Carr theoretical physics research group (CTPRG). The educational materials which were disseminated with the open source code are aimed at the level of graduate students performing research in strongly correlated physics, and are designed to enable users to modify the open source codes to meet their own research goals. The materials designed for use by the CTPRG are aimed more to the level of undergraduates who may not have completed a quantum mechanics course and may have no coding experience. Hence, they must strike a balance between being conveying the important ideas of the algorithms without being too technical, and not require coding experience while still enabling the student to perform calculations which are meaningful and exciting. Within the CTPRG several undergraduates at the senior level were able to use these educational materials successfully towards modifying and using the open source codes as part of their senior thesis projects.

11.1 Materials Distributed with the Open Source Packages

The open source time-evolving block decimation (OSTEBD) [1] package includes a manual written by the present author which provides background on the time-evolving block decimation (TEBD) algorithm [2, 3] and manual pages for all routines contained in the package. This manual is included with the thesis as Appendix B. The manual also contains case studies whose documentation begins with Appendix B.4. Provided with each case study is a main file from which a user can reproduce the figures given in the manual in order to understand how to run the code and post-process its output. The case studies, in order, are a study of the ground state properties of the Bose-Hubbard Hamiltonian, the dynamics of the Bose-Hubbard Hamiltonian following a linear ramp of the on-site interaction from the Mott insulator to the superfluid phase and back [4, 5], the ground state properties of spinless fermions, and the dynamics of a domain wall defect in the XX model [6]. At the end of each case study are exercises which require the user to modify the case study base code to extract other observables or perform other tasks. For example, in the Bose-Hubbard case study, the user is required to add a harmonic trap in one of the exercises and study the physics of the trapped system. Several of the exercises, in particular those dealing with dynamics, have the user study the behavior of the simulation as the bond dimension χ, the time step δt, the system size, and the boundary conditions are changed. These exercises help the user gain an intuition for the convergence behavior of typical simulations.

The educational materials distributed with the Algorithms and Libraries for Physics Simulations (ALPS) [7–9] TEBD routines take the form of case studies which were posted on the ALPS wiki pages at http://alps.comp-phys.org. These case studies and other ALPS TEBD documentation from the ALPS wiki are included as Appendix C with this thesis. The second tutorial is the more pedagogical, as it uses comparisons with analytically known results to demonstrate the convergence of TEBD simulations. In particular, this tutorial uses the same system as the last case study for OSTEBD in which a single domain wall defect is initialized at the center of a long chain:

$$|\psi\rangle = |\downarrow\downarrow\dots\downarrow\uparrow\dots\uparrow\uparrow\rangle\,. \tag{11.1}$$

At time $t=0$, this wavefunction is evolved according to the XX model

$$\hat{H}_{\mathrm{XX}} = -\sum_i \left(\hat{S}^x_i\hat{S}^x_{i+1} + \hat{S}^y_i\hat{S}^y_{i+1}\right)\,, \tag{11.2}$$

where $\hat{S}_i^{\nu}$ is the spin-1/2 operator along the ν^{th} Cartesian direction at lattice site i. This Hamiltonian is equivalent to a system of free fermions via a Jordan-Wigner transformation, and so we can solve exactly for the magnetization a distance n from the initial defect position at time t, $M\left(n,t\right)$, as [10]

$$M\left(n,t\right) = -\frac{1}{2}\sum_{i=1-n}^{n-1} j_l^2\left(t\right) \,, \tag{11.3}$$

where $j_l\left(x\right)$ is the Bessel function of order l. This result also implies that in the limit as $n \to \infty$ and $t \to \infty$, this function approaches as scaling form which depends only on the variable n/t as

$$\lim_{n\to\infty}\lim_{t\to\infty} M\left(n,t\right) = \phi\left(\frac{n}{t}\right) \equiv -\frac{1}{\pi}\arcsin\left(\frac{n}{t}\right) \,. \tag{11.4}$$

The results comparing these predictions with the outputs of the ALPS TEBD tutorial `tutorial2a.py` are shown in Figure 11.1. We see very good visual agreement between the numerical results and the predictions for the magnetization in Figure 11.1(a). Figure 11.1(b) demonstrates that the magnetization does indeed approach the scaling limit Eq. (11.4), with agreement improving for large n and t, as expected. This part of the tutorial also demonstrates the power of the Python front end for performing complex post-processing such as the extraction of scaling forms from the numerical data, see Sec. 10.2.2.

In the next two parts of this tutorial, we examine the deviations of the magnetization with these exact results as we change the two main convergence parameters in TEBD: the time step δt and the bond dimension χ. In Figure 11.1(c) we see the deviation of the magnetization as a function of time and the time step δt. At short times, the smallest δt has the smallest error in accordance with our expectations given by the error bounds on the Trotterization of the propagator. However, at long times, we begin to see an exponential growth in the errors for the smallest δt, and

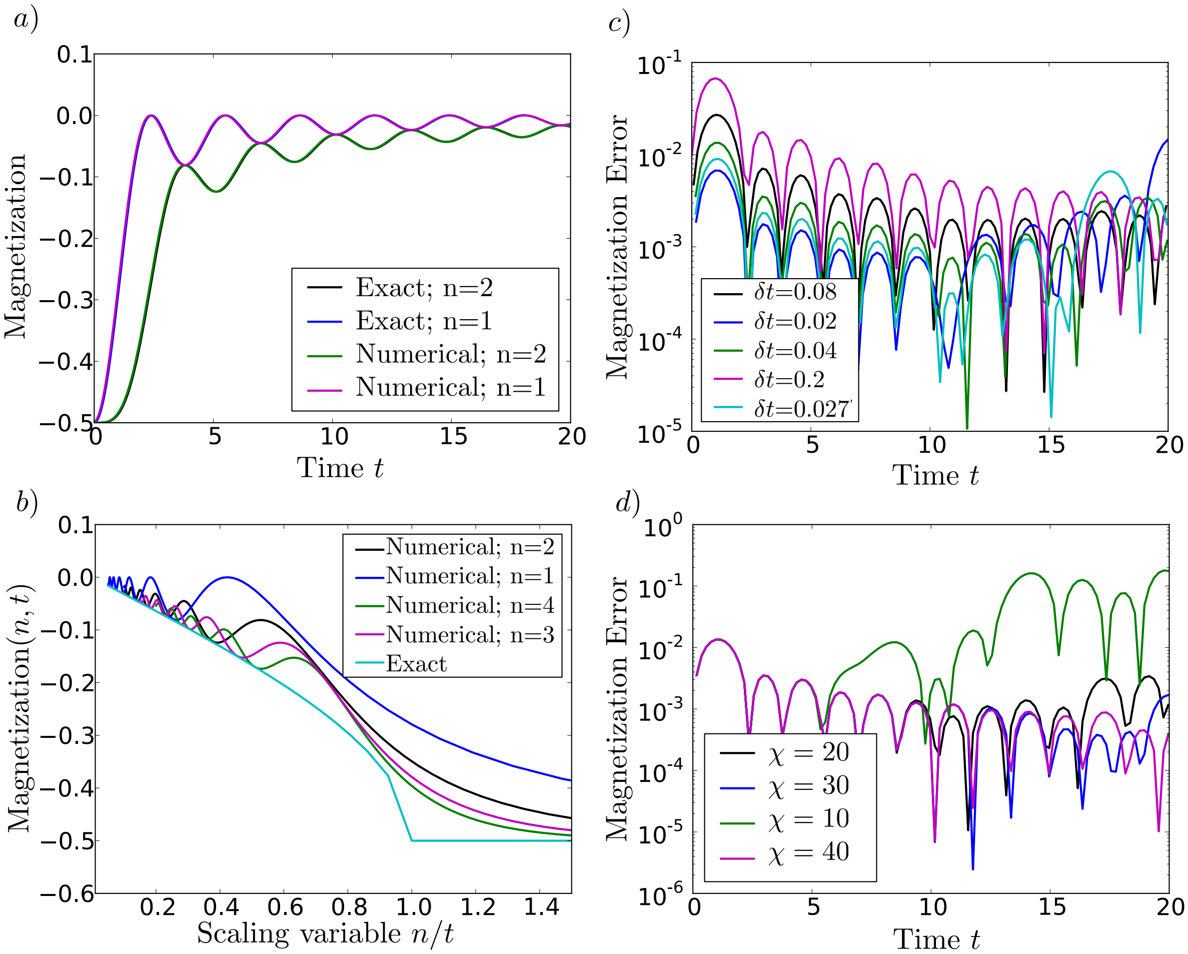


Figure 11.1: Results of ALPS TEBD tutorial 2. a) The magnetization computed by TEBD for the given situation compared to the prediction Eq. (11.3). b) The scaling limit of the magnetization compared to the scaling function predicted in Eq. (11.4). c) The error in the magnetization as a function of time and the time step δt. d) The error in the magnetization as a function of time and the bond dimension χ.

this simulation eventually has the largest error of all the simulations. The reason for this is that at each time step we perform a truncation of the bond dimension, and this truncation involves a renormalization of the wavefunction. Hence, in contrast to the error due to the Trotterization of the propagator which grows linearly in time, the error due to this renormalization grows approximately exponentially in time. The timescale on which this exponentially growing error overtakes the linearly growing error can be pushed to later times by increasing χ, as shown in Figure 11.1(d). Hence, the errors incurred by the time step and the bond dimension have a subtle interplay in which larger time steps make better use of the available bond dimension to reach a certain fixed time, but introduce a larger linearly growing error. This causes special difficulty for systems which are required to have a small time step when using non-time ordered Trotterization schemes due to a fast rate of change of the Hamiltonian. This observation was part of the motivation for the general use time-ordered time evolution scheme proposed in Chapter 7. The final part of this tutorial introduces interactions[89] $J_z \hat{S}_i^z \hat{S}_{i+1}^z$ to the spin chain, and investigates how the behavior of the time-evolved magnetization changes [11].

The other tutorial provided for the ALPS routines is simpler, and so was included first in spite of the fact that the second tutorial is more pedagogical. This tutorial studies the behavior of a model of hardcore bosons with nearest neighbor hopping and nearest-neighbor density-density interactions,

$$\hat{H} = -t \sum_{\langle i,j \rangle} \left[\hat{b}_i^\dagger \hat{b}_j + \text{h.c.} \right] + V \sum_{\langle i,j \rangle} \hat{n}_i \hat{n}_j \,, \tag{11.5}$$

as the strength of the interactions V/t[90] is changed in time. The measure of adiabaticity we use is the Loschmidt echo

---

[89] These are interactions in that they provide terms which are quartic in the fermion field operators following a Jordan-Wigner transformation.

[90] In the expression V/t, t denotes the nearest-neighbor tunneling and not the time.

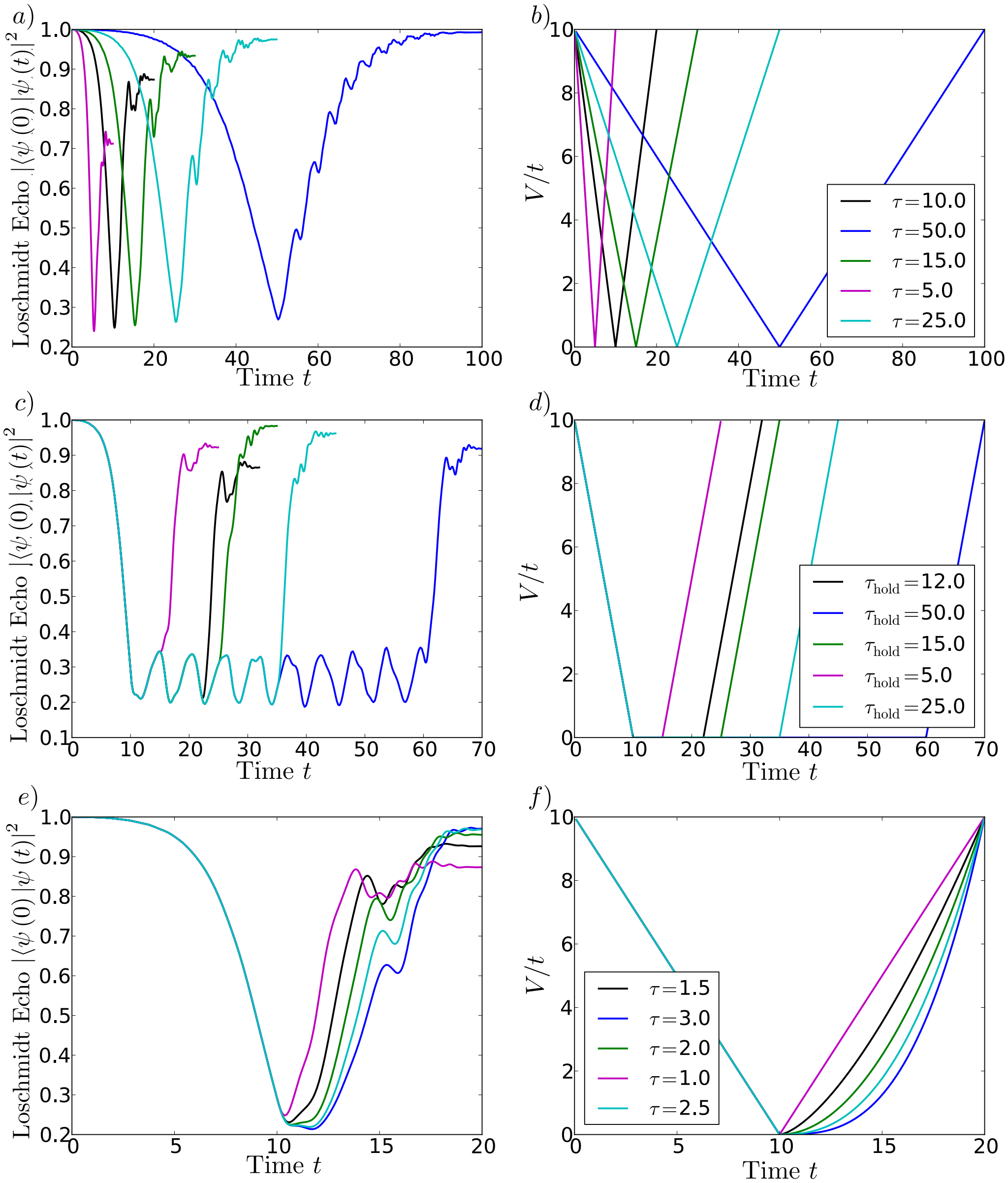


Figure 11.2: Results of ALPS TEBD tutorial 1. The color scheme in a panel in the left column is the same as the neighboring panel in the right column. a) The Loschmidt echo for a process which quenches V/t from the CDW phase to the superfluid and back as a function of the linear timescale of the quench τ. b) The form of V/t as a function of time for the simulation in (a). c) The Loschmidt echo for a process which quenches V/t from the CDW phase to the superfluid, holds the parameters constant for a time $\tau_{\rm hold}$, and then quenches back as a function of the hold time $\tau_{\rm hold}$. d) The form of V/t as a function of time for the simulation in (c). e) The Loschmidt echo for a process which quenches V/t linearly from the CDW phase to the superfluid and nonlinearly quenches back as a function of the power of the return quench. f) The form of V/t as a function of time for the simulation in (e).

$$L\left(t;\gamma\right)=\left|\langle\psi\left(t\right)|\psi\left(0\right)\rangle\right|^2\ , \tag{11.6}$$

which is the squared overlap of the time-evolved wavefunction with the initial wavefunction. This quantity depends on general on the way in which the parameters are changed in time; this dependence is denoted by the symbol γ.

The first part of the tutorial focuses on quenches from the gapped charge-density wave (CDW) phase to the superfluid phase and back. The exercises explore how difficult it is for such a quench to be adiabatic in the sense that the system returns to its initial configuration, and how this depends on the system size [12–16]. An exercise which represents a possible research project is to extend the simulation to the soft-core boson model with nearest-neighbor interactions in which there exist Mott insulating, CDW, and superfluid phases at unit filling. One can then explore how difficult it is to be adiabatic when quenching between two gapped phases, here the Mott insulator and the CDW. In the next part of the tutorial, the system is held in the superfluid phase for a time τ_{hold} before quenching back to the CDW. The final part of the tutorial utilizes the capability of the ALPS TEBD routines to also simulate time evolutions described by nonlinear power law behavior of Hamiltonian parameters. The outputs of the simulations in this tutorial are shown in Figure 11.2.

11.2 Materials intended for Carr Theoretical Physics Research Group Use

In addition to the OSTEBD manual, which contains a great amount of detail, a smaller work “A gentle introduction to Time Evolving Block Decimation (TEBD)” (henceforth, the TEBD intro) was developed for use by advanced undergraduate and beginning graduate students within the CTPRG. The purpose of this document is to acquaint the student with the basic idea of how MPS algorithms make the many-body problem tractable through restricting entanglement, to provide experience in running and post-processing MPS code, to develop familiarity with the high-performance and parallel computing environments available to CTPRG members, and to enable the student to perform an exciting, cutting-edge calculation regarding quantum phase transitions. This document is included as Appendix D.

The TEBD intro builds upon an earlier set of exercises "Introduction to the Bose-Hubbard Model and Fock State Basis" which provides a basic introduction to strongly correlated lattice models, Fock space, and second quantization. The final exercise in this latter problem set is to write an exact diagonalization program to compute the spectrum of the Bose-Hubbard model on M sites with N particles and at most $d-1$ particles per site. The introductory TEBD exercises begin by having the student explore for what size systems exact diagonalization is numerically practical. For most of the implementations that the author has seen, this is usually 8 to 10 sites. Then, the notion of the singular value decomposition (SVD) and reduced rank approximations are introduced. Rather than discuss tensor networks in general, it is demonstrated how a matrix represented as its singular value decomposition may be applied to vectors much more cheaply than the full matrix representation when the rank of the matrix is small compared to its dimension. Mathematical and programming exercises on the SVD acquaint the user with its properties, and also demonstrate that the rank of a matrix may be difficult to ascertain from its other properties. A physical connection between quantum mechanical states and the SVD is provided by the Schmidt decomposition [17, 18], and the matrices in the mathematical SVD exercises are related to quantum states. Finally, the ideas are put together noting that TEBD performs a truncated Schmidt decomposition at every bipartite splitting, hence restricting the amount of entanglement in the state and allowing for more efficient operations.

At the end of the introductory TEBD materials, some exercises are provided to orient the user with running the OSTEBD code and to introduce quantum phase transitions [6, 19]. The first few exercises consist of comparing the results of small Bose-Hubbard systems with exact diagonalization with and without number conservation. The next exercise requires the student to diagonalize the bosonic tight-binding chain via a Fourier transform for both open and periodic boundary conditions and compute the spectrum, the on-site number, and the single-particle density matrix. These results are compared with the open source code. The exercise also acquaints the student with other useful many-body tools such as canonical transformations. The final exercise makes use of the data-parallel capability of OSTEBD (see Sec. 10.1.1) to explore the Mott insulator-superfluid transition in the Bose-Hubbard model. As such, this exercise also serves to introduce students to high-performance and parallel computing environments. The exercise begins with a qualitative treatment of why the ground state must be non-analytic based on strong and weak coupling limits of the depletion, relying on the student's intuition for the single-particle density matrix developed by the tight-binding chain. The critical point is then located numerically via the maximum of the fidelity susceptibility, which gives a system-independent measure of changes in the ground state as a system passes through a quantum critical point [20]. An algebraic scaling function for the position of the fidelity susceptibility maximum as a function of the system size is provided [21], and the student is asked to find the critical point using a scaling analysis for a series of system sizes. Estimates for the appropriate range of parameters are provided by fidelity susceptibility data on small system sizes.

The final educational document to be discussed is “Introduction to MPS Algorithms,” (henceforth, the MPS intro) which is included in its most recent version as Appendix E. The MPS intro was developed for future code developers in the CTPRG to write their own small working variational MPS code for the Ising model. This process is guided by introducing a theoretical concept, for example canonical forms for MPSs, and then having the student write code with a specific interface to perform the procedure outlined in theory. A highly stripped-down and de-optimized version of the author's code was provided along with this document and discussed in Appendix A. Each of the procedures of the author's code has the same interface as the procedures outlined in the document, and so students can consult the author's code for implementation hints if they are stuck. The author's code can also be used as a black box to obtain observables for the Ising model for varying system parameters, and so the student can verify his or her own code for correctness. The topics covered are canonical forms, matrix product operators including a section on long-range operators [22–24], caching of effective Hamiltonian overlaps [25] and the formation of the effective Hamiltonian from them, the sparse solution of the effective Hamiltonian eigenproblem using the Lanczos recursion [26, 27], computation of observables, calculation of excited states using eMPS,[91] and TEBD-style time evolution.

11.3 References Cited

[1] Open source time-evolving block decimation. http://physics.mines.edu/downloads/software/tebd/.

[2] Guifré Vidal. Efficient Classical Simulation of Slightly Entangled Quantum Computations. *Phys. Rev. Lett.*, 91(14):147902, Oct 2003. doi: 10.1103/PhysRevLett.91.147902.

[3] Guifré Vidal. Efficient Simulation of One-Dimensional Quantum Many-Body Systems. *Phys. Rev. Lett.*, 93(4):040502, Jul 2004. doi: 10.1103/PhysRevLett.93.040502.

[4] S. R. Clark and D. Jaksch. Dynamics of the superfluid to Mott-insulator transition in one dimension. *Phys. Rev. A*, 70:043612, Oct 2004. doi: 10.1103/PhysRevA.70.043612. URL http://link.aps.org/doi/10.1103/PhysRevA.70.043612.

[5] Andrew John Daley. *Manipulation and Simulation of Cold Atoms in Optical Lattices*. PhD thesis, Leopold-Franzens-Universität Innsbruck, 2005.

---

[91] See Chapter 7 for a discussion of eMPS.

[6] S. Sachdev. *Quantum Phase Transitions.* Cambridge University Press, New York, 1999.

[7] ALPS. `http://alps.comp-phys.org`.

[8] A. F. Albuquerque, F. Alet, P. Corboz, P. Dayal, A. Feiguin, S. Fuchs, L. Gamper, E. Gull, S. Gurtler, A. Honecker, R. Igarashi, M. Korner, A. Kozhevnikov, A. Lauchli, S. R. Manmana, M. Matsumoto, I. P. McCulloch, F. Michel, R. M. Noack, G. Pawlowski, L. Pollet, T. Pruschke, U. Schollwock, S. Todo, S. Trebst, M. Troyer, P. Werner, and S. Wessel. The ALPS project release 1.3: Open-source software for strongly correlated systems. *Journal of Magnetism and Magnetic Materials*, 310:1187–1193, 2007.

[9] B. Bauer, L. D. Carr, H. G. Evertz, A. Feiguin, J. Freire, S. Fuchs, L. Gamper, J. Gukelberger, E. Gull, S. Guertler, A. Hehn, R. Igarashi, S. V. Isakov, D. Koop, P. N. Ma, P. Mates, H. Matsuo, O. Parcollet, G. Pawłowski, J. D. Picon, L. Pollet, E. Santos, V. W. Scarola, U. Schollwöck, C. Silva, B. Surer, S. Todo, S. Trebst, M. Troyer, M. L. Wall, P. Werner, and S. Wessel. The ALPS project release 2.0: open source software for strongly correlated systems. *Journal of Statistical Mechanics: Theory and Experiment*, 2011(05):P05001, 2011. URL `http://stacks.iop.org/1742-5468/2011/i=05/a=P05001`.

[10] T. Antal, Z. Rácz, A. Rákos, and G. M. Schütz. Transport in the XX chain at zero temperature: Emergence of flat magnetization profiles. *Phys. Rev. E*, 59:4912–4918, May 1999. doi: 10.1103/PhysRevE.59.4912. URL `http://link.aps.org/doi/10.1103/PhysRevE.59.4912`.

[11] Dominique Gobert, Corinna Kollath, Ulrich Schollwöck, and Gunter Schütz. Real-time dynamics in spin-$\frac{1}{2}$ chains with adaptive time-dependent density matrix renormalization group. *Phys. Rev. E*, 71:036102, Mar 2005. doi: 10.1103/PhysRevE.71.036102. URL `http://link.aps.org/doi/10.1103/PhysRevE.71.036102`.

[12] T. W. B. Kibble. Topology of cosmic domains and strings. *J. Phys. A: Math Gen.*, 9:1387, 1980.

[13] W. H. Zurek. Cosmological experiments in superfluid Helium? *Nature*, 317: 505–508, 1985.

[14] T. Kibble. Phase-transition dynamics in the lab and the universe. *Phys. Today*, 60(9):47, 2007.

[15] Bogdan Damski and Wojciech H. Zurek. Dynamics of a Quantum Phase Transition in a Ferromagnetic Bose-Einstein Condensate. *Phys. Rev. Lett.*, 99:130402, 2007.

[16] Hiroki Saito, Yuki Kawaguchi, and Masahito Ueda. Kibble-Zurek mechanism in a quenched ferromagnetic Bose-Einstein condensate. *Phys. Rev. A*, 76:043613, 2007.

[17] Erhard Schmidt. Zur Theorie der linearen und nichtlinearen Integralgleichungen. *Mathematische Annalen*, 63:433–476, 1907. ISSN 0025-5831. URL `http://dx.doi.org/10.1007/BF01449770`. 10.1007/BF01449770.

[18] M. Nielsen and I. Chuang. *Quantum Computation and Quantum Information.* Cambridge University Press, Cambridge, 2000.

[19] Lincoln D. Carr, editor. *Understanding Quantum Phase Transitions*. CRC press, 2010.

[20] L. D. Carr, M. L. Wall, D. G. Schirmer, R. C. Brown, J. E. Williams, and Charles W. Clark. Mesoscopic effects in quantum phases of ultracold quantum gases in optical lattices. *Phys. Rev. A*, 81(1):013613, Jan 2010. doi: 10.1103/PhysRevA.81.013613.

[21] P. Buonsante and A. Vezzani. Ground-state Fidelity and Bipartite Entanglement in the Bose-Hubbard Model. *Phys. Rev. Lett.*, 98:110601, 2007.

[22] Gregory M. Crosswhite, A. C. Doherty, and Guifré Vidal. Applying matrix product operators to model systems with long-range interactions. *Phys. Rev. B*, 78:035116, Jul 2008. doi: 10.1103/PhysRevB.78.035116. URL `http://link.aps.org/doi/10.1103/PhysRevB.78.035116`.

[23] B Pirvu, V Murg, J I Cirac, and F Verstraete. Matrix product operator representations. *New Journal of Physics*, 12(2):025012, 2010. URL `http://stacks.iop.org/1367-2630/12/i=2/a=025012`.

[24] F. Fröwis, V. Nebendahl, and W. Dür. Tensor operators: Constructions and applications for long-range interaction systems. *Phys. Rev. A*, 81:062337, Jun 2010. doi: 10.1103/PhysRevA.81.062337. URL `http://link.aps.org/doi/10.1103/PhysRevA.81.062337`.

[25] Gregory M. Crosswhite and Dave Bacon. Finite automata for caching in matrix product algorithms. *Phys. Rev. A*, 78:012356, Jul 2008. doi: 10.1103/PhysRevA.78.012356. URL `http://link.aps.org/doi/10.1103/PhysRevA.78.012356`.

[26] G. H. Golub and C. F. Van Loan. *Matrix Computations.* Johns Hopkins Studies in Mathematical Sciences. The Johns Hopkins University Press, Baltimore, 3 edition, 1996.

[27] Lloyd N. Trefethen and David Bau III. *Numerical Linear Algebra*. SIAM, 1997.

PART VI

CONCLUSIONS AND APPENDICES

CHAPTER 12

CONCLUSIONS AND SUGGESTIONS FOR FUTURE RESEARCH

This thesis presents models for the low energy physics of molecules trapped in optical lattices and simulation methods to elucidate the many-body physics of these models in one dimension. Part II focused on the molecular Hubbard Hamiltonian (MHH), a model for the low-energy physics of $^1\Sigma$ heteronuclear bialkali molecules in an optical lattice [1, 2]. A strong DC electric field applied to these molecules gives rise to resonant dipole-dipole interactions which are long-range, and also allows for tunable access to rotational states other than the lowest rotational level via an AC microwave field. The nuclear spin degrees of freedom couple most strongly to a DC magnetic field, but the coupling of rotational and nuclear spin degrees of freedom provided by nuclear quadrupole interactions implies that the interplay between the rotational and nuclear spin degrees of freedom may be tuned by changing the angle between the electric and magnetic fields. Thus, not only the magnitudes of the parameters of the MHH, but also the number of states involved in the dynamics are amenable to experimental control. Hence, the MHH may be used as a simulator of a quantum complex system, one in which are large number of degrees of freedom are interacting quantum mechanically on a multitude of timescales.

For nearest-neighbor interactions and a single internal state, the MHH admits only superfluid and charge-density wave (CDW) phases [3]. The CDW is characterized by a "checkerboard" pattern in the density which arises as a peak in the structure factor at a wavevector of π/a. For convex long-range interactions, insulating phases appear at every rational filling [4, 5], and so the phase diagram is much richer. While theoretically appealing, these phases may exist only in a very narrow parameter range, and may be destroyed by terms such as tunneling between sites which are not nearest neighbors and thermal fluctuations. Hence, a consistent analysis would involve carefully enumerating all terms appearing in the many-body Hamiltonian in order of decreasing energy and then performing a self-consistent truncation down to an energy scale set by experimental constraints on timescale. The microscopic connections between the few-body physics and the many-body physics presented in this thesis are crucial for this kind of analysis, and hence also for practical guidance of experiments.

When more than one internal state is populated the possibilities for statics become even greater. As discussed in Chapters 3 and 4, both the tunneling and the interactions depend on the rotational state. This allows for the study of many-body systems in mixtures different quantum phases and possibly far from equilibrium. Furthermore, away from the strong DC field regime studied in Chapter 4, there is a long-ranged interaction term which represents the exchange of a rotational quantum between two molecules in different rotational states [6, 7]. For two internal states $\sigma = \{1, 2\}$, this term has the form

$$\sum_{i<j} E^{i,j} \hat{a}_{i1}^{\dagger} \hat{a}_{j2}^{\dagger} \hat{a}_{j1} \hat{a}_{i2} \,, \tag{12.1}$$

where i and j are lattice sites, $\hat{a}_{i\sigma}$ destroys a particle in internal molecular state σ at site i, and $E^{i,j}$ obeys a power law form for large distances $|i-j|$. The *exchange* contribution $E^{i,j}$ has a comparable power-law form to $U^{i,j}$, the direct part of the dipole-dipole interaction, but its magnitude can be tuned independently. The action of this term in the internal degrees of freedom of a molecule is similar to that of *ring-exchange* terms in real space [8]. Such terms have been shown to induce novel Bose metal and gapless Mott insulator phases in ladder systems of hard-core bosons [8, 9] and also drive electronic systems into a non-Fermi liquid phase [10]. The exchange term frustrates the system, and so studies of the MHH with the exchange term must be performed with either exact diagonalization or tensor network methods due to a sign problem in quantum Monte Carlo. Studies of the MHH with all terms accounted for to a self-consistent energetic cutoff, across a wide range of static and dynamical field regimes, and for different molecular species are currently under way [11].

In addition to the statics, the dynamics of the MHH were investigated in this thesis. The particular quench process studied initializes the system in the ground state of the MHH in the absence of an AC field. An AC field is suddenly turned on at a frequency corresponding to a single-molecule rotational resonance. In the absence of couplings between sites, each molecule would undergo coherent Rabi flopping independently of the others. However, when averaged over the many-body wavefunction, the Rabi oscillations of transitions between the single-molecule levels driven on resonance were seen to damp out exponentially with an emergent timescale. This effect was studied for a variety of Hamiltonian parameters near half filling, but a more thorough study of how the emergent timescale depends on the experimental parameters is lacking. Such a study may also shed light on the true microscopic mechanism of quantum dephasing. Is it governed only by equilibrium properties, as suggested by the Kibble-Zurek hypothesis [12–16]? What is the nature of the state as $t \to \infty$? A final interesting avenue for further research is to study the interplay between the complex many-molecule behavior of quantum dephasing with the complex behavior of a single molecule in the presence of non-collinear electric and magnetic fields.

Molecular Hubbard Hamiltonians may also be derived and studied for more complex molecules using the procedures outlined in this thesis. Near-term extensions include $^2\Sigma$ molecules formed from alkali metals and alkaline earths. These molecules have an unpaired spin giving rise to magnetic dipole moments in addition to electric dipole moments, which may lead to a rich interplay of crystalline and magnetic phases. Longer-term extensions can include molecules in Π or Δ states which have orbital angular momenta. Essentially nothing is known about the possibilities for many-body physics with such molecules.

Part III of this thesis laid the foundations for the Fermi resonance Hamiltonian (FRH) [17]. The FRH is a lattice projection of a two-channel model for a Feshbach resonance between two-component fermions and a bound molecular state. The two-body bound states in the lattice are nontrivial combinations of all Bloch bands from both the open and closed channels. Using a projection method to separate the two-particle Hilbert space into a low energy piece spanned by open channel fermions in the lowest band and a high energy piece spanned by open channel fermions in excited bands and all bands of the closed channel, the relevant high-energy degrees of freedom at low density are identified. The derivation of the FRH consists of re-coupling the low-energy and high energy sectors of Hilbert space at the many-body level, resulting in a multi-channel resonance model between fermions in the lowest open channel band and dressed molecules which are nontrivial linear combinations of an infinite number of bands chosen to reproduce the two-body scattering length exactly. The use of the numerically exact lattice solution of the two-body problem leads to novel features of the Hamiltonian, such as diagonal hopping of dressed molecules.

The physics related to the FRH is very much in its infancy. The most immediate question is, What are the many-body features of the FRH? How is the phase diagram affected by diagonal hopping and other features arising from the lattice solution? How do the Hubbard parameters for the FRH behave in confined geometries or geometries which are not separable for a single particle? Furthermore, one can imagine extending the FRH analysis to pairing in higher relative orbital angular momentum, multichannel situations, mass-imbalanced systems, or more realistic interchannel potentials. A final practical application of the FRH would be to optimize production of Feshbach molecules directly in an optical lattice using the many-body properties of the FRH.

In Parts IV and V of this thesis we devised variational matrix product state (MPS) algorithms for eigenstates and dynamics of generic time-dependent 1D Hamiltonians [18, 19], discussed open source software efforts for MPS algorithms [20], and presented educational materials to aid in the use of the open source implementations to study strongly correlated physics. Our stand-alone implementation of time-evolving block decimation (TEBD), open source TEBD [21], and the TEBD code written for the algorithms and libraries for physics simulations (ALPS) [20, 22, 23] package are both in use by many groups across the world [20, 24–36]. ALPS is also regularly used as a pedagogical tool for learning many-body physics at summer schools. At present, open source implementations of the generic algorithms presented in this thesis are being prepared for merger into the ALPS open source software package in its next release. This will greatly extend the capability of the ALPS package to study the statics and dynamics of a wider range of models, including long-range interactions and general time-dependence. In order to make these tools useful, we also are writing new educational materials to demonstrate the use of these algorithms and their enhanced capabilities over other MPS algorithms for time evolution.

MPS algorithms are very powerful in that they do not depend on the nature of the microscopic constituents of the model under study. Also, MPS methods produce wavefunctions, allowing for the simulation of dynamics and access to a broad range of quantum measures. The flexibility of MPS algorithms makes them especially well suited both to open source implementation and studies of many-body physics with ultracold molecules. Speaking broadly, as more complex molecules approach quantum degeneracy, the number of many-body models and their complexity will also dramatically increase. Searching for emergent phenomena such as quantum order in strongly correlated systems requires exploration of large parameter regimes and careful finite-size or finite-entanglement scaling. Open source tools for modern strongly correlated physics must be adaptable to different physical degrees of freedom, interactions, and dynamical processes. Furthermore, they should support massive parallelization over parameter regimes, and be efficient enough to handle large-scale parameter exploration. Finally, they must be able to calculate a broad range of quantum measures, and contain powerful post-processing tools to extract and manipulate data from large simulations. Many open questions remain regarding MPSs in general. Is it possible to simulate the dynamics of a generic time-dependent infinite system using Krylov methods as outlined for finite systems? How do we improve the numerical conditioning of the higher-dimensional tensor network algorithms so as to make them practical? What is the relevant operator structure, analogous to MPOs in 1D, for higher dimensions? The answer to these questions would provide a "holy grail" of strongly correlated physics: a black box numerical method applicable to any many-body system.

12.1 References Cited

[1] M. L. Wall and L. D. Carr. Emergent timescales in entangled quantum dynamics of ultracold molecules in optical lattices. *New J. Phys.*, 11:055027, 2009.

[2] M. L. Wall and L. D. Carr. Hyperfine molecular Hubbard Hamiltonian. *Phys. Rev. A*, 82(1):013611, Jul 2010. doi: 10.1103/PhysRevA.82.013611.

[3] Till D. Kühner, Steven R. White, and H. Monien. One-dimensional Bose-Hubbard model with nearest-neighbor interaction. *Phys. Rev. B*, 61(18):12474–12489, May 2000. doi: 10.1103/PhysRevB.61.12474.

[4] M. Dalmonte, G. Pupillo, and P. Zoller. One-Dimensional Quantum Liquids with Power-Law Interactions: The Luttinger Staircase. *Phys. Rev. Lett.*, 105:140401, Sep 2010. doi: 10.1103/PhysRevLett.105.140401. URL http://link.aps.org/doi/10.1103/PhysRevLett.105.140401.

[5] B. Capogrosso-Sansone, C. Trefzger, M. Lewenstein, P. Zoller, and G. Pupillo. Quantum Phases of Cold Polar Molecules in 2D Optical Lattices. *Phys. Rev. Lett.*, 104:125301, 2010.

[6] Alexey V. Gorshkov, Salvatore R. Manmana, Gang Chen, Jun Ye, Eugene Demler, Mikhail D. Lukin, and Ana Maria Rey. Tunable Superfluidity and Quantum Magnetism with Ultracold Polar Molecules. *Phys. Rev. Lett.*, 107:115301, Sep 2011. doi: 10.1103/PhysRevLett.107.115301. URL http://link.aps.org/doi/10.1103/PhysRevLett.107.115301.

[7] Alexey V. Gorshkov, Salvatore R. Manmana, Gang Chen, Eugene Demler, Mikhail D. Lukin, and Ana Maria Rey. Quantum magnetism with polar alkali-metal dimers. *Phys. Rev. A*, 84:033619, Sep 2011. doi: 10.1103/PhysRevA.84.033619. URL `http://link.aps.org/doi/10.1103/PhysRevA.84.033619`.

[8] Matthew S. Block, Ryan V. Mishmash, Ribhu K. Kaul, D. N. Sheng, Olexei I. Motrunich, and Matthew P. A. Fisher. Exotic Gapless Mott insulators of Bosons on Multileg Ladders. *Phys. Rev. Lett.*, 106:046402, Jan 2011. doi: 10.1103/PhysRevLett.106.046402. URL `http://link.aps.org/doi/10.1103/PhysRevLett.106.046402`.

[9] Ryan V. Mishmash, Matthew S. Block, Ribhu K. Kaul, D. N. Sheng, Olexei I. Motrunich, and Matthew P. A. Fisher. Bose metals and insulators on multileg ladders with ring exchange. *Phys. Rev. B*, 84:245127, Dec 2011. doi: 10.1103/PhysRevB.84.245127. URL `http://link.aps.org/doi/10.1103/PhysRevB.84.245127`.

[10] Hong-Chen Jiang, Matthew S. Block, Ryan V. Mishmash, James R. Garrison, D. N. Sheng, Olexei I. Motrunich, and Matthew P. A. Fisher. Non-Fermi liquid d-wave metal phase of strongly interacting electrons. *http://arxiv.org/abs/1207.6608*, 2012.

[11] M. L. Wall, E. Bekaroglu, and L. D. Carr. Molecular Hubbard Hamiltonian: Field Regimes and Molecular Species, 2012. In preparation.

[12] T. W. B. Kibble. Topology of cosmic domains and strings. *J. Phys. A: Math Gen.*, 9:1387, 1980.

[13] W. H. Zurek. Cosmological experiments in superfluid Helium? *Nature*, 317:505–508, 1985.

[14] T. Kibble. Phase-transition dynamics in the lab and the universe. *Phys. Today*, 60(9):47, 2007.

[15] Bogdan Damski and Wojciech H. Zurek. Dynamics of a Quantum Phase Transition in a Ferromagnetic Bose-Einstein Condensate. *Phys. Rev. Lett.*, 99:130402, 2007.

[16] Hiroki Saito, Yuki Kawaguchi, and Masahito Ueda. Kibble-Zurek mechanism in a quenched ferromagnetic Bose-Einstein condensate. *Phys. Rev. A*, 76:043613, 2007.

[17] M. L. Wall and L. D. Carr. Microscopic Model for Feshbach Interacting Fermions in an Optical Lattice with Arbitrary Scattering Length and Resonance Width. *Phys. Rev. Lett.*, 109:055302, Jul 2012. doi: 10.1103/PhysRevLett.109.055302. URL http://link.aps.org/doi/10.1103/PhysRevLett.109.055302.

[18] Michael L. Wall and Lincoln D. Carr. Finite temperature matrix product state algorithms and applications. *"Chapter in "Quantum Gases: Finite Temperature and Non-Equilibrium Dynamics" (Vol. 1 Cold Atoms Series), N. P. Proukakis, S. A. Gardiner, M. J. Davis and M. H. Szymanska, eds. (Imperial College Press, 2012); e-print http://arxiv.org/abs/1008.4303"*, 2011.

[19] M. L. Wall and L. D. Carr. Out of equilibrium dynamics with matrix product states. *New J. Phys. under review, arXiv:1205.1020v1*, 2012.

[20] B. Bauer, L. D. Carr, H. G. Evertz, A. Feiguin, J. Freire, S. Fuchs, L. Gamper, J. Gukelberger, E. Gull, S. Guertler, A. Hehn, R. Igarashi, S. V. Isakov, D. Koop, P. N. Ma, P. Mates, H. Matsuo, O. Parcollet, G. Pawłowski, J. D. Picon, L. Pollet, E. Santos, V. W. Scarola, U. Schollwöck, C. Silva, B. Surer, S. Todo, S. Trebst, M. Troyer, M. L. Wall, P. Werner, and S. Wessel. The ALPS project release 2.0: open source software for strongly correlated systems. *Journal of Statistical Mechanics: Theory and Experiment*, 2011(05):P05001, 2011. URL http://stacks.iop.org/1742-5468/2011/i=05/a=P05001.

[21] Open source time-evolving block decimation. http://physics.mines.edu/downloads/software/tebd/.

[22] ALPS. http://alps.comp-phys.org.

[23] A. F. Albuquerque, F. Alet, P. Corboz, P. Dayal, A. Feiguin, S. Fuchs, L. Gamper, E. Gull, S. Gurtler, A. Honecker, R. Igarashi, M. Korner, A. Kozhevnikov, A. Lauchli, S. R. Manmana, M. Matsumoto, I. P. McCulloch, F. Michel, R. M. Noack, G. Pawlowski, L. Pollet, T. Pruschke, U. Schollwock, S. Todo, S. Trebst, M. Troyer, P. Werner, and S. Wessel. The ALPS project release 1.3: Open-source software for strongly correlated systems. *Journal of Magnetism and Magnetic Materials*, 310:1187–1193, 2007.

[24] M. L. Wall and L. D. Carr. Emergent timescales in entangled quantum dynamics of ultracold molecules in optical lattices. *New J. Phys.*, 11:055027, 2009.

[25] L. D. Carr and M. K. Oberthaler. Spatial dependence of entropy in trapped ultracold Bose gases. *e-print arXiv:0906.4708*, 2009.

[26] R. V. Mishmash and L. D. Carr. Quantum Entangled Dark Solitons formed by Ultracold Atoms in Optical Lattices. *Phys. Rev. Lett.*, 103:140403, 2009.

[27] R. V. Mishmash, I. Danshita, Charles W. Clark, and L. D. Carr. Quantum Many-Body Dynamics of Dark Solitons in Optical Lattices. *Phys. Rev. A*, 80: 053612, 2009.

[28] L. D. Carr, M. L. Wall, D. G. Schirmer, R. C. Brown, J. E. Williams, and C. W. Clark. Mesoscopic Effects in Quantum Phases of Ultracold Quantum Gases in Optical Lattices. *Phys. Rev. A*, 81:013613, 2010.

[29] M. L. Wall and L. D. Carr. Hyperfine molecular Hubbard Hamiltonian. *Phys. Rev. A*, 82:013611, 2010.

[30] J. A. Glick and L. D. Carr. Macroscopic quantum tunneling of solitons in Bose-Einstein condensates. *Phys. Rev. Lett. under review, arXiv:1105.5164*, 2011.

[31] Giuseppe Carleo, Frederico Becca, Marco Schiró, and Michele Fabrizo. Localization and Glassy Dynamics Of Many-Body Quantum Systems. *Sci. Rep.*, 2(243), 2011.

[32] Yinyin Qian, Ming Gong, and Chuanwei Zhang. Quantum transport of bosonic cold atoms in double-well optical lattices. *Phys. Rev. A*, 84:013608, Jul 2011. doi: 10.1103/PhysRevA.84.013608. URL http://link.aps.org/doi/10.1103/PhysRevA.84.013608.

[33] Zheng-Wei Zhou, Shao-Liang Zhang, Xiang-Fa Zhou, Guang-Can Guo, Xingxiang Zhou, and Han Pu. Quantum phase transition of Bose-Einstein condensates on a nonlinear ring lattice. *Phys. Rev. A*, 83:043626, Apr 2011. doi: 10.1103/PhysRevA.83.043626. URL http://link.aps.org/doi/10.1103/PhysRevA.83.043626.

[34] Hong Lu, L. O. Baksmaty, C. J. Bolech, and Han Pu. Expansion of 1D Polarized Superfluids: The Fulde-Ferrell-Larkin-Ovchinnikov State Reveals Itself. *Phys. Rev. Lett.*, 108:225302, May 2012. doi: 10.1103/PhysRevLett.108.225302. URL http://link.aps.org/doi/10.1103/PhysRevLett.108.225302.

[35] Yinyin Qian, Ming Gong, and Chuanwei Zhang. Many-body Landau-Zener Transition in Cold Atom Double Well Optical Lattices. *http://arxiv.org/abs/1110.1653*, 2012.

[36] Shuai Yin, Xizhou Qin, Chaohong Lee, and Fan Zhong. Finite-time scaling of dynamic quantum criticality. *http://arxiv.org/abs/1207.1602*, 2012.

APPENDIX A - OVERVIEW OF SOURCE CODE CD CONTENTS

A.1 Codes Associated with Open Source Projects and Educational Materials

The open source time-evolving block decimation (OSTEBD) project was overviewed in Secs. 10.1 and 10.1.1, and a user's guide containing a thorough exposition of its use and operation is given as Appendix B. Hence, we will not discuss OSTEBD further in this appendix. The most recent release of the code, version 2.0, is included as the file `OpenSourceTEBD_v2.0.tar.gz`.

The time-evolving block decimation (TEBD) code which was included with the algorithms and libraries for physics simulations (ALPS) open source package was discussed in Secs. 10.2-10.2.3, and so we will not discuss it further here. The source code of the most recent major release, version 2.1, is included as the file `alps-2.1.1-r6176-src.tar.gz`. The TEBD routines may be found in the `alps/applications/dmrg/tebd` directory, and the tutorial files are located in the subdirectories of `alps/tutorials/` which start with `tebd-`. ALPS uses a CMake [1]-based build system, and binary installers are available for many popular operating systems. Setup and installation instructions are included on the ALPS installation page and as part of Ref. [2].

The code `SimpleMPS` was circulated with the document "Introduction to MPS Algorithms" discussed in Sec. 11.2 and reprinted as Appendix E. This code is a stripped-down version of the matrix product state (MPS) algorithms proposed in Chapter 7. As it is intended for pedagogy, this implementation focuses on clarity rather than optimization. The main file `IsingGSMain.f90` computes the ground state energy for the Ising model in a transverse field with user-specified parameters. This file is intended to provide a black box with which students can check the results of their own implementations. Additionally, the interfaces of the procedures in the core of this code are the same as those used in the educational document so that a student can consult the source code for implementation hints while debugging.

A.2 Code for the Fermi Resonance Hamiltonian

The code used to generate the data in Part III is included as the file `FRH_Code.tar.gz`. The ultimate outputs of the code are the Hubbard parameters defining the Fermi resonance Hamiltonian (FRH), Eq. (5.6), for a given s-wave scattering length a_s and isotropic lattice height V. This computation is performed in 3 stages:

1. The exact bound states at the extremal points of the Brillouin zone (BZ) with total quasimomentum $\mathbf{K} = \Gamma$, X, M, and R[92] are computed. This is done by specifying a range of bound state energies $E_{\mathbf{K}}$ and solving the nonlinear eigenvalue problem in E, Eq. (5.1), as a linear eigenvalue problem for a_s with $E_{\mathbf{K}}$ fixed. These solutions, along with their associated matrices $\chi^{\mathbf{K}}_{\mathbf{st}}(E_{\mathbf{K}})$, are written to the disk for later use. `DataParallelInitializeMain.f90` performs this operation in parallel over bound state energies.

2. The bound states from (1) are sorted according to their parity, which is always possible at the extremal points of the BZ. A fixed s-wave scattering length a_s is specified, and the results of (1) are used to find the bound states of a given parity at fixed a_s in the entire BZ. The bound states are found via a Newton-Raphson iteration [3], starting from the point computed in (1) which is nearest the desired a_s. Due to the parity symmetry, only the bound states in a single octant of the cubic BZ are computed and the others can be generated by symmetry operations. `DataParallelNRMain.f90` performs these tasks in parallel over the total quasimomentum $\mathbf{K}$.

3. The bound states at fixed a_s calculated in (2) are used to compute the Hubbard parameters appearing in Eq. (5.6). `HubbMain.f90` performs this computation.

---

[92] See the caption of Figure 5.2 for the definitions of the irreducible BZ points.

This part of the calculation is not computationally intensive, and so is done in serial.

Implementation and optimization details of the computation will appear in an upcoming paper [4].

A.3 Code for Variational Matrix Product State Simulations

In this section we discuss the author's implementations of the MPS algorithms discussed in this part of the thesis. The numerical routines are written in Fortran 90, with a Python [5] front end named `MPSPyLib` similar to the front end for the ALPS routines, see Sec. 10.2.2. The local Hilbert space, operators, observables, and other simulation metadata are generated using Python. The front end then sends this data to a Fortran back end which performs the intensive variational MPS computations. The output is then read in and post-processed by the Python front end. A Python front end is also used for the code to compute molecular Hubbard parameters, see Appendix A.4. The MPS codes discussed here are found in `MPS_Code.tar.gz`.

A.3.1 Generic Fortran Programming with the Python Preprocessor

Generic programming is implemented in Fortran by using type placeholders in template routines, and then using Python preprocessing scripts to build Fortran modules for user-defined types. This functionality is provided by the `regex` Python module which performs operations with regular expressions. As an example of the templating procedure, consider the subroutine below:

```
SUBROUTINE Project_QTENSOR_TYPE(B,A,psiProjs)
!Purpose: Apply the projector 1-\sum_{\alpha}|psiProjs_{\alpha}><
    psiProjs_{\alpha}| to A and store in B.
! Used in orthogonalizing an MPS against another set of MPSs
IMPLICIT NONE
TYPE(QTENSOR_TYPE) :: A,B
TYPE(QTENSOR_TYPE), POINTER :: psiProjs(:)
INTEGER :: j
REAL_OR_COMPLEX(KIND=rKind) :: proj
```

```
CALL Copy(B,A)
DO j=1,SIZE(psiProjs)
     proj=-Dot(psiProjs(j),A)
    CALL GAXPY(B,proj,psiProjs(j))
END DO
END SUBROUTINE Project_QTENSOR_TYPE
```

Here, a `QTENSOR_TYPE A` is orthogonalized against a set of `QTENSOR_TYPE`s stored in `psiProjs`. All that is required is that the `QTENSOR_TYPE` have the `Copy`, `Dot`, and `GAXPY`[93] capabilities. The Python preprocessor replaces `QTENSOR_TYPE` with the actual types used by the code and places a generic interface to this procedure with other similar procedures into a Fortran module. This methodology prevents the copying of code, making the project easier to maintain. The preprocessing takes place automatically as part of a Python-based build system run by the file `BuildMPSLib.py`.

At the time of the writing of this thesis, the author's MPS code consists of 12433 lines of Fortran, and 3404 lines of Python. The templating procedures discussed above make this code smaller than the Open Source TEBD, see Sec. 10.1, for example. For direct comparison, the Fortran code after preprocessing the template files is 23277 lines. While no manual for this code exists at this time, `MPSPyLib` is documented using the docstring capability of Python. Thus, documentation for any code object `p` in the Python libraries is provided by calling `help(p)` in the Python interpreter.

A.3.2 Defining Local Hilbert Spaces: the `Ops` Module

The on-site Hilbert space is specified using the `Ops.py` module, which produces operators as Python dictionaries which map strings to matrix representations of the operators. For example, consider the code

```
import MPSPyLib as mps

Operators=mps.BuildSpinOperators(spin=0.5)
print Operators['sz']
```

---

[93] Generalized A times X plus Y, $x = a * x + y$

Now `Operators` contains a complete set of spin-1/2 operators. These are accessed as e.g. `Operators['sz']`, which produces $\hat{S}^z$ in matrix representation. Explicitly, the output from the above code is

```
[[ 0.5  0. ]
 [ 0.  -0.5]]
```

Convenience functions exist to build operators for spins, bosons and fermions with arbitrary internal degrees of freedom, and combinations of bosons and fermions with internal degrees of freedom. User defined operators can be defined simply by naming a new key in the dictionary holding the operators. For example, to generate the Bose-Hubbard interaction operator $\frac{1}{2}\hat{n}_i\left(\hat{n}_i-1\right)$ from the boson number operator `nbtotal`, we could perform

```
import numpy as np
Operators['Interaction']=0.5*(np.dot(Operators['nbtotal'],
    Operators['nbtotal'])-Operators['nbtotal'])
```

Here `np.dot` is matrix multiplication implemented in the numerical Python package [6]. In addition to building operators, `Ops.py` can also transform them into covariant form appropriate for conserving Abelian symmetries[94] using the `OperatorstoQOperators` procedure. Given a set of operators as a Python dictionary and a set of diagonal generators of the Abelian symmetries as a list of matrices, the routine transforms the operators into their canonical forms Eq. (6.87)[95] in which nonzero elements are labeled by their incoming and outgoing charges expressed as integers and sorted in decreasing order. As an example, consider

```
import MPSPyLib as mps

Operators=mps.BuildSpinOperators(spin=0.5)
QOperators=mps.OperatorstoQOperators(Operators['sz'],Operators)
```

---

[94] See Sec. 6.6 for the general theory of tensors which are covariant under the action of Abelian symmetries.

[95] All operators are assumed to be covariant, with an exception thrown if this is not the case.

```
print 'Sz:'
for element in QOperators['sz']:
        print element, QOperators['sz'][element]
print 'SPlus:'
for element in QOperators['splus']:
        print element, QOperators['splus'][element]
```

This will convert the spin operators into a form appropriate for conserving the total magnetization. The output from this code is

```
Sz:
q [[(1,), (1,)], [(0,), (0,)]]
Blocks [array([[ 0.5]]), array([[-0.5]])]
SPlus:
q [[(1,), (0,)]]
Blocks [array([[ 1.]])]
```

`q` gives the incoming and outgoing quantum numbers for each irrep, and `Blocks` gives the matrix representations in the degeneracy spaces. For $\hat{S}^z$, the matrix is diagonal in the quantum numbers, but for $\hat{S}^+$, the matrix takes quantum number 0 (corresponding to spin down, as can be seen from $\hat{S}^z$) to quantum number 1 (spin up).[96] Because all on-site irreps are one-dimensional in this case, the blocks are all 1×1 matrices.

A.3.3 Defining Hamiltonians: the `MPO` Module

The `MPO.py` module builds MPO representations of operators using a set of pre-defined finite state automaton rules as outlined in Chapter 7. In particular, the rules are

- `site` rule. Example :

```
H.AddMPOTerm(Operators,'site','sz',hparam='h_z',weight=1.0)
```

---

[96] One can compare this form of $\hat{S}^z$ to the invariant form of $\hat{n}$ in Eq. (6.89). Also, compare $\hat{S}^+$ to $\hat{b}$ given in Eq. (6.91).

generates $\hat{H} = h_z \sum_{i=1}^{L} \hat{S}_i^z$, where $\hat{S}^z$=`Operators['sz']`. In this and the other examples, the first argument of `AddMPOTerm` is a Python dictionary of operators and the third argument contains the relevant names of the operators to be used. Because only the names of the operators are used, this routine handles operators and q-operators on the same footing.

- `bond` rule. Examples:

```
H.AddMPOTerm(Operators,'bond',['splus','sminus'],hparam='J_xy
    ',weight=0.5)
```

generates $\hat{H} = \frac{J_{xy}}{2} \sum_{\langle i,j \rangle} \left(\hat{S}_i^+ \hat{S}_j^- + \text{h.c.} \right)$, where $\langle i,j \rangle$ denotes all nearest neighbor pairs i and j,

```
H.AddMPOTerm(Operators,'bond',['sz','sz'],hparam='J_z',weight
    =1.0)
```

generates $\hat{H} = J_z \sum_{\langle i,j \rangle} \hat{S}_i^z \hat{S}_j^z$, and

```
H.AddMPOTerm(Operators,'bond',['fdagger','f'],hparam='t',
    weight=-1.0,Phase=True)
```

generates $\hat{H} = -t \sum_{\langle i,j \rangle} \left(\hat{f}_i^\dagger \hat{f}_j + \text{h.c.} \right)$, where $\hat{f}$ is a fermionic destruction operator. That is, terms with fermi phases are cared for with the optional `Phase` argument and Hermiticity is automatically enforced by this routine.

- `product` rule. Example:

```
H.AddMPOTerm(Operators,'product','sz',hparam='h_p',weight
    =-1.0)
```

generates $\hat{H} = -h_p \prod_{i=1}^{L} \hat{S}_i^z$. The operator used in the product must be Hermitian.

- `FiniteFunction` rule. Example:

```
f=[]
for i in range(6):
    f.append(1.0/(i+1.0)**3)
H.AddMPOTerm(Operators,'FiniteFunction',['sz','sz'],f=f,
    hparam='J_z',weight=-1.0)
```

generates $\hat{H} = -J_z \sum_{1\le i,j\le 6} \frac{1}{|j-i|^3} \hat{S}_i^z \hat{S}_j^z$, where the summation is over all i and j pairs separated by at least 1 and at most 6 lattice spacings. The range of the potential is given by the number of elements in the list `f`. This rule enforces Hermiticity and cares for Fermi phases as in the bond rule case.

- `exponential` rule

```
H.AddMPOTerm(Operators,'exponential',['sz','sz'],hparam='J_z'
    ,decayparam=a, weight=1.0)
```

generates $\hat{H} = J_z \sum_{i<j} a^{j-i+1} \hat{S}_i^z \hat{S}_j^z$. This rule also enforces Hermiticity and cares for Fermi phases as in the bond rule case.

- `InfiniteFunction` rule. Examples

```
invrcube = lambda x: 1.0/(x*x*x)
H.AddMPOTerm(Operators,'InfiniteFunction',['sz','sz'],hparam=
    'J_z',weight=1.0,func=invrcube,L=1000,tol=1e-9)
```

generates $\hat{H} = \sum_{i<j} \frac{1}{|j-i|^3} \hat{S}_i^z \hat{S}_j^z$, where the functional form is valid to a distance `L` with a residual of at most `tol`. Similarly,

```
H.AddMPOTerm(Operators,'InfiniteFunction',['sz','sz'],hparam=
    'J_z',weight=1.0,x=x,y=f,tol=1e-9)
```

generates $\hat{H} = \sum_{i<j} f(j-i) \hat{S}_i^z \hat{S}_j^z$, where the functional form $f(x)$ is determined by the array of values `f` and the evaluation points `x`. The distance of validity `L` is determined from the array `x`. This rule applies only to monotonically decreasing functions `func` or `f`, and so outside the range of validity `L` the corrections are also monotonically decreasing. This rule enforces Hermiticity and cares for Fermi phases as in the bond rule case.

In all of the above expressions, `H` is an object of the `MPO` class. The strings passed to the argument `hparams` represent tags for variable Hamiltonian parameters. Hence, a template MPO for the hard-core boson model with $1/r^3$ interactions,

$$\hat{H} = -t \sum_{\langle i,j \rangle} \left(\hat{b}_i^{\dagger} \hat{b}_j + \text{h.c.} \right) + U \sum_{i<j} \frac{\hat{n}_i \hat{n}_j}{|j-i|^3} - \mu \sum_i \hat{n}_i \,, \tag{A.1}$$

is provided by

```
import MPSPyLib as mps

#Build operators
Operators=mps.BuildBoseOperators(1)
#Define Hamiltonian MPO
H=mps.MPO()
H.AddMPOTerm(Operators,'site','nbtotal',hparam='mu',weight=-1.0)
H.AddMPOTerm(Operators,'bond',['bdagger','b'],hparam='t',weight
    =-1.0)
invrcube = lambda x: 1.0/(x*x*x)
H.AddMPOTerm(Operators,'InfiniteFunction',['nbtotal','nbtotal'],
    hparam='U',weight=1.0,func=invrcube,L=1000,tol=1e-9)
```

The actual numerical values of t, U, and μ are not specified until a particular simulation is defined, see Appendix A.3.5.

A.3.4 Defining Observables: the `Obs` Module

Observables are specified using the `Obs.py` module. The types of observables supported are `site` observables such as $\langle \hat{n}_i \rangle$, `corr` observables such as $\langle \hat{n}_i \hat{n}_j \rangle$, and general MPOs. Each observable is given a tag which is used to post-process it later. For example, to tag the computation of the single-particle density matrix $\langle \hat{b}_i^{\dagger} \hat{b}_j \rangle$ with `'spdm'`, one would call

```
myObservables.AddObservable(Operators,['bdagger','b'],'corr','
    spdm')
```

where `myObservables` is an object of the `Observables` class. For finite systems, `site` observables are measured at every position and returned as a vector and `corr` observables are stored as a (Hermitian) matrix whose indices run over the sites in the system. For infinite systems, `site` observables are measured within the unit cell and the function `SpecifyCorrelationRange` allows the user to specify the maximum distance of correlation functions. After a simulation has been run, the function `Outputs=ReadStaticObservables(staticsParameters,myObservables)` reads all of the observables specified in `myObservables` and computed by a simulation whose metadata is stored in `staticsParameters` and places them along with all simulation metadata and convergence information in `Outputs`. This can be compared with a similar capability given by `LoadTimeEvolution` in the ALPS Routines of Sec. 10.2.2.

A.3.5 Interfacing with the Fortran Back End: the `tools` Module

The `tools.py` module contains routines to write class data and other pertinent information in a format understood by the Fortran routines, as well as writing build scripts, compiling, and running simulations. A simulation is specified as a Python dictionary of metadata. As an example, consider a simulation of noninteracting spinless fermions given by the following code

```
import MPSPyLib as mps
import numpy as np

#Build operators
Operators=mps.BuildFermiOperators()
#Define Hamiltonian MPO
H=mps.MPO()
H.AddMPOTerm(Operators,'bond',['fdagger','f'],hparam='t',weight=-1.0,Phase=True)

#ground state observables
myObservables=mps.Observables()
#Site terms
myObservables.AddObservable(Operators, 'nftotal','site','n')
#correlation functions
myObservables.AddObservable(Operators,['nftotal','nftotal'],'corr','nn')
myObservables.AddObservable(Operators,['fdagger','f'],'corr','spdm',Phase=True)
```

```

t=1.0
L=10
N=5
#define statics
staticsParameters=[{
            }Directories
            'job_ID'                        : 'SpinlessFermions_',
            'unique_ID'                     : 'L_'+str(L)+'N'+str(N),
            'Write_Directory'               : 'TMP/',
            'Output_Directory'              : 'OUTPUTS/',
            #System size and Hamiltonian parameters
            'L'                             : L,
            't'                             : t,
            #Specification of symmetries and good quantum numbers
            'Abelian_generators'           : Operators['nftotal'],
            'Abelian_quantum_numbers'        : N,
            #Convergence parameters
            'MaxnLanczosIterations'         : 100,
            #IMPS
            'imps_tol'                      : 1E-12,
            'imps_bond_dimension'           : 170,
            #FMPS
            'bond_dimension'                : 500,
            'n_excited_states'              : 1,
            'variance_tol'                  : 1E-10,
            'lanczos_tol'                   : 1E-12,
            'max_num_Sweeps'                : 10,
            #Other parameters
            'verbose'                       : 0,
            'strict'                        : False,
            'simtype'                        : 'Finite'

          }]

#Write Fortran-readable main files
MainFiles=mps.WriteFiles(staticsParameters,Operators,H,
    myObservables,teParameters=None)
#Run the simulations
mps.runMPS(MainFiles)

#post-processing and plotting
Outputs=mps.ReadStaticObservables(staticsParameters,myObservables
    )
mps.PrintObservable(Outputs,'energy')
mps.PrintObservable(Outputs,'spdm')

```

```
A.62 mps.CleanFiles(staticsParameters)
```

Here, `staticsParameters` contains all of the simulation metadata, including a numerical value for the Hamiltonian parameter `t` specified in line 8. The parameters appearing are:

1. `'job_ID'`: a string identifying a group of simulations. This parameter is not optional.
2. `'unique_ID'`: a string identifying a particular simulation. This parameter is optional and defaults to an empty string. It is used for batching simulations to ensure that each simulation is distinguishable from the others.
3. `'Write_Directory'`: the directory where the Fortran-readable files are written. This parameter keeps the main directory from becoming cluttered. It is optional and defaults to the current directory.
4. `'Output_Directory'`: the directory where files are written from the Fortran code. It is optional and defaults to the current directory.
5. `'L'`: the number of sites for a finite simulation or the number of sites in the unit cell for an infinite simulation. This parameter is not optional.
6. `'bond_dimension'`: the maximum bond dimension allowed for static calculations. For finite simulations, this is an integer. For infinite simulations, this parameter is a list so that a finite-entanglement scaling may be performed. This parameter is not optional.
7. `'imps_bond_dimension'`: the maximum bond dimension allowed when using iMPS[97] to generate an initial ansatz for a finite-sized ground state search. This parameter is optional and defaults to one-third of `'bond_dimension'`.

---

[97] See Chapter 8 for a discussion of the iMPS algorithm.

8. `'n_excited_states'`: the number of excited states calculated. This optional parameter is only relevant for finite systems where eMPS[98] applies and defaults to 0.

9. `'variance_tol'`: the tolerance ϵ on the variance of the state for exit from variational state search, i. e. ,$\langle\left(\hat{H}-E\right)^2\rangle < L\epsilon$. This parameter is optional and defaults to machine precision. For infinite simulations, this controls the truncation error rather than the variance.

10. `'imps_tol'`: The tolerance on the truncation error used when applying iMPS to generate an initial ansatz for a finite-sized ground state search. This parameter is optional and defaults to the machine precision.

11. `'lanczos_tol'`: the tolerance ϵ for exiting the Lanczos iteration with either the norm of the k^{th} Lanczos vector less than ϵ or the residual $\left|\hat{H}A-\lambda A\right| < \epsilon$. This parameter is optional and defaults to the machine precision.

12. `'MaxnLanczosIterations'`: the maximum number of iterations allowed by the Lanczos algorithm when solving an effective Hamiltonian eigenproblem. This parameter is optional, and defaults to 100.

13. `max_num_Sweeps` : the maximum number of sweeps (complete optimization cycles over the lattice) allowed in the variational state search. This parameter is optional and defaults to 6.

14. `'verbose'`: denotes the level of verbosity of standard output from the Fortran code. 0 is no output, 1 is normal, and 2 is debug level of output. Optional, defaults to 0.

15. `'strict'`: a value of True prevents existing files with the same name from being overwritten. This parameter is optional and defaults to False.

---

[98] See Chapter 7 for a discussion of the eMPS algorithm.

16. `'Abelian_generators'`: the diagonal matrices whose elements are the on-site charges which are to be conserved by the simulation. Here, the presence of the on-site number operator `nftotal` specifies that we are conserving the total number of fermions.

17. `'Abelian_quantum_numbers'`: the values of the conserved charges associated with the operators in `'Abelian_generators'`.

18. `'simtype'`: specifies whether the simulation uses finite-size or infinite-size algorithms.

The order of the terms has no importance. This data is passed together with the MPO template for the Hamiltonian and the observables to `WriteFiles`, which writes all of the data pertinent to the simulations to Fortran-readable files. These files are then passed to `runMPS`, which runs the simulations. After simulations have been run, `ReadStaticObservables` is used as described above to extract the observable quantities from the outputs of the Fortran code. At the end of the simulation, `CleanFiles` is used to remove all intermediate files which are no longer needed.

A.3.6 Data-Parallelsim: the `Paralleltools` module

Data-parallel simulations may be prepared using the `Paralleltools.py` module. Here a set of tags and a set of iterators are passed along with a Python dictionary template which contains all of the other parameters relevant to the simulations. The routines write all of the necessary input files to the run the simulations, and additionally write a job pool file which as master node uses to delegate simulations to workers in parallel. A record of progress is kept so that simulations which are prematurely aborted can be restarted. For example, to loop over the parameters t and μ in parallel, we could use the code

```
mumin=0.5
```

```
mumax=1.8
muvals=50
muiter=np.linspace(mumin,mumax,muvals)

tmin=0.01
tmax=0.4
tvals=50
titer=np.linspace(tmin,tmax,tvals)

iterparams=['mu','t']
iters=[muiter,titer]

mps.WriteMPSParallelFiles(staticsParameterstemplate,comp_info,H,
    iterparams,iters,Operators,myObservables)
```

`staticsParameterstemplate` is a Python dictionary containing all other parameters relevant to the simulation as in the example above, `comp_info` is a Python dictionary containing metadata about the number of nodes, the desired time, etc. which are relevant for parallel environments, `H` is the MPO template, and `myObservables` is an observables object. The result of calling this procedure is a pbs script which will run the desired simulations in parallel.

A.3.7 Dynamics: the `Dynamics` module

Dynamics based on the general method discussed in Chapter 7 are supported in an alpha version at the time of the writing of this thesis, and are not included as part of the stable version on the source code CD. A dynamic process is specified as

```
Quenches=mps.QuenchList()
Quenches.AddQuench(hparamslist,tau,deltat,functionalforms)
```

`hparamslist` is a list containing the Hamiltonian parameters which change in time, `tau` is the length of the particular dynamical process, `deltat` is the time step, and `functionalforms` is a list of function handles describing how the parameters in `hparamslist` change. For example, to define a linear quench of the transverse field `hx` in the Ising model from $5J$ to 0 over a timescale $\tau = \hbar/J$, we would call

```
h0=5.0
```

```
hf=0.0
tau=1.0
hfunc= lambda t : h0+(hf-h0)*t/tau
Quenches=mps.QuenchList()
Quenches.AddQuench(['hx'],tau,0.1,[hfunc])
```

A.4 Code for Computation of Molecular Hubbard Parameters

Code to generate Hubbard parameters for the molecular Hubbard Hamiltonian using the methods of Chapter 2 is included as the file `MHHLib.tar.gz`. The computation of the Hubbard parameters proceeds in three stages. In the first stage, we solve the internal Hamiltonian including static external fields. This Hamiltonian is discussed in Appendices 4.5 and 4.6 of Chapter 4, with explicit matrix elements given in Appendix 4.7. For strong DC electric fields, several rotational levels are required to accurately converge the results, see Sec. 3.7. The large number of rotational states together with the hyperfine spin states can make the linear dimension of the internal Hamiltonian a few thousands, and so pre-computation of matrix elements and sparse diagonalization routines [7] are used to find the eigenstates which correlate to the lowest two rotational manifolds.

Once a basis for the internal Hamiltonian has been found, the dipole moments, polarizability tensor, and other operators relevant to the calculation are transformed to this basis. The complete single-particle Hamiltonian is now diagonalized to yield the single-particle eigenstates in Bloch form, see Sec. 2.1. For the lowest rotational level, the polarizability tensor is diagonal to a relative precision of 10^{-9} in most cases of interest, and so may be computed using the methods of Sec. 2.1 which assume the lattice has no matrix elements connecting different eigenstates of the internal Hamiltonian. For internal states which correlate to rotational angular momentum $N = 1$ and rotational projection $M_N = \pm 1$ along the DC electric field axis in zero field, the polarizability is not diagonal in the internal degrees of freedom, and so a more sophisticated procedure is required. This procedure will be elucidated in a forthcoming paper [8]. In the final stage, the Bloch functions are transformed to Wannier functions and used to compute Hubbard parameters as in Eqs. (2.26), (2.34), and (2.39). The ranges of the Hubbard parameters are truncated according to a user-supplied energetic cutoff.

As with the MPS codes discussed in Appendix A.3, this code uses a Python front end to define, run, and post-process simulations. This makes integration of the Hubbard parameters and execution of many-body simulations of the resulting molecular Hubbard Hamiltonians using the MPS code straightforward. The relevant experimental parameters are provided as input using a Python dictionary in a similar manner to the MPS code. Example main files are provided in `MHHLib.tar.gz`. Finally, the computation of Hubbard parameters for a range of experimental controls, for example the strength of the DC or AC electric fields, may be parallelized over using a list of iterators as discussed for the MPS code in Appendix A.3.6.

A.5 References Cited

[1] Cross-platform make http://www.cmake.org/.

[2] B. Bauer, L. D. Carr, H. G. Evertz, A. Feiguin, J. Freire, S. Fuchs, L. Gamper, J. Gukelberger, E. Gull, S. Guertler, A. Hehn, R. Igarashi, S. V. Isakov, D. Koop, P. N. Ma, P. Mates, H. Matsuo, O. Parcollet, G. Pawłowski, J. D. Picon, L. Pollet, E. Santos, V. W. Scarola, U. Schollwöck, C. Silva, B. Surer, S. Todo, S. Trebst, M. Troyer, M. L. Wall, P. Werner, and S. Wessel. The ALPS project release 2.0: open source software for strongly correlated systems. *Journal of Statistical Mechanics: Theory and Experiment*, 2011(05):P05001, 2011. URL http://stacks.iop.org/1742-5468/2011/i=05/a=P05001.

[3] P. Lancaster, *Lambda-matrices and Vibrating Systems,* (Pergammon Press, Oxford, 1966).

[4] . M. L. Wall and L. D. Carr, The Fermi resonance Hamiltonian: an effective model for strongly interacting fermions in an optical lattice, Phys. Rev. A to be submitted (2012).

[5] http://www.python.org/.

[6] http://numpy.scipy.org/.

[7] ARPACK package for sparse solution of eigenvalue problems: http://www.caam.rice.edu/software/ARPACK/.

[8] . M. L. Wall and L. D. Carr, In preparation.

APPENDIX B - OPEN SOURCE TEBD_V2.0 USER'S GUIDE

This appendix is the User's guide associated with the v2.0 release of the Open Source TEBD package with some slight updates. This was the last release of Open Source TEBD before the routines were merged into the ALPS collaboration.

B.1 What's New in v2.0

The v2.0 release of Open Source TEBD is a beta version supported by the Carr Theoretical Physics Research Group which contains many improvements over the v1.0 alpha release. In decreasing order of importance, the improvements are

1. a fifth order Trotter expansion of the propagator, discussed in Sec. B.3.5.2,
2. support for systems with periodic boundary conditions, discussed in Sec. B.3.5.3,
3. support for fermionic systems, discussed in Sec. B.3.6.3,
4. new routines for constructing the propagator based on the boundary conditions and the order of Trotter expansion, discussed in Sec. B.9.1.24,
5. a compact structure for calculating and storing observables, discussed in Sec. B.3.6.6,
6. a more efficient means of computing nonlocal observables which are tensor products, discussed in Sec. B.3.6.4,
7. the usage of NAMELIST i/o in all case studies to avoid the need to recompile when parameters are changed,
8. several small changes related to code readability and performance,
9. correction of typographical errors and updating of material in this document.

B.2 Introduction

The development of efficient classical algorithms to simulate quantum many-body phenomena is a very active area of contemporary theoretical physics research. The challenge of simulation lies in the exponential growth of the full many-body Hilbert space with the size of the system. For example, in the Bose-Hubbard model, which accurately describes the behavior of ultracold alkali atoms in deep optical lattices [1], the dimension of a Hilbert space describing N bosons distributed among L lattice sites is

$$\Omega_{\mathrm{BH}} = \frac{(L+N-1)!}{L!\,(N-1)!}. \tag{B.1}$$

Typical experimental systems are on the order of hundreds of lattice sites with comparable numbers of particles, making exact diagonalization of the full many-body Hamiltonian intractable even with sparse matrix methods. Also, the introduction of an internal Hilbert space (e.g. spin or rotational degrees of freedom) will make the Hilbert space even larger, furthering the need for efficient algorithms.

Great progress has been made in the past decade in numerical studies of the quantum many-body problem. In particular, the density matrix renormalization group (DMRG) technique proposed by White [2, 3] has been used to great success in a variety of 1-dimensional (1D) systems with so-called "limited entanglement" (see [4] for a review). The initial limitation of DMRG is that it was a static method–it could predict the ground state of a given Hamiltonian but could not simulate time evolution efficiently. After many proposed extensions of DMRG with varying degrees of success, Vidal was able to produce a successful time evolution method [5, 6] which we now refer to as the time-evolving block decimation (TEBD) algorithm. With TEBD, one can study the dynamics of quantum many-body systems efficiently, and also find ground states through propagation in imaginary time. In the remainder of this document we will provide a thorough discussion of the theory and implementation of TEBD. The heart of this manual is a collection of manpages which describe the syntax and uses of all routines included in the package. This part of the manual begins with Sec. B.7.

B.3 The Time-Evolving Block Decimation Algorithm

In this section we will provide a detailed description of the Time-Evolving Block Decimation algorithm (TEBD) with a focus on theoretical and conceptual aspects.

B.3.1 Schmidt Decomposition and Related Theorems

Theorem: (Singular value decomposition) Let A be an $m \times n$ complex matrix. Then there exists an $m \times m$ unitary matrix Γ_A, an $n \times n$ unitary matrix Γ_B, and an $m \times n$ positive diagonal (as defined for rectangular matrices) matrix λ such that

$$A = \Gamma_A \lambda \Gamma_B^{\dagger} . \tag{B.2}$$

This is referred to as the *singular value decomposition* of A, and the diagonal elements of λ are referred to as the *singular values* of A. Note that the form of the decomposition implies that A has at least one and at most $\min(m, n)$ distinct singular values. The number of nonzero singular values of a matrix is its *rank*.

The most important property of the SVD for our purposes is that the matrix $A^{(l)}$ defined by the matrix elements

$$A_{ij}^{(l)} = \sum_{k=1}^{l} \left[\Gamma_A\right]_{ik} \lambda_{kk} \left[\Gamma_B^{\dagger}\right]_{kj} \tag{B.3}$$

is the closest rank-l matrix to A, meaning that $A^{(l)}$ minimizes the *Frobenius norm*: the sum of the absolute squares of the element-wise difference between the rank-l approximation and the full matrix, $\sum_{ij} \left|A_{ij} - A_{ij}^{(l)}\right|^2$. Thus, the SVD gives us a means to find the best approximation to a matrix in a reduced-rank space.

Theorem: (Schmidt Decomposition) Let $|\psi\rangle$ be a pure state in the d_Ad_B dimensional Hilbert space $\mathcal{H}_A\otimes\mathcal{H}_B$. Then there exist vectors $\left\{|\phi^A_\alpha\rangle\right\}$ and $\left\{|\phi^B_\alpha\rangle\right\}$ and scalars χ_S and $\{\lambda_\alpha\}$ such that

$$|\psi\rangle = \sum_{\alpha=1}^{\chi_S}\lambda_\alpha|\phi^A_\alpha\rangle\otimes|\phi^B_\alpha\rangle\,, \tag{B.4}$$

$$1\le\chi_S\le\min\left(d_A,d_B\right)\,, \tag{B.5}$$

$$\lambda_1\ge\lambda_2\ge\cdots\ge\lambda_{\chi_S}\ge 0\,, \tag{B.6}$$

$$\sum_\alpha\lambda_\alpha^2=1\,. \tag{B.7}$$

This is referred to as the *Schmidt decomposition* of $|\psi\rangle$. χ_S is referred to as the *Schmidt rank*, and the $\{\lambda_\alpha\}$ are referred to as the *Schmidt coefficients*. We shall provide two proofs of this theorem. The first is a mathematical proof which sheds light on our numerical procedure, and the second is a physically motivated proof which elucidates the physical motivation for TEBD.

Proof 1: Let $\{|j_A\rangle\}$ and $\{|n_B\rangle\}$ be two orthonormal bases of $\mathcal{H}_A$ and $\mathcal{H}_B$ of dimension d_A and d_B, respectively. The most general decomposition of $|\psi\rangle$ in this basis is

$$|\psi\rangle = \sum_{j=1}^{d_A}\sum_{n=1}^{d_b}c_{jn}|j_A\rangle\otimes|n_B\rangle\,. \tag{B.8}$$

Applying the singular value decomposition to the coefficient matrix C, we have

$$|\psi\rangle = \sum_{j=1}^{d_A}\sum_{i=1}^{\min(d_A,d_B)}\sum_{n=1}^{d_B}\Gamma^A_{ji}\lambda_{ii}{\Gamma^B}^\dagger_{in}|j_A\rangle\otimes|n_B\rangle\,. \tag{B.9}$$

If we now define

$$|\phi^A_i\rangle\equiv\sum_{j=1}^{d_A}\Gamma^A_{ji}|j_A\rangle\,,\ \ |\phi^B_i\rangle\equiv\sum_{n=1}^{d_B}{\Gamma^B}^\dagger_{in}|n_B\rangle\,,\ \ \lambda_i\equiv\lambda_{ii}\,, \tag{B.10}$$

we have

$$|\psi\rangle = \sum_{i=1}^{\chi_S} \lambda_i |\phi_i^A\rangle \otimes |\phi_i^B\rangle \, , \tag{B.11}$$

as was to be proven. We can identify χ_S generally as the number of nonzero singular values of the decomposition matrix, with $1 \leq \chi_S \leq$min(d_A, d_B). We also note that the Schmidt decomposition may be easily performed numerically using the singular value decomposition of the coefficient matrix C.

Our second proof of the Schmidt decomposition begins as before, with

$$|\psi\rangle = \sum_{j=1}^{d_A} \sum_{n=1}^{d_b} c_{jn} |j_A\rangle \otimes |n_B\rangle \, . \tag{B.12}$$

We now define the (not necessarily orthonormal) basis

$$|\tilde{\mu}_B\rangle \equiv \sum_n c_{jn} |n_B\rangle \, , \tag{B.13}$$

yielding

$$|\psi\rangle = \sum_j |j_A\rangle \otimes |\tilde{\mu}_B\rangle \, . \tag{B.14}$$

We now choose as a basis for $\mathcal{H}_A$ a set $\{|m_A\rangle\}$ that diagonalizes the reduced density operator $\hat{\rho}_A$

$$\hat{\rho}_A = \mathrm{Tr}_B |\psi\rangle\langle\psi| = \sum_{m=1}^{\chi_S} p_m |m_A\rangle\langle m_A| \, , \tag{B.15}$$

where $\chi_S \leq d_A$ is the number of nonzero eigenvalues p_m of the reduced density matrix obtained by tracing out the degrees of freedom of subsystem B. If $\chi_S <$min(d_A, d_B), we complete the set with a set of $N_A - \chi_S$ vectors orthonormal to the $\{|m_A\rangle\}$. Explicitly taking the partial trace over the B degrees of freedom, we find

$$\hat{\rho}_A = \sum_{m\mu} \langle \tilde{\mu}_B | \tilde{m}_B \rangle | m_A \rangle \langle \mu_A | \,, \tag{B.16}$$

$$\Rightarrow \langle \tilde{\mu}_B | \tilde{m}_B \rangle = p_m \delta_{m\mu} \,, \tag{B.17}$$

and so the set $\{|\tilde{\mu}_B\rangle\}$ is, in fact, orthogonal with this choice for $\{|m_A\rangle\}$. Normalizing this set gives

$$|\mu_B\rangle = p_m^{-1/2} |\tilde{\mu}_B\rangle \,, \tag{B.18}$$

$$|\psi\rangle = \sum_{\mu} \sqrt{p_\mu} |\mu_A\rangle \otimes |\mu_B\rangle \,, \tag{B.19}$$

which is the Schmidt decomposition, as was to be shown. If we now trace over A, we find

$$\hat{\rho}_B = \sum_{\mu} p_\mu |\mu_B\rangle \langle \mu_B| \,, \tag{B.20}$$

and so we arrive at the conclusion that $\hat{\rho}_A$ and $\hat{\rho}_B$ have the same (nonzero) eigenvalues. The Schmidt rank χ_S is now identified with the number of nonzero eigenvalues of the reduced density matrices, and the Schmidt vectors are their eigenvectors. A very important result is that the Schmidt rank is invariant under a *local evolution* described by a unitary operator $U_A \otimes U_B$ (local in this case means our system does not interact with other systems, for example because all other systems are far away). We deduce from this that a product state cannot be transformed into an entangled state through a local evolution in which the systems $\mathcal{H}_A$ and $\mathcal{H}_B$ evolve independently (also, we can't disentangle a state by the same means).

A two-body state is a tensor product (i.e. not entangled) iff its Schmidt rank is one. Similarly, a two-body state is maximally entangled if the Schmidt rank is $\min(d_A, d_B)$. This motivates the following entanglement measure associated with the Schmidt rank. Consider an n-body system of qu-*dits* (d-level quantum systems) described by a pure state $|\psi\rangle$ which can be expressed in terms of the local bases

$\{|i_1\rangle, |i_2\rangle, ..., |i_L\rangle\}$ as

$$|\Psi\rangle = \sum_{i_1,i_2,\ldots,i_L=1}^{d} c_{i_1,i_2,\ldots,i_L} |i_1\rangle|i_2\rangle \cdots |i_n\rangle \, , \tag{B.21}$$

and consider all *bipartite splittings* of this system. That is, consider all decompositions of the form

$$|\Psi\rangle = \sum_{i_l=1}^{\chi_{Sl}} \lambda_{i_l}^{[l]} |\phi_{i_l}^{[l]}\rangle |\phi_{i_l}^{[n-l]}\rangle \, , \tag{B.22}$$

where the Schmidt vectors $|\phi^{[l]}\rangle$ span the Hilbert space also spanned by $|i_1\rangle|i_2\rangle \cdots |i_l\rangle$, the Schmidt vectors $|\phi^{[n-l]}\rangle$ span the Hilbert space also spanned by $|i_{l+1}\rangle|i_{l+2}\rangle \cdots |i_L\rangle$, and $1 \leq \chi_{Sl} \leq \min\left(d^l, d^{L-l}\right)$ is the Schmidt rank associated with bipartite splitting l. Note that these decompositions are guaranteed to exist by the Schmidt decomposition. We define $E_{\chi_S} \equiv \log_d(\chi_S)$ where $\chi_S \equiv \max_l(\chi_{Sl})$ as the *Schmidt measure*, a measure of entanglement central to TEBD. Some properties of the Schmidt measure are explored in the exercises.

Exercise: Prove that a local unitary operation $U_A \otimes U_B$ leaves the Schmidt rank unchanged. Hint: Act on Eq. (B.19) with the given operator.

Exercise: You proved in the last exercise that the Schmidt measure is *entanglement monotone*, i.e. that it is unchanged by local unitary operations. Prove that it is bounded below by zero (when does it reach this bound?) and also that it bounds the *entropy of entanglement*

$$S = -\sum_{\alpha} \lambda_{\alpha}^2 \log_d \lambda_{\alpha}^2. \tag{B.23}$$

Finally, prove that it is additive under tensor products.

$$E_{\chi_S}\left(|\psi\rangle \otimes |\psi'\rangle\right) = E_{\chi_S}\left(|\psi\rangle\right) + E_{\chi_S}\left(|\psi'\rangle\right) \tag{B.24}$$

These conditions show more rigorously that E_{χ_S} is a faithful measure of entanglement for pure states.

B.3.2 Conceptual Basis of TEBD

The essential fact that allows for the efficiency and success of TEBD is that the singular values of reduced density matrices, when arranged in non-increasing fashion, have an approximately exponential decay. This means that a rank-χ approximation to the reduced density matrix, $\hat{\rho}^{(\chi)}$, formed from the singular value decomposition as in Eq. (B.3) with $\chi \ll \chi_S$ will provide an excellent approximation to the true reduced density matrix. We can understand this fact using an analogy to image compression.

Consider the JPEG image shown in Figure B.1(a). We can represent this picture as a 300×416 array of pixels, requiring 124800 words of storage. If we look at the singular values of this pixel array, we find the distribution shown in Figure B.1(b), namely that the singular values decay roughly exponentially. This implies that the best approximation to the image given by the SVD will be excellent even if we keep only a fraction of the total singular values. To see this in action, examine Figures Figure B.1(c)–Figure B.1(e), which show the best approximation to the image for various numbers of singular values. Keeping 100 singular values gives an excellent approximation to the original image with only a quarter of the data storage. If we consider that the original image was a JPEG which itself was compressed down from a ∼3000×3000 pixel (10 Megapixel) raw array, we have benefitted enormously by carefully sampling the parts of "image space" that are the most important via the SVD.

Why were we able to represent the image with so little data? The reason is that a physical images are a very special subset of all 2D pixel arrays; they have a great deal of structure and regularity. If we consider instead the most probable image of the same size as our original image–one consisting of random pixel values–we get the singular value scaling shown in Figure B.1(f). In this case the singular values do scale exponentially with the data size, and our SVD compression would yield a miserable approximation to the original image if we were to use only a quarter of the singular values. In the same manner typical physical states in Hilbert space, including the ground state, have limited entanglement as quantified by *some* entanglement measure. This limited entanglement means the singular value spectra of physical states decay exponentially more quickly than a general state, enabling them to be "entanglement compressed" by algorithms such as TEBD. There is no general proof of this fact just as there is no guarantee that an image will come out perfectly crisp after JPEG compression, it is simply a trend observed in many-body quantum systems.

B.3.3 Vidal's State Decomposition

The starting point for TEBD is to express the L-body coefficient tensor $c_{i_1 \dots i_L}$ defined as

$$|\Psi\rangle = \sum_{i_1 \dots i_L = 1}^{d} c_{i_1 \dots i_L} |i_1\rangle \dots |i_L\rangle \tag{B.25}$$

as a sum over products of local tensors $\Gamma^{[l]}$ and local vectors $\lambda^{[l]}$ as

$$c_{i_1 i_2 \dots i_L} = \sum_{\alpha_0 \dots \alpha_L}^{\chi_S} \lambda^{[1]}_{\alpha_0} \Gamma^{[1] i_1}_{\alpha_0 \alpha_1} \lambda^{[2]}_{\alpha_1} \Gamma^{[2] i_2}_{\alpha_1 \alpha_2} \lambda^{[3]}_{\alpha_2} \Gamma^{[3] i_3}_{\alpha_2 \alpha_3} \dots \Gamma^{[L] i_L}_{\alpha_{L-1} \alpha_L} \lambda^{[L+1]}_{\alpha_L} \tag{B.26}$$

or, for open boundary conditions, as

$$c_{i_1 i_2 \dots i_L} = \sum_{\alpha_1 \dots \alpha_{L-1}}^{\chi_S} \Gamma^{[1] i_1}_{\alpha_1} \lambda^{[2]}_{\alpha_1} \Gamma^{[2] i_2}_{\alpha_1 \alpha_2} \lambda^{[3]}_{\alpha_2} \Gamma^{[3] i_3}_{\alpha_2 \alpha_3} \dots \Gamma^{[L] i_L}_{\alpha_{L-1}} , \tag{B.27}$$

where we have defined

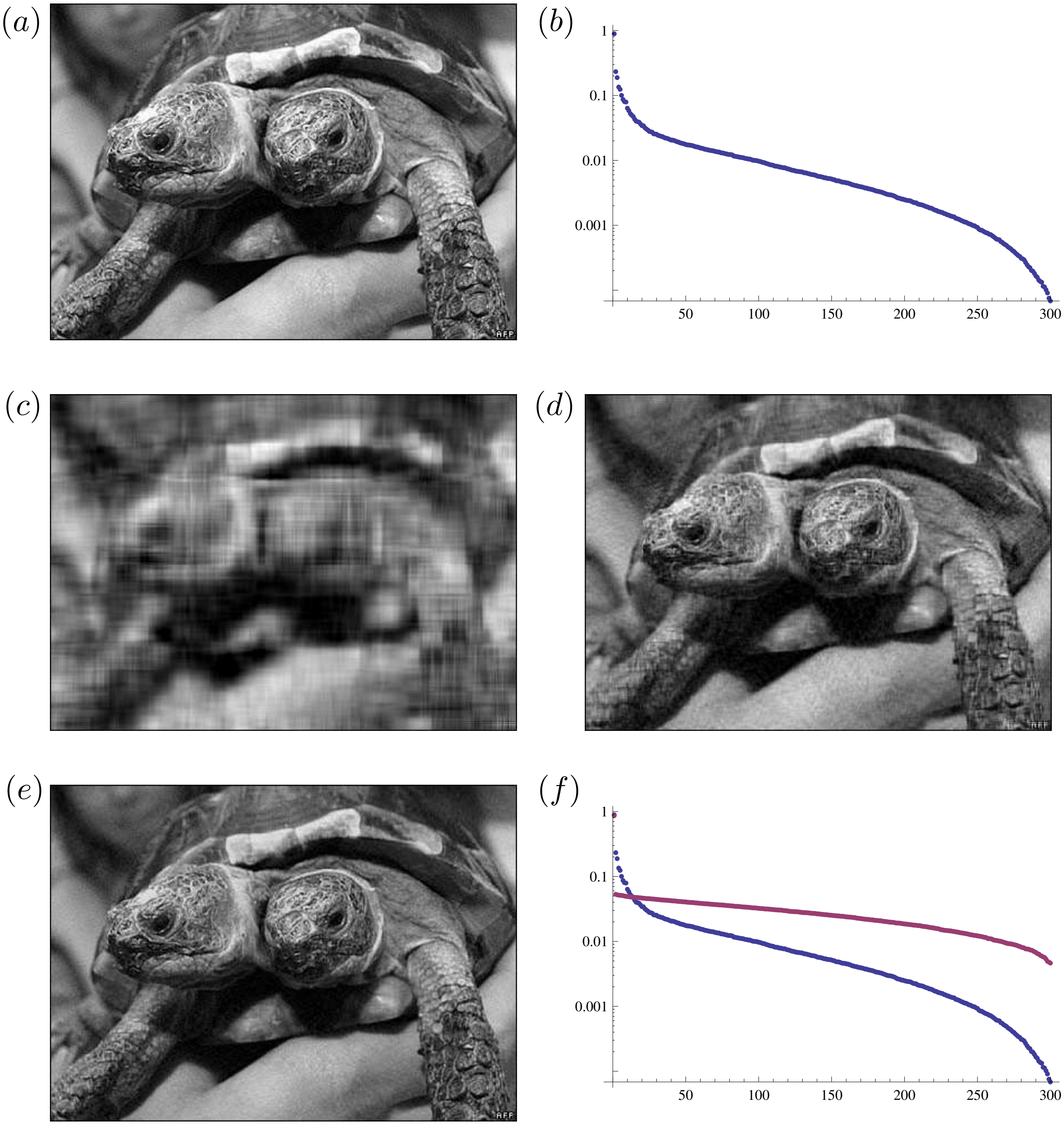


Figure B.1: SVD representation of an image. (a) Original Image. (b) Log plot of singular values. (c) 10 Singular values. (d) 50 Singular values. (e) 100 Singular values. (f) Log plot of singular values: General pixel array (pink), Physical pixel array (blue).

$$\Gamma^{[1]i_1}_{\alpha_1} \equiv \sum_{\alpha_0} \lambda^{[1]}_{\alpha_0} \Gamma^{[1]i_1}_{\alpha_0\alpha_1}, \quad \Gamma^{[L]i_L}_{\alpha_{L-1}} \equiv \sum_{\alpha_L} \Gamma^{[L]i_L}_{\alpha_{L-1}\alpha_L} \lambda^{[L+1]}_{\alpha_L} . \tag{B.28}$$

For open boundary conditions, $\lambda^{[1]}_{\alpha_0} = \delta_{\alpha_0,1}$ and $\lambda^{[L]}_{\alpha_L} = \delta_{\alpha_L,1}$, and so the two decompositions are identical. We will use the latter expression, Eq. (B.27), most often in theoretical considerations, only using Eq. (B.26) when periodic boundary conditions require us to.

In Eq. (B.25), d is the Hilbert space of site l, which may or may not be a physical site, we simply mean it refers to the local Hilbert space spanned by $|i_l\rangle$. This implies that the Hilbert space of the full system is d^L, meaning we need d^L coefficients, the $\{c_{i_1,i_2,\ldots,i_L}\}$, to specify our state $|\Psi\rangle$. In the second expression, we have replaced the d^L coefficients with $d\chi^2 L + \chi\,(L+1)$ coefficients, the Γs and λs.

We refer to the Γs and λs as being local because each one is assigned either to a site (Γs) or a *link* (also called a *bond* in the condensed matter literature) between sites (λs). The schematic replacement is shown in Figure B.2. The top row of the figure shows the system before the decomposition as being a collection of sites (blue ovals) linked together via the tensor product. After the decomposition, each site is replaced with a four-index tensor $\Gamma^{[l]i_l}_{\alpha_{l-1},\alpha_l}$. The superscript index in brackets l denotes that the Γ is associated with the l^{th} site, the superscript index without brackets i_l denotes that the system is in the local state $|i_l\rangle$, and the subscripts α_{l-1} and α_l are the *Schmidt indices* which relay information about the remainder of the system with this particular on-site configuration. Similarly, we replace each tensor product with a “link,” and upon each link is associated with a $\lambda^{[l]}_{\alpha_l}$. The superscript index in brackets l denotes the link index, with link l being between sites $l-1$ and l, and the subscript index is the Schmidt index which gives the relative importance of the state represented by the α_l^{th} Schmidt index of the corresponding Γ.

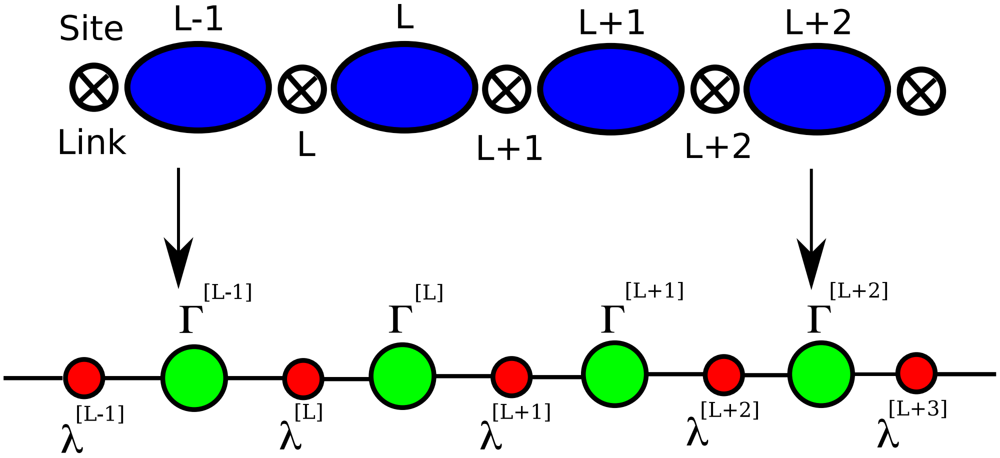


Figure B.2: Schematic of Vidal Decomposition. In the lower representation a line connecting two objects is an implied summation over a shared index. The lines shown are the contractions over the Schmidt indices as in Eq. (B.26).

We have written the Vidal Decomposition, Eq. (B.26), in a way that reflects the indexing used in the actual code. Other works [7, 8] write the decomposition for open boundary conditions as

$$c_{i_1 i_2 \dots i_L} = \sum_{\alpha_1 \dots \alpha_{L-1}=1}^{\chi_S} \Gamma^{[1]i_1}_{\alpha_1} \lambda^{[1]}_{\alpha_1} \Gamma^{[2]i_2}_{\alpha_1 \alpha_2} \lambda^{[2]}_{\alpha_2} \Gamma^{[3]i_3}_{\alpha_2 \alpha_3} \dots \lambda^{[L-1]}_{\alpha_{L-1}} \Gamma^{[L]i_L}_{\alpha_{L-1}}, \tag{B.29}$$

which becomes ours upon the substitution

$$\text{this work's } \lambda^{[l]}_{\alpha_l} = \lambda^{[l-1]}_{\alpha_l} \text{ other work.} \tag{B.30}$$

The reason that we have reindexed the λs is purely pragmatic: Fortran naturally indexes from 1 and not from 0. This definition allows for direct comparison of the manual and the code. We hope that this is not the source of any confusion.

The Γs and λs are chosen such that a splitting of our system between sites l and $l+1$ into two subsystems (a *bipartite splitting*) is exactly the Schmidt decomposition

$$|\psi\rangle = \sum_{\alpha_l=1}^{\chi_S} \lambda^{[l+1]}_{\alpha_l} |\phi^{[1\dots l]}_{\alpha_l}\rangle |\phi^{[l+1\dots n]}_{\alpha_l}\rangle \tag{B.31}$$

meaning that the Schmidt vectors

$$|\phi^{[1\ldots l]}_{\alpha_l}\rangle = \sum_{\alpha_0,\ldots,\alpha_{l-1}}^{\chi_S} \lambda^{[1]}_{\alpha_0}\Gamma^{[1]i_1}_{\alpha_0\alpha_1}\lambda^{[2]}_{\alpha_1}\Gamma^{[2]i_2}_{\alpha_1\alpha_2}\lambda^{[3]}_{\alpha_2}\Gamma^{[3]i_3}_{\alpha_2\alpha_3}\ldots\Gamma^{[l]i_l}_{\alpha_{l-1}\alpha_l}|i_1\rangle\ldots|i_l\rangle \tag{B.32}$$

and

$$|\phi^{[l+1\ldots L]}_{\alpha_l}\rangle = \sum_{\alpha_{l+1},\ldots,\alpha_L}^{\chi_S} \Gamma^{[l+1]i_{l+1}}_{\alpha_l\alpha_{l+1}}\lambda^{[l+1]}_{\alpha_{l+2}}\Gamma^{[l+2]i_{l+2}}_{\alpha_{l+1}\alpha_{l+2}}\ldots\Gamma^{[L]i_L}_{\alpha_{L-1}\alpha_L}\lambda^{[L]}_{\alpha_L}|i_{l+1}\rangle\ldots|i_L\rangle \tag{B.33}$$

span the reduced density matrices

$$\rho^{(1\ldots l)} = \mathrm{Tr}_{(l+1)\ldots L}|\psi\rangle\langle\psi| \tag{B.34}$$

and

$$\rho^{(l+1\ldots L)} = \mathrm{Tr}_{1\ldots l}|\psi\rangle\langle\psi| \tag{B.35}$$

respectively, and the $\left(\lambda^{[l+1]}_{\alpha_l}\right)^2$ are their eigenvalues.

With the Γs and λs chosen in this manner, we have incurred no error as long as we select χ_S to be the Schmidt rank $\min\left[d^l, d^{L-l}\right]$. The approximation comes when we select out the χ largest eigenvalues $\lambda^{[l+1]}_{\alpha_l}$, rejecting the other $\chi_S - \chi$. Since we have replaced d^L parameters with $d\chi^2 L + \chi\left(L+1\right)$ parameters, for this to be of significant value χ_S must scale polynomially in L, which is the same essential idea as the singular value spectrum decaying exponentially discussed in Sec. B.3.3. In light of the discussion of Sec. B.3.1, we note that the precise way in which TEBD best approximates the ground state is by forming the best rank-χ approximation to the coefficient tensor of each bipartite splitting.

To give some physical motivation for the polynomial scaling of χ with the system size, consider the following argument due to Zwolak[9]. For a 1D critical system with finite range interactions the entropy of entanglement (von Neumann entropy) of a contiguous block of length ξ scales as

$$S \sim \frac{c}{3} \log \xi \,, \tag{B.36}$$

where c is the central charge of the underlying conformal field theory[10]. If we consider the block to be half of a system with open boundary conditions of size L, the entropy scales as

$$S \sim \frac{c}{6} \log L \,; \tag{B.37}$$

it is half of what may have been expected from above because the block has two boundaries. This growth is much slower than the maximum growth possible, which would be linear in L. The maximum entropy that a truncated Vidal decomposition can hold is $\log \chi$ (as proved in the exercises). For the truncation error to remain fixed as we increase the system size, we should thus scale the entanglement cutoff roughly as

$$\chi \sim L^{c/6} \,, \tag{B.38}$$

that is, polynomially in the system size. The polynomial scaling can be made much more precise[11], but the important idea is that, even for *critical* 1D systems, the Vidal decomposition *can* efficiently represent the state.

B.3.3.1 Construction of the Local Tensors using the Singular Value Decomposition

The decomposition, Eq. (B.26), cannot be used directly to construct an initial state because we do not know the full coefficient tensor in most cases. Nevertheless, it is instructive to show how to construct the local tensors from the full coefficient tensor in simple cases. If we consider a bipartite splitting of our full wavefunction into two systems spanned by $|i_A\rangle$ and $|j_B\rangle$, then we have

$$|\Psi\rangle = \sum_{i,j} c_{ij} |i_A j_B\rangle \,. \tag{B.39}$$

Via the singular value decomposition, we find the coefficient matrix to be

$$c_{ij} = \sum_k \Gamma^A_{ik} \lambda_{kk} \Gamma^B_{kj} \,, \tag{B.40}$$

$$\Rightarrow |\Psi\rangle = \sum_k \lambda_{kk} |\phi^{[A]}_k\rangle |\phi^{[B]}_k\rangle \,, \tag{B.41}$$

$$|\phi^{[A]}_k\rangle \equiv \sum_i \Gamma^A_{i,k} |i_A\rangle \,, \qquad |\phi^{[B]}_k\rangle \equiv \sum_j \Gamma^B_{j,k} |j_B\rangle. \tag{B.42}$$

We now show how to use the SVD to perform Vidal's decomposition. For the sake of clarity, we will consider only a four site system, as the generalization to more sites is straightforward. The Vidal Decomposition of the 4-party coefficient tensor is

$$c_{i_1 i_2 i_3 i_4} = \sum_{\alpha_1,\alpha_2,\alpha_3=1}^{\chi} \Gamma^{[1]i_1}_{\alpha_1} \lambda^{[2]}_{\alpha_1} \Gamma^{[2]i_2}_{\alpha_1\alpha_2} \lambda^{[3]}_{\alpha_2} \Gamma^{[3]i_3}_{\alpha_2\alpha_3} \lambda^{[4]}_{\alpha_3} \Gamma^{[4]i_4}_{\alpha_3} \,. \tag{B.43}$$

Performing a Schmidt decomposition at the first link yields

$$|\Psi\rangle = \sum_{\alpha_1=1}^{\chi} \lambda^{[2]}_{\alpha_1} |\phi^{[1]}_{\alpha_1}\rangle |\phi^{[234]}_{\alpha_1}\rangle = \sum_{\alpha_1;i_1,i_2,i_3,i_4} \lambda^{[2]}_{\alpha_1} \Gamma^{[1]i_1}_{\alpha_1} |i_1\rangle \otimes \left(\Gamma^{[234]i_2 i_3 i_4}_{\alpha_1} |i_2 i_3 i_4\rangle\right) \,. \tag{B.44}$$

We obtain the numbers $\lambda^{[2]}_{\alpha_1}$, $\Gamma^{[1]i_1}_{\alpha_1}$, and $\Gamma^{[234]i_2 i_3 i_4}_{\alpha_1}$ from performing an SVD at this splitting as outlined above. Comparing with the full decomposition, Eq. (B.43), we have the relation

$$\Gamma^{[234]i_2 i_3 i_4}_{\alpha_1} = \sum_{\alpha_2 \alpha_3} \Gamma^{[2]i_2}_{\alpha_1\alpha_2} \lambda^{[3]}_{\alpha_2} \Gamma^{[3]i_3}_{\alpha_2\alpha_3} \lambda^{[4]}_{\alpha_3} \Gamma^{[4]i_4}_{\alpha_3} \tag{B.45}$$

which does not yet allow us to solve for all of the local tensors uniquely. Performing a second bipartite splitting, now at the second link, yields

$$|\Psi\rangle = \sum_{\alpha_2=1}^{\chi} \lambda^{[3]}_{\alpha_2} |\phi^{[12]}_{\alpha_2}|\phi^{[34]}_{\alpha_2}\rangle = \sum_{\alpha_2;i_1i_2i_3i_4} \lambda^{[3]}_{\alpha_2} \Gamma^{[12]i_1i_2}_{\alpha_2} |i_1i_2\rangle \otimes \left(\Gamma^{[34]i_3i_4}_{\alpha_2}|i_3i_4\rangle\right) \, . \tag{B.46}$$

Comparing again with the full decomposition, we find

$$\Gamma^{[34]i_3i_4}_{\alpha_2} = \sum_{\alpha_3} \Gamma^{[3]i_3}_{\alpha_2\alpha_3} \lambda^{[4]}_{\alpha_3} \Gamma^{[4]i_4}_{\alpha_3} \, , \tag{B.47}$$

$$\Rightarrow \Gamma^{[12]i_1i_2}_{\alpha_2} = \sum_{\alpha_1} \Gamma^{[1]i_1}_{\alpha_1} \lambda^{[2]}_{\alpha_1} \Gamma^{[2]i_2}_{\alpha_1\alpha_2} \, . \tag{B.48}$$

We now know $\lambda^{[3]}_{\alpha_2}$, $\Gamma^{[12]i_1i_2}_{\alpha_2}$, and $\Gamma^{[34]i_3i_4}_{\alpha_2}$ in addition to the coefficients from the first splitting. With this information we can solve for $\Gamma^{[2]i_2}_{\alpha_1\alpha_2}$ on the right hand side of Eq. (B.48). Performing one last bipartite splitting at the third link provides us with the equation

$$\Gamma^{[34]i_3i_4}_{\alpha_3} = \sum_{\alpha_3} \Gamma^{[3]i_3}_{\alpha_2\alpha_3} \lambda^{[3]}_{\alpha_3} \Gamma^{[4]i_4}_{\alpha_3} \tag{B.49}$$

along with $\lambda^{[4]}_{\alpha_3}$, $\Gamma^{[123]i_1i_2i_3}_{\alpha_3}$, and $\Gamma^{[4]i_4}_{\alpha_3}$. With all of this we can solve for $\Gamma^{[4]i_4}_{\alpha_3}$, uniquely specifying all of the local tensors.

For a few particular states we can compute the local tensors intuitively "by inspection." One such state is the L-party vacuum state $|\Psi\rangle = |0\rangle^{\otimes L}$. Recall that $\chi = 1$ for a product state such as this and so all bipartite splittings will involve only one term. Performing a splitting at the first link yields

$$|\Psi\rangle = \sum_{\alpha_1} \lambda^{[2]}_{\alpha_1} |\phi^{[1]}_{\alpha_1}\rangle |\phi^{[2\ldots L]}_{\alpha_1}\rangle = |0\rangle|0\rangle^{\otimes L-1} \tag{B.50}$$

from which we can read off $\lambda^{[1]}_{\alpha_1} = \delta_{\alpha_1,1}$. Inspecting the first Schmidt vector,

$$|\phi^{[1]}_{\alpha_1}\rangle = \sum_{i_1} \Gamma^{[1]i_1}_{\alpha_1} |i_1\rangle \, , \tag{B.51}$$

we are also able to read off $\Gamma^{[1]i_1}_{\alpha_1} = \delta_{\alpha_1,1}\delta_{i_1,0}$. If we now expand the second Schmidt vector in a local basis for the second site we find

$$|\phi^{[2\ldots L]}_{\alpha_1}\rangle = \sum_{i_2} |i_2\rangle|\tau^{[3\ldots L]}_{\alpha_1 i_2}\rangle. \tag{B.52}$$

This allows us to easily identify

$$|\tau^{[3\ldots L]}_{\alpha_1 i_2}\rangle = \delta_{\alpha_1,1}\delta_{i_2,0}|0\rangle^{\otimes L-2} \tag{B.53}$$

and, with the decomposition definition

$$|\tau^{[3\ldots M]}_{\alpha_1 i_2}\rangle = \sum_{\alpha_2} \Gamma^{[2]i_2}_{\alpha_1\alpha_2}\lambda^{[3]}_{\alpha_2}|\phi^{[3\ldots L]}_{\alpha_2}\rangle \tag{B.54}$$

$$|\phi^{[3\ldots L]}_{\alpha_2}\rangle = |0\rangle^{\otimes L-2}\,, \tag{B.55}$$

we find $\Gamma^{[2]i_2}_{\alpha_1\alpha_2} = \delta_{\alpha_1,1}\delta_{\alpha_2,1}\delta_{i_2,0}$. Noting the general trend, we can write down $\Gamma^{[l]i_l}_{\alpha_{l-1}\alpha_l} = \delta_{\alpha_{l-1},1}\delta_{\alpha_l,1}\delta_{i_2,0}$ and $\lambda^{[l+1]}_{\alpha_l} = \delta_{\alpha_l,1}$. This generalizes immediately to the most general product state

$$|\Psi\rangle = \bigotimes_{k=1}^{L} |\psi_k\rangle \text{ where } |\psi_k\rangle = \sum_{i_k=1}^{d} c^{(k)}_{i_k}|i_k\rangle. \tag{B.56}$$

As this is a product state, all of the $\lambda^{[l+1]}_{\alpha_l} = \delta_{\alpha_l,1}$ as above, but $\Gamma^{[l]i_l}_{\alpha_{l-1}\alpha_l} = c^{(l)}_{i_l}\delta_{\alpha_{l-1},1}\delta_{\alpha_l,1}$.

Exercise: The product state examples above were particularly simple because they were not entangled. Let us now consider the Vidal decomposition of a state of maximal entanglement (by some entanglement measures), the L-party GHZ state

$$|\Psi\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle^{\otimes L} + |1\rangle^{\otimes L}\right)\,. \tag{B.57}$$

Noting that a bipartite splitting at an arbitrary link l will involve only $\chi = 2$ terms

$$|\Psi\rangle = \frac{1}{\sqrt{2}} \left(|0\rangle^{\otimes l}|0\rangle^{\otimes L-l} + |1\rangle^{\otimes l}|1\rangle^{\otimes L-l}\right) \, , \tag{B.58}$$

solve for the $\lambda^{[l]}_{\alpha_l}$ and $\Gamma^{[l]i_l}_{\alpha_{l-1}\alpha_l}$.

B.3.3.2 Initial State Selection

The methods developed in the last section give us a means to initialize the local tensors of our system. The most straightforward means of initializing the system is to generate a product state where the coefficient of state i on site l is the (i,l) entry of a specified matrix C. In this case we have

$$\Gamma^{[l]i}_{\alpha_{l-1}\alpha_l} = C_{il}\delta_{\alpha_{l-1},1}\delta_{\alpha_l,1} \, , \tag{B.59}$$

$$\lambda^{[l+1]}_{\alpha_l} = \delta_{\alpha_l,1} \, . \tag{B.60}$$

This operation is performed in the procedure `ProductStateMPD` discussed in Sec. B.9.2.1. If the initial state is a product of number eigenstates, the lists of number conserving vectors `LabelLeft` and `LabelRight` (Sec. B.3.4.3) can be constructed using the routine `ProductStateLabels` discussed in Sec. B.9.2.2.

Some states, namely the generalized GHZ state

$$|\mathrm{GHZ}\rangle = \frac{1}{\sqrt{d}} \left[|00\dots0\rangle + |11\dots1\rangle + \dots + |(d-1)(d-1)\dots(d-1)\rangle\right] \tag{B.61}$$

and the W state

$$|\mathrm{W}\rangle = \frac{1}{\sqrt{L}} \sum_{k=1}^{L} |0\rangle^{\otimes L-k}|1\rangle|0\rangle^{\otimes k-1} \tag{B.62}$$

have analytically known Vidal representations, and so can be used as initial states. This is done using the procedure `SpecialState` discussed in Sec. B.9.8.38. Also included in `SpecialState` is a routine to generate a hard-core boson cluster state[12] by the following steps:

1. Begin in the state $|00\dots0\rangle$ which corresponds to a tensor product of spin up qubits.

2. Act on each site with $\exp\left[-\frac{\pi}{4}\left(\hat{a}_i-\hat{a}_i^{\dagger}\right)\right]$ corresponding to rotation of each qubit by $\pi/4$ with respect to the y-axis.

3. Evolve under the Hamiltonian $\hat{H}=\sum_i\left(1-2\hat{a}_i^{\dagger}\hat{a}_i\right)\left(1-2\hat{a}_{i+1}^{\dagger}\hat{a}_{i+1}\right)$, corresponding to the Ising Hamiltonian for the qubits, for t=$\pi/4$.

4. Act on each site with $\exp\left[-i\frac{\pi}{4}\left(1-2\hat{a}_i^{\dagger}\hat{a}_i\right)\right]$ corresponding to rotation of each qubit by $\pi/4$ with respect to the z-axis.

Often one wishes to begin time evolution in the ground state of a particular Hamiltonian, but the ground state is unknown. By a method known as *imaginary time propagation* we let $\tau\equiv it$ replace time in the governing Schrödinger equation and then evolve the system. With this replacement the Schrödinger equation takes the form of a diffusion equation with the highest energy eigenmodes decaying the most quickly. Thus, after some long "time" τ, if the overlap of the initial state used in imaginary time propagation with the actual ground state of the Hamiltonian is nonzero, we will recover the ground state. For finite systems with a gap $\Delta>0$, we can make this more precise as

$$|\langle\psi_\tau|\psi_g\rangle|>1-\mathcal{O}\left(\frac{e^{-2\Delta\tau}}{\delta^2}\right)\,,\quad \delta\equiv|\langle\psi_0|\psi_g\rangle|\,, \tag{B.63}$$

where $|\psi_0\rangle$ is the initial state used in imaginary time propagation, $|\psi_\tau\rangle$ is the state at imaginary time τ, and $|\psi_g\rangle$ is the true ground state [13]. Because this evolution is not unitary, we must also renormalize and re-orthogonalize the state after each time step, see Sec. B.3.4.4.

An appropriate state for imaginary time propagation, if we are not concerned with conservation of any quantum numbers, is the product state that contains all possible states in the same amount: $c_{i_1,i_2,\ldots,i_L} = 1/\sqrt{d^L}\;\forall\; i_i, i_2, \ldots, i_L$. The Vidal decomposition for this state is

$$\Gamma^{[l]i}_{\alpha_{l-1}\alpha_l} = \frac{1}{\sqrt{d}}\delta_{\alpha_{l-1},1}\delta_{\alpha_l,1}\,, \tag{B.64}$$

$$\lambda^{[l+1]}_{\alpha_l} = \delta_{\alpha_l,1}\,. \tag{B.65}$$

This operation is performed in the procedure `AllStates` discussed in Sec. B.9.2.3.

The state generated by `AllStates` is not an eigenstate of any operator except in highly atypical cases. Thus, this state is not acceptable for symmetry conserving implementations (Sec. B.3.4.3) which require the Γs to be expressed in the eigenbasis of the symmetry generator. In the case of number conservation, the routine `InitialSetNC` will generate an initial state with total number `totNum` such that all Γs are expressed in a basis of number eigenstates(Sec. B.8.1.31). The routine was also designed so that the given state is the closest approximation to a gaussian centered at the center of the system, referred to in the code as a "wedding cake" structure. We outline the method here.

The wedding cake structure has two possible tiers, differing in on-site number by one. The number per site in the upper tier region is denoted "center" and the number per site in the lower tier region is center-1. The region to the left of the upper tier is "hole" sites long, the upper tier region is "tops" sites long, and the region to the right of the upper tier is the remainder up to L. hole, tops, and center are defined as

$$\mathrm{tops} \equiv \mathrm{MOD}\,(N-1, L) + 1\,, \tag{B.66}$$

$$\mathrm{center} \equiv \lfloor \frac{N}{L} - 10^{-8} \rfloor + 1\,, \tag{B.67}$$

$$\mathrm{hole} \equiv \begin{cases} \begin{array}{c} \text{if MOD}(L\text{-MOD}(N,L),2)=0 \\ \text{or MOD}(N,L)=0 \end{array} & \begin{cases} 0 \text{ if MOD}(N,L)=0 \\ (L-\mathrm{MOD}(N,L))/2 \text{ otherwise} \end{cases} \\ (L-\mathrm{MOD}(N,L)+1)/2 \text{ otherwise} & \end{cases}, \tag{B.68}$$

where N is the total number of particles in the system and $\lfloor x \rfloor$ denotes the greatest integer less than or equal to x. The procedure first generates the vectors `LabelLeft` and `LabelRight` which hold the cumulative number to the left and right of a particular link, respectively. The procedure then constructs the local tensors according to

$$\Gamma^{[l]i}_{\alpha_{l-1}\alpha_l} = \frac{1}{\sqrt{d_{n_l}}}\delta_{\alpha_{l-1},1}\delta_{\alpha_l,1}\delta_{n_i,n_l}\,, \tag{B.69}$$

$$\lambda^{[l+1]}_{\alpha_l} = \delta_{\alpha_l,1}\,. \tag{B.70}$$

where n_i is the number in the on-site state $|i_l\rangle$, n_l is the number on site l by the construction above, and d_{n_l} is the number of on-site states with $n_i = n_l$. In the absence of internal degrees of freedom $d_{nl} = 1$, but with internal degrees of freedom we must account for all states with a given number. Some examples make this clearer. Consider Table B.1, which shows hole, tops, center, and the onsite configuration (with the number of dots within a site representing the number of particles in that particular site) for 4 sites and various numbers.

Further implementation details are discussed under the `InitialSetNC` procedure listing in Sec. B.9.2.6.

B.3.4 Local Operations in the Vidal Representation

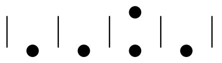
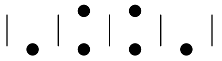
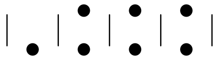
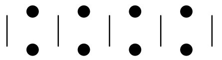
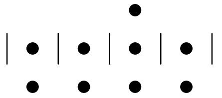
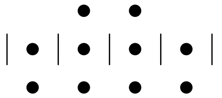
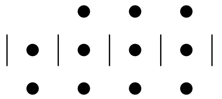
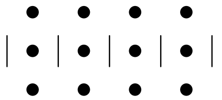
B.3.4.1 One Site Operations

Consider acting on a state with a unitary operator $\hat{U}$ that acts only on a single site:

$$\hat{U} = \sum_{i_l,i_l'} U_{i_l i_l'}|i_l\rangle\langle i_l'|. \tag{B.71}$$

In the Schmidt decomposition for a bipartite splitting at link l, this operator will not modify the left Schmidt vector, and so all $\Gamma^{[i]}$ and $\lambda^{[i+1]}$ with $i \le l-1$ remain unchanged. In the Schmidt decomposition for a bipartite splitting at link $l+1$, this operator will not modify the right Schmidt vector, and so all $\Gamma^{[i]}$ and $\lambda^{[i]}$ with $i \ge l+1$

Table B.1: Result of the number conserving initial state routine for four sites.

Number of particles	hole	tops	center	On-site configuration
1	2	1	1	| | | • | |
2	1	2	1	| | • | • | |
3	1	3	1	| | • | • | • |
4	0	4	1	| • | • | • | • |
5	2	1	2	
6	1	2	2	
7	1	3	2	
8	0	4	2	
9	2	1	3	
10	1	2	3	
11	1	3	3	
12	0	4	3	

remain unchanged. Thus, the only affected local tensor is $\Gamma^{[l]}$, which transforms as

$$\tilde{\Gamma}^{[l]i_l}_{\alpha_{l-1}\alpha_l} = \sum_{i'_l} U_{i_l i'_l} \Gamma^{[l]i'_l}_{\alpha_{l-1}\alpha_l} \, . \tag{B.72}$$

Such a transformation was to be expected, as $\Gamma^{[l]}$ is the only local tensor that carries any information about the site l. One-site operations are performed numerically with the procedure `OneSiteOp` discussed in Sec. B.9.8.3.

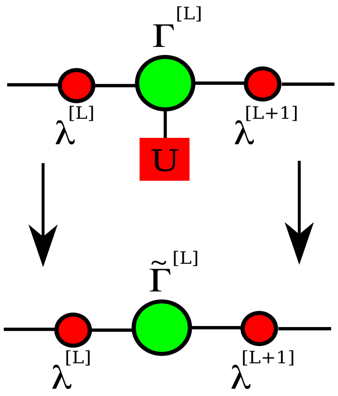


Figure B.3: Schematic of a one site operation.

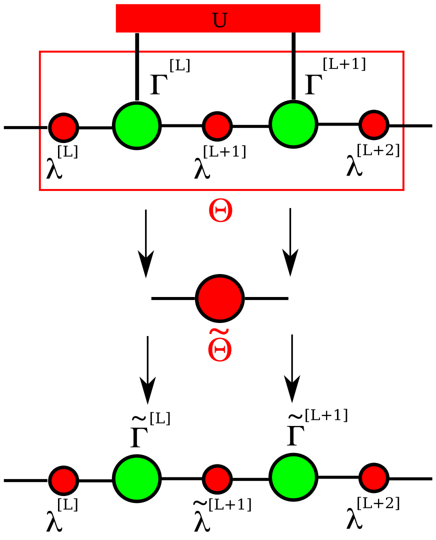


Figure B.4: Schematic of a two site operation.

B.3.4.2 Two Site Operations

Let us now consider acting on a state with a unitary operator $\hat{V}$ that acts on two sites:

$$\hat{V} = \sum_{i_l, i_{l+1}; i'_l i'_{l+1}} V^{i_l i_{l+1}}_{i'_l i'_{l+1}} |i_l i_{l+1}\rangle\langle i'_l i'_{l+1}|. \tag{B.73}$$

If we write the initial state as a bipartite splitting between sites l and $l+1$

$$|\Psi\rangle = \sum_{\alpha_{l-1}\alpha_l\alpha_{l+1}; i_l, i_{l+1}} \lambda^{[l]}_{\alpha_{l-1}} \Gamma^{[l]i_l}_{\alpha_{l-1}\alpha_l} \lambda^{[l+1]}_{\alpha_l} \Gamma^{[l+1]i_{l+1}}_{\alpha_l \alpha_{l+1}} \lambda^{[l+2]}_{\alpha_{l+1}} |\phi^{[1\ldots l-1]}_{\alpha_{l-1}}\rangle \otimes |i_l i_{l+1}\rangle \otimes |\phi^{[l+2\ldots M]}_{\alpha_{l+1}}\rangle \tag{B.74}$$

then we can define the object Θ as (see Figure B.4)

$$|\Psi\rangle = \sum_{\alpha_{l-1}\alpha_{l+1}; i_l, i_{l+1}} \Theta^{i_l i_{l+1}}_{\alpha_{l-1}\alpha_{l+1}} |\phi^{[1\ldots l-1]}_{\alpha_{l-1}}\rangle \otimes |i_l i_{l+1}\rangle \otimes |\phi^{[l+2\ldots M]}_{\alpha_{l+1}}\rangle\,, \tag{B.75}$$

$$\Rightarrow \Theta^{i_l i_{l+1}}_{\alpha_{l-1}\alpha_{l+1}} \equiv \sum_{\alpha_l} \lambda^{[l]}_{\alpha_{l-1}} \Gamma^{[l]i_l}_{\alpha_{l-1}\alpha_l} \lambda^{[l+1]}_{\alpha_l} \Gamma^{[l+1]i_{l+1}}_{\alpha_l \alpha_{l+1}} \lambda^{[l+2]}_{\alpha_{l+1}}. \tag{B.76}$$

Note that this differs from the object Θ defined in Vidal's original paper[5] in that we have also included $\lambda^{[l]}$ and $\lambda^{[l+2]}$. With this definition we can write the updated state (the state after action by $\hat{V}$) as

$$\hat{V}|\Psi\rangle = \sum_{\alpha_{l-1}\alpha_{l+1}; i_l, i_{l+1}} \tilde{\Theta}^{i_l i_{l+1}}_{\alpha_{l-1}\alpha_{l+1}} |\phi^{[1\ldots l-1]}_{\alpha_{l-1}}\rangle \otimes |i_l i_{l+1}\rangle \otimes |\phi^{[l+2\ldots M]}_{\alpha_{l+1}}\rangle \tag{B.77}$$

$$\tilde{\Theta}^{i_l i_{l+1}}_{\alpha_{l-1}\alpha_{l+1}} = \sum_{i'_l i'_{l+1}} V^{i_l i_{l+1}}_{i'_l i'_{l+1}} \Theta^{i'_l i'_{l+1}}_{\alpha_{l-1}\alpha_{l+1}} \tag{B.78}$$

$$= \sum_{\tilde{\alpha}_l} \lambda^{[l]}_{\alpha_{l-1}} \tilde{\Gamma}^{[l]i_l}_{\alpha_{l-1}\alpha_l} \tilde{\lambda}^{[l+1]}_{\alpha_l} \tilde{\Gamma}^{[l+1]i_{l+1}}_{\alpha_l \alpha_{l+1}} \lambda^{[l+1]}_{\alpha_{l+2}}. \tag{B.79}$$

Algorithmically, the procedure for a two-site operation is as follows: (1) form Θ from the current Γs and λs as in Eq.(B.76), (2) form $\tilde{\Theta}$ from $\hat{V}$ and Θ as in Eq.(B.78), (3) normalize such that $\sum_{\alpha,i,j,\gamma} \left|\tilde{\Theta}^{ij}_{\alpha\gamma}\right|^2 = 1$ and repack the normalized $\chi \times d \times d \times \chi$ rank-four tensor $\tilde{\Theta}$ into a $(\chi d) \times (\chi d)$ matrix $\tilde{\Theta}_{\mathrm{RS}}$, (4) perform a singular value decomposition (SVD) on this matrix, keeping only the χ largest singular values $\tilde{\lambda}^{[l+1]}_{\alpha_l}$, and (5) compute the updated tensors $\tilde{\Gamma}^{[l]}$ and $\tilde{\Gamma}^{[l+1]}$ from the matrix elements obtained via SVD as

$$\tilde{\Theta}_{\mathrm{RS}} \overrightarrow{\mathrm{SVD}}\, USV\,, \tag{B.80}$$

$$\tilde{\lambda}^{[l+1]}_{\alpha_l} = \frac{S_{\alpha_l}}{\sqrt{\sum_{\alpha=1}^{\chi}\left(S_{\alpha}\right)^2}}\,, \tag{B.81}$$

$$\tilde{\Gamma}^{[l]i_l}_{\alpha_{l-1}\alpha_l} = U_{(i_l-1)\chi+\alpha_{l-1},\alpha_l}/\lambda^{[l]}_{\alpha_{l-1}}\,, \tag{B.82}$$

$$\tilde{\Gamma}^{[l+1]i_{l+1}}_{\alpha_l\alpha_{l+1}} = V_{\alpha_l,(i_{l+1}-1)\chi+\alpha_{l+1}}/\lambda^{[l+2]}_{\alpha_{l+1}}\,. \tag{B.83}$$

Note that a two-site operation applied on site l updates the local tensors of site $l+1$ as well as those of site l. The numerical routines that comprise each step are (1) `FormTheta`, (2) `ThetaOperation`, (3) `ReshapeTheta`, (4) `SVDTruncation`, and (5) `FormLambda1`, `FormGamma1`, and `FormGamma2` for Eqs. (B.81), (B.82), and (B.83), respectively. The procedure listings for these are found in sections (1) B.9.8.4, (2) B.9.8.5, (3) B.9.8.6, (4) B.9.8.7, and (5) B.9.8.8, B.9.8.9, and B.9.8.10. The two-site operation defined by steps (1)-(5) is performed by `TwoSiteOp`, with procedure listing in Sec. B.9.8.11.

Note that step (4) involves discarding $\chi\,(d-1)$ of the $d\chi$ eigenvalues of $\tilde{\Theta}$. It is in this sense that the new local tensors $\tilde{\Gamma}$ and $\tilde{\lambda}$ are approximate. Because of the normalization of the Schmidt coefficients $\sum_{\alpha_l}^{\chi_S}\left(\lambda^{[l+1]}_{\alpha_l}\right)^2 = 1$, we can explicitly write down the truncation error due to this procedure, called the *Schmidt error* (also known as the *discarded weight* in the DMRG community):

$$\tau^S_l \equiv 1 - \sum_{\alpha_l}^{\chi}\left(\lambda^{[l+1]}_{\alpha_l}\right)^2\,. \tag{B.84}$$

This error depends explicitly on the degree of entanglement of the system, as measured by the Schmidt measure. It is in this sense that TEBD performs best on lowly spatially entangled systems. Note, however, that we are *not* minimizing the entanglement in the system but rather the “distance” (Frobenius norm) between our state in the subspace of limited entanglement and the true state of the full Hilbert space. For large enough χ the Schmidt error is identically zero, and so we refer to TEBD as being quasi-exact.

B.3.4.3 Two Site Operations in the Presence of an Abelian Symmetry

Let us now turn our attention to a two-site operation performed in the presence of a symmetry whose representations are one-dimensional. We call such symmetries *Abelian*. The fact that the representations are one dimensional implies that the associated quantum number (which labels each irreducible representation) also forms a group, and so composition of two irreducible representations with quantum numbers i and j, $\mathcal{D}\left(i\right)\otimes\mathcal{D}\left(j\right)$, is the irreducible representation with quantum number given by the group operation $i*j$, $\mathcal{D}\left(i*j\right)$.

The Abelian symmetry that we shall henceforth consider is total number conservation, with operator $\hat{N}=\bigoplus_{i=1}^{L}\hat{n}_i$. When the two-site operator $\hat{V}$ commutes with the total number operator of the two site block, acting with $\hat{V}$ will not produce states with a different total number of particles than the initial state. Thus, the object Θ defined to be

$$\Theta^{i_l i_{l+1}}_{\alpha_{l-1}\alpha_{l+1}}\equiv\sum_{\alpha_l}\lambda^{[l]}_{\alpha_{l-1}}\Gamma^{[l]i_l}_{\alpha_{l-1}\alpha_l}\lambda^{[l+1]}_{\alpha_l}\Gamma^{[l+1]i_{l+1}}_{\alpha_l\alpha_{l+1}}\lambda^{[l+2]}_{\alpha_{l+1}}. \tag{B.85}$$

can be reshaped into a $(\chi d)\times(\chi d)$ block diagonal matrix, with each block corresponding to a fixed number of particles on the left. To see this, we write

$$|\Psi\rangle=\sum_{\alpha_{l-1}\alpha_{l+1};i_l,i_{l+1}}\Theta^{i_l i_{l+1}}_{\alpha_{l-1}\alpha_{l+1}}|\phi^{[1\dots l-1]}_{\alpha_{l-1}}\rangle\otimes|i_l i_{l+1}\rangle\otimes|\phi^{[l+2\dots M]}_{\alpha_{l+1}}\rangle\,, \tag{B.86}$$

and note that, in order for $|\phi^{[1\dots l-1]}_{\alpha_{l-1}}\rangle\otimes|i_l i_{l+1}\rangle\otimes|\phi^{[l+2\dots M]}_{\alpha_{l+1}}\rangle$ to be an eigenstate of $\hat{N}$ we must have

$$N_L\left(\alpha_{l-1}\right)+N_S\left(i_l\right)+N_S\left(i_{l+1}\right)+N_R\left(\alpha_{l+1}\right)=N\,, \tag{B.87}$$

in an obvious notation. This is equivalent to the physically more transparent statement that the reduced density matrix defined by

$$\begin{aligned}\rho^{[1\ldots l]}&=\mathrm{Tr}_{(l+1)\ldots L}|\Psi\rangle\langle\Psi|\,, &\text{(B.88)}\\ &=\sum_{i_{l+1}i'_{l+1}}\sum_{\alpha_{l+1}\alpha'_{l+1}}\left[\sum_{\alpha_{l-1}i_l}\Theta^{i_li_{l+1}}_{\alpha_l\alpha_{l+1}}\left(\Theta^{i_li'_{l+1}}_{\alpha_l\alpha'_{l+1}}\right)^{\star}\right]|i_{l+1}\phi^{[l+2\ldots L]}_{\alpha_{l+1}}\rangle\langle i'_{l+1}\phi^{[l+2\ldots L]}_{\alpha_{l+1}]}|\,. &\text{(B.89)}\end{aligned}$$

is block diagonal, as the object in square brackets is nonzero iff $N_L\left(\alpha_{l-1}\right)+N_S\left(i_l\right)=N_S\left(i'_{l+1}\right)+N_L\left(\alpha'_{l+1}\right)$. This special structure allows us to perform a series of smaller singular value decompositions–one on each block of fixed number on the left of the bipartite splitting–which, in light of the cubic complexity class of SVD algorithms, amounts to significant speedup.

In order to form Θ consistent with number conservation, it is important to know the number associated with a given left (right) Schmidt vector $|\phi^{[1\ldots l]}_{\alpha_l}\rangle$ ($|\phi^{[l+1\ldots L]}_{\alpha_l}\rangle$) and also the total number in the onsite state $|i_l\rangle$. In the code, we store the number of the α_l^{th} left Schmidt vector as `LabelLeft(l)%vi(alpha)`, the number of the α_l^{th} right Schmidt vector as `LabelRight(l)%vi(alpha)`, and the number of the i_l^{th} onsite state as `Conserv%vi(i)` (see Sec. B.8.2.13). These vectors allow us to insert IF statements to enforce Eq. (B.87) when e. g. forming Θ. As an example, `LabelLeft` and `LabelRight` for the state $|010101\rangle$ are given as in Table B.2, which may be easily verified by counting the cumulative number to the left or right of the given link. If a particular Schmidt index is unused on a given `Label`, it is set to 10000 to distinguish it from being identically zero.

The procedure for a two-site operation that conserves the total number is (1) form Θ the current Γs and λs as in Eq.(B.76) subject to the number constraint Eq. (B.87), (2) form $\tilde{\Theta}$ from $\hat{V}$ and Θ as in Eq.(B.78) subject to the number con-

Table B.2: Number conserving labels for the state $|010101\rangle$.

link l	`LabelLeft(l)%vi(alpha)`	`LabelRight(l)%vi(alpha)`
1	$0\delta_{\alpha,1}$	$3\delta_{\alpha,1}$
2	$0\delta_{\alpha,1}$	$3\delta_{\alpha,1}$
3	$1\delta_{\alpha,1}$	$2\delta_{\alpha,1}$
4	$1\delta_{\alpha,1}$	$2\delta_{\alpha,1}$
5	$2\delta_{\alpha,1}$	$1\delta_{\alpha,1}$
6	$2\delta_{\alpha,1}$	$1\delta_{\alpha,1}$
7	$3\delta_{\alpha,1}$	$0\delta_{\alpha,1}$

straint Eq. (B.87), (3) normalize such that $\sum_{\alpha,i,j,\gamma}\left|\Theta^{ij}_{\alpha\gamma}\right|^2=1$, (4) find the maximum and minimum values of the total number to the left of the link separating the two sites (5) find the size of each block in the block-diagonal Θ, (6) find the on-site and Schmidt indices that correspond to the allowed values of the total number to the left of the link from (4), (7) repack the N_{Blocks} blocks of the block-diagonal $\tilde{\Theta}$ into a N_{Blocks} length vector of matrices $\tilde{\Theta}_i$, (8) perform an SVD on every $\tilde{\Theta}_i$, (9) gather all of the singular values into one vector and order them in non-decreasing fashion, (10) update the `Label` vectors with the new configuration, and (11) update the local tensors from the proper SVD results. The corresponding procedure listings are found in (1) Sec. B.9.8.18, (2) Sec. B.9.8.19, (3) Sec. B.9.8.20, (4) Sec. B.9.8.22, (5) Sec. B.9.8.23, (6) Sec. B.9.8.24 and Sec. B.9.8.25, (7) Sec. B.9.8.26, (8) Sec. B.9.8.27, (9) Sec. B.9.8.28, Sec. B.9.8.29, and Sec. B.9.8.30, (10) Sec. B.9.8.31 and Sec. B.9.8.32, and (11) Sec. B.9.8.33, Sec. B.9.8.34, and Sec. B.9.8.35.

B.3.4.4 Swapping Routines and the Canonical Form

In the above we have always taken for granted that the Γs are all expressed in terms of orthonormal bases and the λs are the Schmidt coefficients of the corresponding bipartition. Such a Vidal representation is a tensor network that is said to be in the *canonical form*[14]. Given a bipartition which is not in canonical form,

$$|\Psi\rangle = \sum_{\alpha} |\Phi^{[A]}_{\alpha}\rangle |\Phi^{[B]}_{\alpha}\rangle \,, \tag{B.90}$$

we restore it to canonical form as follows. We first form the matrix M^A, which has elements

$$M^A_{\alpha,\alpha'} \equiv \langle \Phi^{[A]}_{\alpha'} | \Phi^{[A]}_{\alpha} \rangle \tag{B.91}$$

$$= \sum_{\gamma} \left(\Gamma^{i_A}_{\gamma\alpha'}\right)^{\star} (\lambda_{\gamma})^2 \, \Gamma^{i_A}_{\gamma\alpha} \,, \tag{B.92}$$

where i_A is the local index of the site in A which is furthest to the right. We find an orthogonal matrix X and a vector of values S from performing an SVD on this matrix $M^A \overrightarrow{\text{svd}} USV$, $X_{\alpha\beta} = \sqrt{S_{\alpha}} U^{\star}_{\beta\alpha}$. The new set of vectors

$$|e_{\tau}\rangle = \sum_{\alpha} \left(X^{\dagger}\right)_{\tau\alpha} |\Phi^{[A]}_{\alpha}\rangle \tag{B.93}$$

form an orthonormal set. A similar procedure for B leads to a matrix Y defined as $M^B \overrightarrow{\text{svd}} USV$, $Y_{\alpha\beta} = \lambda_{\alpha} U_{\alpha\beta} \sqrt{S_{\beta}}$ and the orthonormal set

$$|f_{\eta}\rangle = \sum_{\alpha} Y_{\eta\alpha} |\Phi^{[B]}_{\alpha}\rangle \,. \tag{B.94}$$

The state may now be written as

$$|\Psi\rangle = \sum_{\tau\eta} \left(X^T Y\right)_{\tau\eta} |e_{\tau}\rangle |f_{\eta}\rangle \,, \tag{B.95}$$

which may be brought to canonical form using the SVD in the usual way

$$\left(X^T Y\right) = \sum_{k} \tilde{\Gamma}^A_{ik} \tilde{\lambda}_k \tilde{\Gamma}^B_{kj} \,, \tag{B.96}$$

$$\Rightarrow |\Psi\rangle = \sum_{k} \tilde{\lambda}_k |\tilde{\Phi}^{[A]}_k\rangle |\tilde{\Phi}^{[B]}_k\rangle \,, \tag{B.97}$$

$$|\tilde{\Phi}^{[A]}_k\rangle \equiv \sum_{i} \tilde{\Gamma}^A_{i,k} |i_A\rangle \,, \quad |\tilde{\Phi}^{[B]}_k\rangle \equiv \sum_{j} \tilde{\Gamma}^B_{j,k} |j_B\rangle. \tag{B.98}$$

When we initialize our state we use the canonical form and this form is preserved for unitary operations such as real time evolution, but two important operations destroy the canonical form.

The first operation which takes states to non-canonical form is imaginary time propagation. Imaginary time propagation involves non-unitary operations on the tensor network, and so causes nonorthogonality in the Γ bases. For small imaginary time steps the propagator is very close to unitary, and the buildup of nonorthogonality is small, but it will lead to unphysical results if not cared for properly. The tensor network may be restored to its canonical form using the routine `CanonicalFormAll` (Sec. B.9.11.6), or in the number conserving case, `CanonicalFormAllNC` (Sec. B.9.11.14). All imaginary time propagation routines restore the tensor network to canonical form after each imaginary time step.

The second operation which takes states to non-canonical form is deformation of the tensor network. The specific case of deformation we consider is that of "swapping" two sites such that the j^{th} Γ in the tensor network contains information for site $j+1$ and the $j+1^{\text{th}}$ Γ in the tensor network contains the information for site j. Such a deformation is the key to implementing periodic boundary conditions, and also allows for the simulation of ladders and systems with long-range interactions.

In the specific case of periodic boundary conditions, we need to apply a two-site operation to the pair of sites $(1, L)$ because there is a Hamiltonian matrix element between these two sites. TEBD only gives us a prescription for two-site operations on sites whose local tensors are neighboring, and so we must deform our tensor network to put site L next to site 1. To accomplish this, we use the swapping routines as in Figure B.5 [15]. We begin by forming Θ using sites $L-1$ and L

$$\Theta^{i_{L-1}i_L}_{\alpha_{L-2}\alpha_L} \equiv \sum_{\alpha_{L-1}} \lambda^{[L-2]}_{\alpha_{L-2}} \Gamma^{[L-1]i_{L-1}}_{\alpha_{L-2}\alpha_{L-1}} \lambda^{[L]}_{\alpha_{L-1}} \Gamma^{[L]i_L}_{\alpha_{L-1}\alpha_L} \lambda^{[L+1]}_{\alpha_L} \, . \tag{B.99}$$

We then swap the site indices of Θ: $\tilde{\Theta}^{ij}_{\alpha_{L-2}\alpha_L}=\Theta^{ji}_{\alpha_{L-2}\alpha_L}$ and then perform an SVD on $\tilde{\Theta}$ (reshaped into a two-tensor as above) to get the new local tensors $\Gamma^{[L](1)}$, $\tilde{\lambda}^{[L]}$, and $\tilde{\Gamma}^{[L-1]}$, which amounts to putting this new bipartite splitting into canonical form. Our tensor network has transformed from

$$\begin{aligned}&\sum_{i_1\dots i_L}\sum_{\alpha_0\dots\alpha_L}^{\chi_S}\lambda^{[1]}_{\alpha_0}\Gamma^{[1]i_1}_{\alpha_0\alpha_1}\lambda^{[2]}_{\alpha_1}\Gamma^{[2]i_2}_{\alpha_1\alpha_2}\lambda^{[3]}_{\alpha_2}\Gamma^{[3]i_3}_{\alpha_2\alpha_3}\dots\\&\times\lambda^{[L-1]}_{\alpha_{L-2}}\Gamma^{[L-1]i_{L-1}}_{\alpha_{L-2}\alpha_{L-1}}\lambda^{[L]}_{\alpha_{L-1}}\Gamma^{[L]i_L}_{\alpha_{L-1}\alpha_L}\lambda^{[L+1]}_{\alpha_L}|i_1i_2i_3\dots i_{L-1}i_L\rangle\end{aligned}\tag{B.100}$$

to

$$\begin{aligned}&\sum_{i_1\dots i_L}\sum_{\alpha_0\dots\alpha_L}^{\chi_S}\lambda^{[1]}_{\alpha_0}\Gamma^{[1]i_1}_{\alpha_0\alpha_1}\lambda^{[2]}_{\alpha_1}\Gamma^{[2]i_2}_{\alpha_1\alpha_2}\lambda^{[3]}_{\alpha_2}\Gamma^{[3]i_3}_{\alpha_2\alpha_3}\dots\\&\times\lambda^{[L-1]}_{\alpha_{L-2}}\Gamma^{[L](1)i_L}_{\alpha_{L-2}\alpha_{L-1}}\tilde{\lambda}^{[L]}_{\alpha_{L-1}}\tilde{\Gamma}^{[L-1]i_{L-1}}_{\alpha_{L-1}\alpha_L}\lambda^{[L+1]}_{\alpha_L}|i_1i_2i_3\dots i_{L-1}i_L\rangle\,;\end{aligned}\tag{B.101}$$

we have swapped the positions of $\Gamma^{[L-1]}$ and $\Gamma^{[L]}$ while keeping their interpretations the same, and then put this new state into canonical form. This process is carried out numerically via the procedure `Swapping` or, in the case of number conservation, `SwappingNC` which are listed in Sec. B.9.8.15 and Sec. B.9.8.37, respectively.

We repeat this process with the pair $(L-2,L)$ and so on until the sites 1 and L are adjacent in the tensor network. We may then apply a two-site operator to this pair of sites and then swap back successively so that the tensors are in their original positions. We note that the swapping process represents an approximation, as we are forming a best approximation to the tensor network with the sites swapped and this involves a truncation. Furthermore, the renormalization associated with this truncation causes a larger Schmidt error in the case of real time propagation. This coupled with the fact that simulations with periodic boundary conditions are more difficult to converge in χ (i.e. the states produced tend to be more highly entangled) imply that care is needed to ensure a simulation with periodic boundary conditions is properly converged.

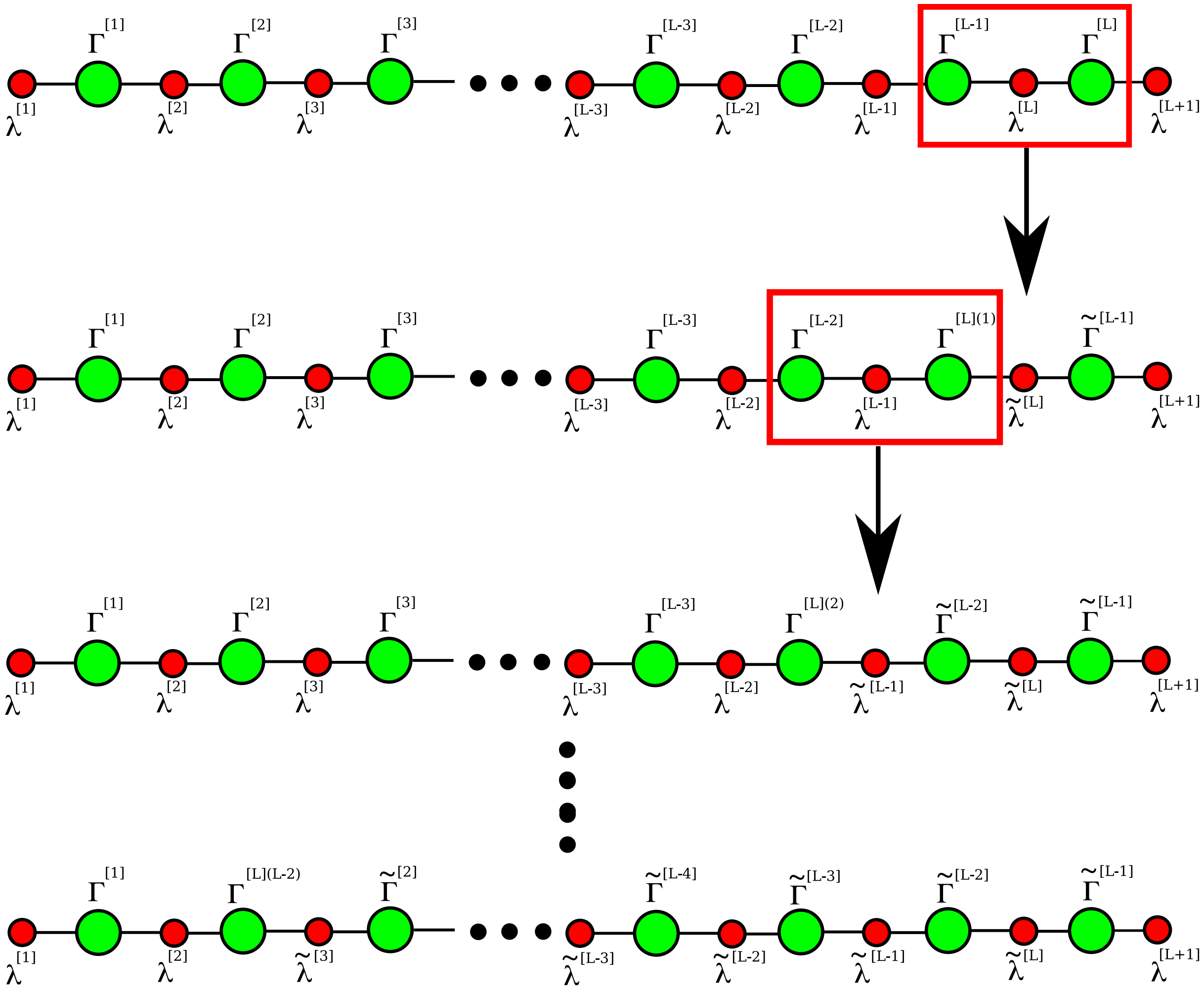


Figure B.5: Schematic of how to place site L next to site 1 in the tensor network using swapping routines. The red box and arrow represent application of a swapping routine to the boxed sites.

B.3.5 Time Propagation

B.3.5.1 The Second-Order Suzuki-Trotter Expansion for Open Boundary Conditions

We now consider the time evolution operator $\hat{U}\left(\delta t\right)\equiv\exp\left(-\frac{i}{\hbar}\hat{H}\delta t\right)$ which evolves the state $|\Psi\rangle$ according to the Hamiltonian $\hat{H}$ a time interval δt. We assume the Hamiltonian can be written as a direct sum of "local Hamiltonians" that act on at most two sites at a time (note that a one-site operation can be made into a two-site operation by a tensor product with the appropriate identity, see Sec. B.3.5.4). This is equivalent to saying that each local Hamiltonian acts on a given link l, and so we can decompose the total Hamiltonian into pieces that act on odd index links and even index links as

$$\hat{H}=\hat{H}_{\mathrm{odd}}+\hat{H}_{\mathrm{even}},\ \text{where } \hat{H}_{\mathrm{odd}}\equiv\sum_{\text{odd } l}\hat{H}_l \text{ and } \hat{H}_{\mathrm{even}}\equiv\sum_{\text{even } l}\hat{H}_l. \tag{B.102}$$

Our time evolution operator thus becomes

$$\hat{U}\left(\delta t\right)=\exp\left[-\frac{i\delta t}{\hbar}\left(\sum_{\text{odd } l}\hat{H}_l+\sum_{\text{even } l}\hat{H}_l\right)\right]. \tag{B.103}$$

We would like to write this as a sequence of two-site operators so that we can use the methods developed above, but unfortunately $\hat{H}_l$ and $\hat{H}_{l+1}$ do not, in general, commute, and so the time evolution operator will not factorize into two-site evolutions. The approximate answer to this trouble comes from the second order Suzuki-Trotter expansion [16], an approximate factorization that has an asymptotic error of order $\mathcal{O}\left(\delta t^3\right)$. The expansion is

$$
\begin{aligned}
e^{-i\hat{H}\delta t} &= e^{-i\hat{H}_{\mathrm{odd}}\delta t/2\hbar}e^{-i\hat{H}_{\mathrm{even}}\delta t/\hbar}e^{-i\hat{H}_{\mathrm{odd}}\delta t/2\hbar}+\mathcal{O}\left(\delta t^3\right)\,, \qquad \text{(B.104)}\\
e^{-i\hat{H}_{\mathrm{odd}}\delta t/2\hbar} &= \prod_{\text{odd } l} e^{-i\hat{H}_l\delta t/2\hbar}\,,\\
e^{-i\hat{H}_{\mathrm{even}}\delta t/\hbar} &= \prod_{\text{even } l} e^{-i\hat{H}_l\delta t/\hbar}\,.
\end{aligned}
$$

We perform a half timestep propagation over odd sites, a full timestep propagation over odd sites, and then another half timestep propagation over odd sites to complete a full timestep over the whole system. Because $\left[\hat{H}_l,\hat{H}_{l+j}\right]=0$ for $|j|\geq 2$, acting with the time evolution operator is now described as a sequence of two-site operations, and so the methods of Sec. B.3.4.2 and Sec. B.3.4.3 apply. TEBD will find the optimal truncation of the Hilbert space based on the current state of the system at each time step, and so it is referred to as being time-adaptive.

B.3.5.2 The Fifth-Order Suzuki-Trotter expansion for Open Boundary Conditions

A fifth-order decomposition of the time evolution operator is given by the *Forest-Ruth* formula

$$
\begin{aligned}
e^{-i\hat{H}\delta t} =& e^{-i\hat{H}_{\mathrm{odd}}\theta\delta t/2\hbar}e^{-i\hat{H}_{\mathrm{even}}\theta\delta t/\hbar}e^{-i\hat{H}_{\mathrm{odd}}(1-\theta)\delta t/2\hbar}e^{-i\hat{H}_{\mathrm{even}}(1-2\theta)\delta t\hbar}e^{-i\hat{H}_{\mathrm{odd}}(1-\theta)\delta t/2\hbar} \qquad \text{(B.105)}\\
&\times e^{-i\hat{H}_{\mathrm{even}}\theta\delta t/\hbar}e^{-i\hat{H}_{\mathrm{odd}}\theta\delta t/2\hbar}+\mathcal{O}\left(\delta t^5\right)\,,
\end{aligned}
$$

where the Forest-Ruth parameter $\theta\equiv 1/\left(2-2^{1/3}\right)$ and the exponentials of $\hat{H}_{\mathrm{odd}}$ and $\hat{H}_{\mathrm{even}}$ factorize as in Eq. (B.104). We see that application of this expansion involves 7 sweeps across the lattice instead of 3 as in the second order case. However, we may take coarser time steps $\delta t_5\sim(\delta t_2)^{3/5}$ and maintain the same accuracy, reducing the number of timesteps to reach a given time, which in turn reduces not only the computation time but also the accumulated Schmidt error. This makes the higher order expansion almost always worthwhile. One case in which the lower order expansion may be preferred is when the Hamiltonian is changing in time on a short timescale which limits the size of the trotter step. Over long times the Schmidt error will dominate and so the error due to the trotter expansion becomes irrelevant. In this case the second order expansion will give the same accuracy with a factor of two speedup over the fifth order expansion.

B.3.5.3 Suzuki-Trotter Expansions for Periodic Boundary Conditions

To perform simulations with periodic boundary conditions, we follow the method of [15]. We break the Hamiltonian into three parts: $\hat{H}=\hat{H}_{\mathrm{odd}}+\hat{H}_{\mathrm{even}}+\hat{H}_{\mathrm{edge}}$, where $\hat{H}_{\mathrm{odd}}$ and $\hat{H}_{\mathrm{even}}$ have their usual meanings and $\hat{H}_{\mathrm{edge}}$ is the part of the Hamiltonian which couples the last site to the first. It is clear that $\left[\hat{H}_{\mathrm{odd}},\hat{H}_{\mathrm{edge}}\right]\neq 0$ by construction, since the edge Hamiltonian acts on the first site which has odd parity by definition, but the edge Hamiltonian commutes with the even site Hamiltonian if L is even. To perform the trotter decomposition, we make the definitions $\hat{H}_{\mathrm{odd}}=A$, $\hat{H}_{\mathrm{even}}=B$, $\hat{H}_{\mathrm{edge}}=C$, and $-i\Delta t/\hbar=\delta$, and investigate $U\left(\delta\right)=\exp\left[\left(A+B+C\right)\delta\right]$. Writing $B+C=D$ and noting that $[A,D]\neq 0$, we can use the standard trotter decompositions to write

$$\begin{aligned} U\left(\delta\right) &= e^{A\delta/2}e^{D\delta}e^{A\delta/2}+\mathcal{O}\left(\delta^3\right) && \text{(B.106)}\\ &= e^{A\theta\delta/2}e^{D\theta\delta}e^{A(1-\theta)\delta/2}e^{D(1-2\theta)\delta}e^{A(1-\theta)\delta/2}e^{D\theta\delta}e^{A\theta\delta/2}+\mathcal{O}\left(\delta^5\right)\,. && \text{(B.107)}\end{aligned}$$

In the case where L is even we can immediately separate B and C from D to write

$$\begin{aligned} U\left(\delta\right) &= e^{A\delta/2}e^{B\delta}e^{C\delta}e^{A\delta/2}+\mathcal{O}\left(\delta^3\right) && \text{(B.108)}\\ &= e^{A\theta\delta/2}e^{B\theta\delta}e^{C\theta\delta}e^{A(1-\theta)\delta/2}e^{B(1-2\theta)\delta}e^{C(1-2\theta)\delta}e^{A(1-\theta)\delta/2}e^{B\theta\delta}e^{C\theta\delta}e^{A\theta\delta/2}+\mathcal{O}\left(\delta^5\right)\,. && \text{(B.109)}\end{aligned}$$

This involves 1 (3) swapping applications (Sec. B.3.4.4) and additional sweeps for the second (fifth) order routine. In the case where L is odd and B and C don't commute, we must apply another trotter decomposition of the same order to the exponentials involving D, yielding

$$U\left(\delta\right) = e^{A\delta/2}e^{B\delta/2}e^{C\delta}e^{B\delta/2}e^{A\delta/2} + \mathcal{O}\left(\delta^3\right) \tag{B.110}$$

$$\begin{aligned}
&= e^{A\theta\delta/2}\left[e^{B\theta^2\delta/2}e^{C\theta^2\delta}e^{B(1-\theta)\theta\delta/2}e^{C(1-2\theta)\theta\delta}e^{B(1-\theta)\theta\delta/2}e^{C\theta^2\delta}e^{B\theta^2\delta/2}\right]\\
&\quad\times e^{A(1-\theta)\delta/2}\left[e^{B\theta(1-2\theta)\delta/2}e^{C\theta(1-2\theta)\delta}e^{B(1-\theta)(1-2\theta)\delta/2}e^{C(1-2\theta)^2\delta}\right.\\
&\qquad\qquad\left.\times e^{B(1-\theta)(1-2\theta)\delta/2}e^{C\theta(1-2\theta)\delta}e^{B\theta(1-2\theta)\delta/2}\right]\\
&\quad\times e^{A(1-\theta)\delta/2}\left[e^{B\theta^2\delta/2}e^{C\theta^2\delta}e^{B(1-\theta)\theta\delta/2}e^{C(1-2\theta)\theta\delta}e^{B(1-\theta)\theta\delta/2}e^{C\theta^2\delta}e^{B\theta^2\delta/2}\right]e^{A\theta\delta/2}\\
&\quad+\mathcal{O}\left(\delta^5\right)\,,
\end{aligned} \tag{B.111}$$

In this case the number of swapping applications remains the same and only one more sweep is required for the second order case, but the fifth order case requires a total of 9 swapping applications and 25 sweeps.

It is clear that the number of different exponential operators differs based on the order of expansion, the type of boundary conditions, and whether the lattice has an even or odd number of sites. The routines `AllocateProp` and `DeallocateProp` listed in Sec. B.9.1.22 and B.9.1.23 will allocate and deallocate the propagators based on the appropriate global variables, respectively. The routine `ConstructPropagators`, when called with the syntax

`CALL ConstructPropagators(H, U, dt)` (Sec. B.9.1.24), will generate the required propagators. The old syntax

`CALL ConstructPropagators(H, U, dtodd, dteven)`, which is specific to the case of second order trotter with open boundary conditions, is supported for compatibility with past versions, but will be deleted in a future version.

B.3.5.4 Putting a Hamiltonian in TEBD Form

A fundamental assumption in the discussion of time evolution above is that the Hamiltonian can be written in the form of Eq. (B.102), as a sum of two-site operations. To include a one-site operator $\hat{A}$ we write it in terms of two two-site operations as

$\frac{1}{2}\left(\hat{A}\otimes\hat{1}+\hat{1}\otimes\hat{A}\right)$. In addition, for box boundary conditions the first and last site are only updated once per time step (see Eq. (B.104)), and so they require an additional $\frac{1}{2}\hat{A}\otimes\hat{1}$ or $\frac{1}{2}\hat{1}\otimes\hat{A}$, respectively. As an example, the list of two-site Hamiltonians representing the Bose-Hubbard Hamiltonian (Sec. B.3.7.2) in TEBD form, H(:), is given by

$$H_i = -J\,\hat{t} + \frac{U}{4}\left\{\left[\hat{n}\left(\hat{n}-\hat{1}\right)\right]\otimes\hat{1}+\hat{1}\otimes\left[\hat{n}\left(\hat{n}-\hat{1}\right)\right]\right\} + V\hat{n}\otimes\hat{n} + \frac{\epsilon_i}{2}\left[\hat{n}\otimes\hat{1}+\hat{1}\otimes\hat{n}\right]\,, \tag{B.112}$$

$$H_1 = H_1 + \frac{U}{4}\left[\hat{n}\left(\hat{n}-\hat{1}\right)\right]\otimes\hat{1} + \frac{\epsilon_1}{2}\hat{n}\otimes\hat{1}\,, \tag{B.113}$$

$$H_{L-1} = H_{L-1} + \frac{U}{4}\hat{1}\otimes\left[\hat{n}\left(\hat{n}-\hat{1}\right)\right] + \frac{\epsilon_{L-1}}{2}\hat{1}\otimes\hat{n}\,, \tag{B.114}$$

where $i = 1, 2, ..., L-1$ is the site index. In the case of periodic boundary conditions, H_L holds the two site operators which act on sites L and 1.

B.3.6 Observables

The expectation value of an observable $\hat{A}$ in the density matrix formalism is

$$\langle\hat{A}\rangle = \mathrm{Tr}\left(\hat{\rho}\hat{A}\right)\,, \tag{B.115}$$

where the density matrix $\hat{\rho}$ defines the state in the Hilbert space upon which the operator $\hat{A}$ acts. In the following sections we describe how to calculate observables in the Vidal representation.

B.3.6.1 Expectation Values of Single-Site Operators

For single-site observables, we calculate the expectation value on the l^{th} site as

$$\langle\hat{A}_l\rangle = \mathrm{Tr}\left(\hat{\rho}_l\hat{A}\right)\,, \tag{B.116}$$

where $\hat{\rho}_l$ is the reduced density matrix obtained by tracing over all sites but l. In the full many-body space this is achieved through

$$\hat{\rho} = |\Psi\rangle\langle\Psi| = \sum_{i_l, i'_l} \left(\sum_{i_1,\dots,i_{l-1},i_{l+1},\dots,i_L} c^{\star}_{i_1\dots i'_l\dots i_L} c_{i_1\dots i_l\dots i_L} \right) |i_l\rangle\langle i'_l| \tag{B.117}$$

$$\Rightarrow (\hat{\rho}_l)_{i'_l i_l} = \sum_{i_1,\dots,i_{l-1},i_{l+1},\dots,i_L} c^{\star}_{i_1\dots i'_l\dots i_L} c_{i_1\dots i_l\dots i_L}\,. \tag{B.118}$$

To obtain this object in the Vidal representation, we first decompose the state to isolate the single site $|i_l\rangle$

$$|\Psi\rangle = \sum_{i_l;\alpha_{l-1}\alpha_l} |\phi^{[1\dots l-1]}_{\alpha_{l-1}}\rangle \left[\lambda^{[l]}_{\alpha_{l-1}} \Gamma^{[l]i_l}_{\alpha_{l-1}\alpha_l} \lambda^{[l+1]}_{\alpha_l} |i_l\rangle \right] |\phi^{[l+1\dots M]}_{\alpha_l}\rangle\,, \tag{B.119}$$

and then take the outer product and trace over all sites but l to obtain

$$(\hat{\rho}_l)_{i_l, i'_l} = \sum_{\alpha_{l-1}\alpha_l} \lambda^{[l]}_{\alpha_{l-1}} \left(\Gamma^{[l]i'_l}_{\alpha_{l-1}\alpha_l} \right)^{\star} \lambda^{[l+1]}_{\alpha_l} \lambda^{[l]}_{\alpha_{l-1}} \Gamma^{[l]i_l}_{\alpha_{l-1}\alpha_l} \lambda^{[l+1]}_{\alpha_l}. \tag{B.120}$$

The single-site density matrix for a particular site is numerically generated using the procedure `FormSingleSiteRho` in Sec. B.9.9.1. The procedure `SingleSiteDensityMatrix` discussed in Sec. B.9.9.2 generates a list of single-site density matrices; one for each site. The procedure `OneSiteExpVal` discussed in Sec. B.9.9.3 will compute the expectation value of a general one-site observable at each site. The procedure `OneSiteVar` discussed in Sec. B.9.9.4 will compute the variance of a general one-site observable at each site. In addition, many common single-site observables have their own expectation value routines. See the procedure listings for `observables_module` in Sec. B.9.9 for a full account.

B.3.6.2 Expectation Values of Two-Site Operators: The General Case for Bosons

If we now wish to calculate the expectation value of a two-site observable $\hat{B}$ at sites k and l, we need to calculate

$$\langle \hat{B} \rangle_{kl} = \mathrm{Tr}\left(\hat{\rho}_{kl} \hat{B} \right) \tag{B.121}$$

where $\hat{\rho}_{kl}$ is the reduced density matrix obtained by tracing over all sites but k and l, and $\hat{B}$ is an operator in the Kronecker product representation spanned by $|i_k i_l\rangle$. For example, if we wish to calculate $\langle \hat{n}_k \hat{n}_l \rangle$ then $\hat{B} = \hat{n} \otimes \hat{n}$.

To calculate the two-site reduced density matrix we first assume, without loss of generality, that $l < k$ so that we can write our Vidal decomposed state as

$$\begin{aligned} |\Psi\rangle = &\sum_{i_l \ldots i_k} \sum_{\alpha_{l-1} \ldots \alpha_k} \lambda^{[l]}_{\alpha_{l-1}} \Gamma^{[l] i_l}_{\alpha_{l-1}\alpha_l} \lambda^{[l+1]}_{\alpha_l} \ldots \lambda^{[k]}_{\alpha_{k-1}} \Gamma^{[k] i_k}_{\alpha_{k-1}\alpha_k} \lambda^{[k+1]}_{\alpha_k} \\ &\times |\phi^{[1\ldots(l-1)]}_{\alpha_{l-1}}\rangle \otimes |i_l \ldots i_k\rangle \otimes |\phi^{[(k+1)\ldots M]}_{\alpha_k}\rangle \,. \end{aligned} \tag{B.122}$$

The density operator associated with this state is

$$\begin{aligned} |\Psi\rangle\langle\Psi| = &\sum_{\substack{i_l \ldots i_k \\ i'_l \ldots i'_k}} \sum_{\substack{\alpha_{l-1} \ldots \alpha_k \\ \alpha'_{l-1} \ldots \alpha'_k}} \left(\lambda^{[l]}_{\alpha_{l-1}} \Gamma^{[l] i_l}_{\alpha_{l-1}\alpha_l} \lambda^{[l+1]}_{\alpha_l} \ldots \lambda^{[k]}_{\alpha_{k-1}} \Gamma^{[k] i_k}_{\alpha_{k-1}\alpha_k} \lambda^{[k+1]}_{\alpha_k} \right) \\ &\times \left(\lambda^{[l]}_{\alpha'_{l-1}} \Gamma^{[l] i'_l}_{\alpha'_{l-1}\alpha'_l}{}^{\star} \lambda^{[l+1]}_{\alpha'_l} \ldots \lambda^{[k]}_{\alpha'_{k-1}} \Gamma^{[k] i'_k}_{\alpha'_{k-1}\alpha'_k}{}^{\star} \lambda^{[k+1]}_{\alpha_k} \right) \\ &\times |\phi^{[1\ldots(l-1)]}_{\alpha_{l-1}}\rangle \otimes |i_l \ldots i_k\rangle \otimes |\phi^{[(k+1)\ldots M]}_{\alpha_k}\rangle \langle \phi^{[1\ldots(l-1)]}_{\alpha'_{l-1}}| \otimes \langle i'_l \ldots i'_k| \otimes \langle \phi^{[(k+1)\ldots M]}_{\alpha'_k}| \,. \end{aligned} \tag{B.123}$$

Tracing over all sites except l and k and using the orthonormality of the Schmidt vectors yields

$$\begin{aligned} \mathrm{Tr}\left(|\Psi\rangle\langle\Psi|\right) = &\sum_{\substack{i_l, i_k \\ i'_l i'_k}} \sum_{i_{l+1} \ldots i_{k-1}} \sum_{\substack{\alpha_{l-1} \ldots \alpha_k \\ \alpha'_l \ldots \alpha'_{k-1}}} |i_l i_k\rangle\langle i'_l i'_k| \\ &\times \left(\lambda^{[l]}_{\alpha_{l-1}} \Gamma^{[l] i_l}_{\alpha_{l-1}\alpha_l} \lambda^{[l+1]}_{\alpha_l} \Gamma^{[l+1] i_{l+1}}_{\alpha_l \alpha_{l+1}} \lambda^{[l+2]}_{\alpha_{l+1}} \ldots \lambda^{[k-1]}_{\alpha_{k-2}} \Gamma^{[k-1] i_{k-1}}_{\alpha_{k-2}\alpha_{k-1}} \lambda^{[k]}_{\alpha_{k-1}} \Gamma^{[k] i_k}_{\alpha_{k-1}\alpha_k} \lambda^{[k+1]}_{\alpha_k} \right) \\ &\times \left(\lambda^{[l]}_{\alpha_{l-1}} \Gamma^{[l] i'_l}_{\alpha_{l-1}\alpha'_l}{}^{\star} \lambda^{[l+1]}_{\alpha'_l} \Gamma^{[l+1] i_{l+1}}_{\alpha'_l \alpha'_{l+1}}{}^{\star} \lambda^{[l+2]}_{\alpha'_{l+1}} \ldots \lambda^{[k-1]}_{\alpha'_{k-2}} \Gamma^{[k-1] i_{k-1}}_{\alpha'_{k-2}\alpha'_{k-1}}{}^{\star} \lambda^{[k]}_{\alpha'_{k-1}} \Gamma^{[k] i'_k}_{\alpha'_{k-1}\alpha_k}{}^{\star} \lambda^{[k+1]}_{\alpha_k} \right) \,. \end{aligned} \tag{B.124}$$

From which we can read off the two-site reduced density matrix

$$
\begin{aligned}
\left(\hat{\rho}_{lk}\right)_{i_l i_k, i'_l i'_k} = & \sum_{i_{l+1}\ldots i_{k-1}} \sum_{\substack{\alpha_{l-1}\ldots\alpha_k \\ \alpha'_l \ldots \alpha'_{k-1}}} \\
& \times \left(\lambda^{[l]}_{\alpha_{l-1}} \Gamma^{[l]i_l}_{\alpha_{l-1}\alpha_l} \lambda^{[l+1]}_{\alpha_l} \Gamma^{[l+1]i_{l+1}}_{\alpha_l \alpha_{l+1}} \lambda^{[l+2]}_{\alpha_{l+1}} \ldots \lambda^{[k-1]}_{\alpha_{k-2}} \Gamma^{[k-1]i_{k-1}}_{\alpha_{k-2}\alpha_{k-1}} \lambda^{[k]}_{\alpha_{k-1}} \Gamma^{[k]i_k}_{\alpha_{k-1}\alpha_k} \lambda^{[k+1]}_{\alpha_k}\right) \\
& \times \left(\lambda^{[l]}_{\alpha_{l-1}} {\Gamma^{[l]i'_l}_{\alpha_{l-1}\alpha'_l}}^{\star} \lambda^{[l+1]}_{\alpha'_l} {\Gamma^{[l+1]i_{l+1}}_{\alpha'_l \alpha'_{l+1}}}^{\star} \lambda^{[l+2]}_{\alpha'_{l+1}} \ldots \lambda^{[k-1]}_{\alpha'_{k-2}} {\Gamma^{[k-1]i_{k-1}}_{\alpha'_{k-2}\alpha'_{k-1}}}^{\star} \lambda^{[k]}_{\alpha'_{k-1}} {\Gamma^{[k]i'_k}_{\alpha'_{k-1}\alpha_k}}^{\star} \lambda^{[k+1]}_{\alpha_k}\right) .
\end{aligned}
\tag{B.125}
$$

Numerically, we calculate the two-site reduced density matrix as follows. We begin by defining an initial Θ at the k^{th} site as

$$
\Theta^{i_k i'_k}_{\alpha_{k-1}\alpha'_{k-1}} = \sum_{\alpha_k} \lambda^{[k]}_{\alpha_{k-1}} \Gamma^{[k]i_k}_{\alpha_{k-1}\alpha_k} \lambda^{[k+1]}_{\alpha_k} \lambda^{[k+1]}_{\alpha_k} {\Gamma^{[k]i'_k}_{\alpha'_{k-1}\alpha_k}}^{\star} \lambda^{[k]}_{\alpha'_{k-1}} . \tag{B.126}
$$

This is performed in `ThetaKernal`, listed in Sec. B.9.9.9. We then perform

$$
\Theta^{i_k i'_k}_{\alpha_{k-2}\alpha'_{k-2}} = \sum_{i_{k-1}} \sum_{\alpha_{k-1}\alpha'_{k-1}} \lambda^{[k-1]}_{\alpha_{k-2}} \Gamma^{[k-1]i_{k-1}}_{\alpha_{k-2}\alpha_{k-1}} \Theta^{i_k i'_k}_{\alpha_{k-1}\alpha'_{k-1}} {\Gamma^{[k-1]i_{k-1}}_{\alpha'_{k-2}\alpha'_{k-1}}}^{\star} \lambda^{[k-1]}_{\alpha'_{k-2}} \tag{B.127}
$$

repeatedly until $\Theta^{i_k i'_k}_{\alpha_l \alpha'_l}$ is obtained. Each application of Eq. (B.127) corresponds to a call of the procedure `ThetaNext`, listed in Sec. B.9.9.10. Once $\Theta^{i_k i'_k}_{\alpha_l \alpha'_l}$ is obtained, we form the two-site reduced density matrix as

$$
\left(\rho_{lk}\right)_{i_l i_k, i'_l i'_k} = \sum_{\alpha_{l-1}\alpha_l\alpha'_l} \lambda^{[l]}_{\alpha_{l-1}} \Gamma^{[l]i_l}_{\alpha_{l-1}\alpha_l} \Theta^{i_k i'_k}_{\alpha_l \alpha'_l} {\Gamma^{[l]i'_l}_{\alpha_{l-1}\alpha'_l}}^{\star} \lambda^{[l]}_{\alpha_{l-1}} , \tag{B.128}
$$

which, of course, agrees with Eq. (B.125). This last step of forming $\hat{\rho}_{lk}$ from $\Theta^{i_k i'_k}_{\alpha_l \alpha'_l}$ is performed in the procedure `TwoSiteRho` listed in Sec. B.9.9.11. The procedure `TwoSiteExpVal` listed in Sec. B.9.9.12 computes the two-site expectation values of the specified operator on all possible two-site arrangements, and returns the values as a `systemSize`×`systemSize` matrix. In addition, many common two-site observables have their own expectation value routines. See the procedure listings for `observables_module` in Sec. B.9.9 for a full listing.

B.3.6.3 Expectation Values of Two-Site Operators: The General Case for Fermions

The fermionic canonical commutation relations

$$\left\{\hat{a}_i, \hat{a}_j^\dagger\right\} = \delta_{ij}\,, \quad \left\{\hat{a}_i, \hat{a}_j\right\} = \left\{\hat{a}_i^\dagger, \hat{a}_j^\dagger\right\} = 0 \tag{B.129}$$

impose a set of nonlocal conditions on the creation and destruction operators. Because all operators in TEBD are local, we can not impose this constraint directly through the operator. The way that we satisfy the anti-commutation relations is by defining a local operator called `Fermiphase_op` as

$$\hat{\theta}_i = (-1)^{\hat{n}_i} \tag{B.130}$$

and then defining the nonlocal creation operators to be the string of local operators

$$\hat{\hat{a}}_i = \prod_{j>i} \hat{\theta}_j \hat{a}_i \tag{B.131}$$

where $\hat{a}_i$ is an operator which displays local anti-commutation properties. This formulation should be compared with the well-known Jordan-Wigner transformation which induces off-site anti-commutation relations by counting the cumulative number of fermions on either side of a particular site in a similar manner. Numerically we include the Fermi phase by introducing an additional step in the formation of Θ. After calling `ThetaKernal` to form

$$\Theta^{i_k i'_k}_{\alpha_{k-1}\alpha'_{k-1}} = \sum_{\alpha_k} \lambda^{[k]}_{\alpha_{k-1}} \Gamma^{[k]i_k}_{\alpha_{k-1}\alpha_k} \lambda^{[k+1]}_{\alpha_k} \lambda^{[k+1]}_{\alpha_k} {\Gamma^{[k]i_k}_{\alpha_{k-1}\alpha_k}}^{\star} \lambda^{[k]}_{\alpha'_{k-1}}\,, \tag{B.132}$$

we then act with the Fermi phase operator on the lower Γ:

$$P^{[k-1]} = \hat{\theta}_{k-1} \Gamma^{[k-1]} \tag{B.133}$$

and then use `ThetaNext` to generate

$$\Theta^{i_k i'_k}_{\alpha_{k-2}\alpha'_{k-2}} = \sum_{i_{k-1}} \sum_{\alpha_{k-1}\alpha'_{k-1}} \lambda^{[k-1]}_{\alpha_{k-2}} P^{[k-1]i_{k-1}}_{\alpha_{k-2}\alpha_{k-1}} \Theta^{i_k i'_k}_{\alpha_{k-1}\alpha'_{k-1}} {\Gamma^{[k-1]i_{k-1}}_{\alpha'_{k-2}\alpha'_{k-1}}}^{\star} \lambda^{[k-1]}_{\alpha'_{k-2}} . \tag{B.134}$$

This process is repeated with lower-index Γs until we reach the desired site and the two-site reduced density matrix is formed as

$$(\rho_{lk})_{i_l i_k, i'_l i'_k} = \sum_{\alpha_{l-1}\alpha_l\alpha'_l} \lambda^{[l]}_{\alpha_{l-1}} P^{[l]i_l}_{\alpha_{l-1}\alpha_l} \Theta^{i_k i'_k}_{\alpha_l \alpha'_l} {\Gamma^{[l]i_l}_{\alpha_{l-1}\alpha'_l}}^{\star} \lambda^{[l]}_{\alpha_{l-1}} , \tag{B.135}$$

via `TwoSiteRho`.

Note that this procedure only applies to two-site operators whose local action involves a product of an odd number of Fermi operators. Operators formed from even numbers of local Fermi operators, e. g. $\langle \hat{n}_i \hat{n}_j \rangle$ have no additional phases and the calculation is identical to the bosonic case.

B.3.6.4 Expectation Values of Two-Site Operators: Special Case of Tensor Product of One-Site Operators

In this section we shall describe how the procedure for computing the expectation of a two-site observable may be improved when the two-site operator $\hat{V}$ is a tensor product of one-site observables $\hat{V} = \hat{A} \otimes \hat{B}$, where $\hat{A}$ acts on site l and $\hat{B}$ on site k. We return to the density operator from above

$$\begin{aligned} |\Psi\rangle\langle\Psi| = &\sum_{\substack{i_l \dots i_k \\ i'_l \dots i'_k}} \sum_{\substack{\alpha_{l-1} \dots \alpha_k \\ \alpha'_{l-1} \dots \alpha'_k}} \left(\lambda^{[l]}_{\alpha_{l-1}} \Gamma^{[l]i_l}_{\alpha_{l-1}\alpha_l} \lambda^{[l+1]}_{\alpha_l} \dots \lambda^{[k]}_{\alpha_{k-1}} \Gamma^{[k]i_k}_{\alpha_{k-1}\alpha_k} \lambda^{[k+1]}_{\alpha_k} \right) \\ &\times \left(\lambda^{[l]}_{\alpha'_{l-1}} {\Gamma^{[l]i'_l}_{\alpha'_{l-1}\alpha'_l}}^{\star} \lambda^{[l+1]}_{\alpha'_l} \dots \lambda^{[k]}_{\alpha'_{k-1}} {\Gamma^{[k]i'_k}_{\alpha'_{k-1}\alpha'_k}}^{\star} \lambda^{[k+1]}_{\alpha_k} \right) \\ &\times |\phi^{[1\dots(l-1)]}_{\alpha_{l-1}}\rangle \otimes |i_1 \dots i_k\rangle \otimes |\phi^{[(k+1)\dots M]}_{\alpha_k}\rangle \langle \phi^{[1\dots(l-1)]}_{\alpha'_{l-1}}| \otimes \langle i'_1 \dots i'_k| \otimes \langle \phi^{[(k+1)\dots M]}_{\alpha'_k}| . \end{aligned} \tag{B.136}$$

Acting on this state with the given operator, we have

$$
\begin{aligned}
\hat{V}|\Psi\rangle\langle\Psi| = \sum_{\substack{i_l \ldots i_k \\ i'_l \ldots i'_k}} \sum_{\substack{\alpha_{l-1} \ldots \alpha_k \\ \alpha'_{l-1} \ldots \alpha'_k}} & \left(\lambda^{[l]}_{\alpha_{l-1}} \lambda^{[l]}_{\alpha'_{l-1}} \left[\sum_{j_l} A_{i_l j_l} \Gamma^{[l]j_l}_{\alpha_{l-1}\alpha_l} \right] {\Gamma^{[l]i'_l}_{\alpha'_{l-1}\alpha'_l}}^{\star} \right) \\
& \times \left(\lambda^{[l+1]}_{\alpha_l} \lambda^{[l+1]}_{\alpha'_l} {\Gamma^{[l+1]i_{l+1}}_{\alpha'_l \alpha'_{l+1}}}^{\star} \Gamma^{[l+1]i_{l+1}}_{\alpha_l \alpha_{l+1}} \right) \ldots \\
& \times \left(\lambda^{[k]}_{\alpha_{k-1}} \lambda^{[k]}_{\alpha'_{k-1}} \left[\sum_{j_k} B_{i_k j_k} \Gamma^{[k]j_k}_{\alpha_{k-1}\alpha_k} \right] {\Gamma^{[k]i'_k}_{\alpha'_{k-1}\alpha'_k}}^{\star} \right) \lambda^{[k]}_{\alpha_k} \lambda^{[k]}_{\alpha'_k} \\
& \times |\phi^{[1\ldots(l-1)]}_{\alpha_{l-1}}\rangle \otimes |i_1 \ldots i_k\rangle \otimes |\phi^{[(k+1)\ldots M]}_{\alpha_k}\rangle \langle \phi^{[1\ldots(l-1)]}_{\alpha'_{l-1}}| \otimes \langle i'_1 \ldots i'_k| \otimes \langle \phi^{[(k+1)\ldots M]}_{\alpha'_k}| \, .
\end{aligned}
\tag{B.137}
$$

Taking the trace and again using Schmidt vector orthonormality, we have

$$
\langle \hat{V} \rangle = \mathrm{Tr}\left[\hat{V}|\Psi\rangle\langle\Psi| \right] = \sum_{\substack{\alpha_{l-1} \ldots \alpha_k \\ \alpha'_l \ldots \alpha'_{k-1}}} \left(\tilde{G}_l \right)^{\alpha_{l-1}\alpha'_l}_{\alpha_{l-1}\alpha_l} \left(G_{l+1} \right)^{\alpha'_l \alpha'_{l+1}}_{\alpha_l \alpha_{l+1}} \ldots \left(\tilde{G}_k \right)^{\alpha'_{k-1}\alpha_k}_{\alpha_{k-1}\alpha_k} \left(\lambda^{[k]}_{\alpha_k} \right)^2 \, ,
\tag{B.138}
$$

where we have defined

$$
\left(G_p \right)^{\alpha'_{p-1}\alpha'_p}_{\alpha_{p-1}\alpha_p} \equiv \sum_{i_p} \lambda^{[p]}_{\alpha_{p-1}} \lambda^{[p]}_{\alpha'_{p-1}} {\Gamma^{[p]i_p}_{\alpha'_{p-1}\alpha'_p}}^{\star} \Gamma^{[p]i_p}_{\alpha_{p-1}\alpha_p}
\tag{B.139}
$$

$$
\left(\tilde{G}_p \right)^{\alpha'_{p-1}\alpha'_p}_{\alpha_{p-1}\alpha_p} \equiv \sum_{i_p i'_p} \lambda^{[p]}_{\alpha_{p-1}} \lambda^{[p]}_{\alpha'_{p-1}} \left[\sum_{j_p} C_{i_p j_p} \Gamma^{[p]j_p}_{\alpha_{p-1}\alpha_p} \right] {\Gamma^{[p]i_p}_{\alpha'_{p-1}\alpha'_p}}^{\star} \, ,
\tag{B.140}
$$

C being the appropriate one-site operator from above. The improvement comes from the fact that we no longer need to carry around the site indices i_p, which becomes especially important in parallel applications. Most two-site operations are of this form and so the more efficient routines described here are preferred over the routines described in Sec. B.3.6.2. The one common exception is the expectation of the Hamiltonian, which is rarely separable into a tensor product of one-site operators.

We perform $\langle \hat{A}\hat{B} \rangle$ numerically by the above in the following way. We first act with $\hat{B}$ on $\Gamma^{[k]}$

$$Q^{[k]} = \hat{B}\Gamma^{[k]} \tag{B.141}$$

and then form the object

$$G^{[k]}_{\alpha_k\beta_k} = \sum_{\beta' i} \left(\lambda^{[k+1]}_{\beta'}\right)^2 Q^{[k]i_k\star}_{\beta_k\beta'} \Gamma^{[k]i_k}_{\alpha_k\beta_k} \tag{B.142}$$

via a call to the procedure `GKernal`, listed in Sec. B.9.9.5. We then form the object $G^{[k-1]}$ as

$$G^{[k-1]}_{\alpha_{k-1}\beta_{k-1}} = \sum_{\alpha'\beta' i} \lambda^{[k]}_{\alpha'}\lambda^{[k]}_{\beta'} P^{[k-1]i_{k-1}\star}_{\beta_{k-1}\beta'} \Gamma^{[k-1]i_{k-1}}_{\alpha_{k-1}\alpha'} G^{[k]}_{\alpha'\beta'} \tag{B.143}$$

where $P^{[k-1]} = \Gamma^{[k-1]}$ for bosonic systems or $\hat{A}$ and $\hat{B}$ consisting of even numbers of local Fermi operators and $P^{[k-1]} = \hat{\theta}_{k-1}\Gamma^{[k-1]}$ where $\hat{\theta}$ is the Fermi phase operator for operators with Fermi phases. This operation is performed in `GNext`, listed in Sec. B.9.9.6. When the desired site has been reached the tensor is contracted as

$$\left[\langle \hat{A}\hat{B}\rangle\right]_{jk} = \sum_{\alpha\beta\beta' i} \left(\lambda^{[j]}_{\alpha}\right)^2 P^{[j]i\star}_{\beta'\beta} \Gamma^{[j]i}_{\beta'\alpha} \lambda^{[j+1]}_{\alpha}\lambda^{[j+1]}_{\beta} G^{[j+1]}_{\alpha\beta} \tag{B.144}$$

where $P^{[j]} = \hat{A}\Gamma^{[j]}$ for Fermi phase-less systems and $P^{[j]} = \hat{A}\hat{\theta}_j\Gamma^{[j]}$ for systems with a Fermi phase. This is performed in `GContraction`, listed in Sec. B.9.9.7. The procedure `TwoSiteExpValG` listed in Sec. B.9.9.8 computes the two-site expectation values of the specified operator on all possible two-site arrangements, and returns the values as a `systemSize`×`systemSize` matrix.

B.3.6.5 Expectation Values of N-Site Operators

Higher-order expectation values are a straightforward extension of the ideas of the last section and will be incorporated as need arises in a later version. The general method is described in [8]. The one special case that we do consider in the code is that of the overlap between two states $|\tilde{\psi}\rangle$ and $|\psi\rangle$. Denoting the Vidal decomposition tensors of the respective states as $\left\{\tilde{\Gamma},\tilde{\lambda}\right\}$ and $\{\Gamma,\lambda\}$, the inner product is

$$\begin{aligned}
\langle\tilde{\psi}|\psi\rangle &= \sum_{i_1,\dots,i_L}\left(\sum_{\beta_1,\dots,\beta_{L-1}}\tilde{\Gamma}^{[1]i_1}_{\beta_1}\tilde{\lambda}^{[2]}_{\beta_1}\tilde{\Gamma}^{[2]i_2}_{\beta_1\beta_2}\dots\tilde{\Gamma}^{[L]i_L}_{\beta_{L-1}}\right)^{\star} && \text{(B.145)}\\
&\quad\times\left(\sum_{\alpha_1,\dots\alpha_{L-1}}\Gamma^{[1]i_1}_{\alpha_1}\lambda^{[2]}_{\alpha_1}\Gamma_{\alpha_2,\alpha_3}\dots\Gamma^{[L]i_L}_{\alpha_{L-1}}\right)\\
&= \sum_{\substack{\beta_1,\dots,\beta_{L-1}\\ \alpha_1,\dots,\alpha_{L-1}}}\left(\sum_{i_1}\tilde{\lambda}^{[2]}_{\beta_1}\tilde{\Gamma}^{\star[1]i_1}_{\beta_1}\Gamma^{[1]i_1}_{\alpha_1}\lambda^{[2]}_{\alpha_1}\right)\left(\sum_{i_2}\tilde{\lambda}^{[3]}_{\beta_2}\tilde{\Gamma}^{\star[2]i_2}_{\beta_1\beta_2}\Gamma^{[2]i_2}_{\alpha_1\alpha_2}\lambda^{[3]}_{\alpha_2}\right)\\
&\quad\times\left(\sum_{i_3}\tilde{\lambda}^{[4]}_{\beta_3}\tilde{\Gamma}^{\star[3]i_3}_{\beta_2\beta_3}\Gamma^{[3]i_3}_{\alpha_2\alpha_3}\lambda^{[4]}_{\alpha_3}\right)\dots\left(\sum_{i_L}\tilde{\Gamma}^{\star[L]i_L}_{\beta_{L-1}}\Gamma^{[L]i_L}_{\alpha_{L-1}}\right)\,. && \text{(B.146)}
\end{aligned}$$

We compute this numerically as

$$\langle\tilde{\psi}|\psi\rangle = \sum_{\alpha,i,\beta}\tilde{\Gamma}^{\star[M]i}_{\alpha}K^{[M]}_{\alpha,\beta}\Gamma^{[M]i}_{\beta}\,, \tag{B.147}$$

where the object K is defined recursively as

$$K^{[l]}_{\alpha,\beta} = \sum_{\gamma,i_l}\tilde{\lambda}^{[l+1]}_{\alpha}\Gamma^{\star[l]i_l}_{\gamma,\alpha}\left[\sum_{\eta}K^{[l-1]}_{\gamma,\eta}\Gamma^{[l]i}_{\eta,\beta}\right]\lambda^{[l+1]}_{\beta} \tag{B.148}$$

subject to the initial condition

$$K^{[0]}_{\alpha,\beta} = \sum_{i}\tilde{\lambda}^{[2]}_{\beta}\tilde{\Gamma}^{\star[1]i}_{\beta}\Gamma^{[1]}_{\alpha}\lambda^{[2]}_{\alpha}\,. \tag{B.149}$$

The procedure `InnerProduct` discussed in Sec. B.9.9.13 performs this operation.

B.3.6.6 The Measure Derived Type

The measures amenable to TEBD can be broadly classified as:

1. Local measures : An observable whose associated operator acts only on a single site at a time, as in Sec. B.3.6.1. Such a measure is stored in a site indexed array.

2. Average measures: The expectations of a local operator at each site are summed and divided by the number of sites $\langle \hat{O} \rangle = \sum_{i=1}^{L} \langle \hat{O}_i \rangle / L$.
3. Correlations : An observable whose associated operator acts on two sites at a time and has no Fermi phases, as in Sec. B.3.6.2. Such a measure is stored in a `systemSize`×`systemSize` array.
4. Fermi Correlations : An observable whose associated operator acts on two sites at a time and has Fermi phases, as in Sec. B.3.6.3. Such a measure is stored in a `systemSize`×`systemSize` array.
5. Entanglement measures: These can further be classified into average entanglement measures such as the Meyer Q-measure, one-body (in the sense that they require only linear storage) measures such as the single-site von Neumann entropy of entanglement and the chain entropy (a link indexed list which gives the entanglement of the sites to the left of link with the sites to the right of link), and two-body measures such as the two-site von Neumann entropy of entanglement.

The `measures` derived type provides a compact means of computing and storing these measures based on their classification by the above scheme. The `measure` derived type is set up using the routine `AllocateMeasures` which has the syntax `AllocateMeasures(Measures,numLocal, numAvg, numCorr, numFermiCorr, numEnt)` The variable `Measures` is a `measure` derived type, `numLocal` is an INTEGER specifying the number of local measures desired, `numAvg` is an INTEGER specifying the number of average measures desired, `numCorr` is an INTEGER specifying the number of Correlations desired, `numFermiCorr` is an INTEGER specifying the number of Fermi Correlations, and `numEnt` specifies the entanglement measures desired. `numEnt`=0 returns the Q measure only, `numEnt=1` returns the Q-measure and the one-body entanglement measures, and `numEnt=2` returns all the entanglement measure listed above. Once `numEnt` has been chosen the entanglement measures are set and will be returned in:

Q-measure	`Measures%ent%qme`
one-body von Neumann entropy	`Measures%ent%vn`
chain entropy	`Measures%ent%chain`
two-body von Neumann entropy	`Measures%ent%tbvn`

For the other measures we must define the operators corresponding to each measure. For example, if we specified `numLocal=2` and we want the both the expectation of the single-site operators `Op1` and `Op2`, we define the first and second operators for local measures as

```
Measures%local(1)%op=Op1%mr
Measures%local(2)%op=Op2%mr
```

where `Op1%mr` represents a (possibly complex) `localSize`×`localSize` array. Similar logic follows for `Measures%avg(:)%op` which holds the average measure operators, `Measures%corr(:)%op` which holds the Correlation operators, and `Measures%FermiCorr(:)%op` which holds the Fermi Correlation operators. The two-body observable (e. g. Correlations and Fermi Correlations) operators are stored in two-site tensor product form. For example, if we want to calculate $\langle \mathrm{Op}_1 \mathrm{Op}_2 \rangle$ where Op_1 acts on site i and Op_2 on site j, we would define e. g.

```
Measures%corr(1)%op=TensorProd(Op1%mr,Op2%mr)
```

Once the operators have been defined, we calculate the measures as

```
CALL EvaluateMeasures(Measures, Gammas, Lambdas, H)
```

The result corresponding to a given operator is stored in a derived type component `%value`. For example:

`Measures%local(1)%value`

`Measures%local(2)%value`

`Measures%corr(1)%value`

return a site-indexed list of the expectation values of `Op1`, a site-indexed list of the expectation values of `Op2`, and a `systemSize`×`systemSize` array of the expectation values of $\mathrm{Op}_1 \otimes \mathrm{Op}_2$, respectively.

To deallocate the `measure` derived type `Measures`, use the routine `DeallocateMeasures` which has the syntax

`DeallocateMeasures(Measures)`

For more information, consult the procedure listing for `EvaluateMeasures` in Sec. B.9.9.35.

B.3.7 Supported Hamiltonians

In this section we list the Hamiltonians which can be generated by routines in the Open Source TEBD package. User-defined routines should be straightforward to construct using these examples.

B.3.7.1 The Heisenberg Spin Chain

The Heisenberg model is common model for studying phase transitions in quantum magnetic systems, defined by

$$\hat{H} = \sum_{i=1}^{L-1} \left[-J_x \hat{S}_i^x \hat{S}_{i+1}^x - J_y \hat{S}_i^y \hat{S}_{i+1}^y - J_z \hat{S}_i^z \hat{S}_{i+1}^z + h \hat{S}_i^z \right] \tag{B.150}$$

where $\hat{S}_i^\nu$ is the spin operator in the ν^{th} cartesian direction on the i^{th} site and h is an applied magnetic field. This Hamiltonian is supported mainly for pedagogical purposes. The 1D spin-1/2 Heisenberg spin chain can be solved exactly via the Bethe Ansatz (it was, in fact, the first application by Bethe of his Ansatz!) and some exact results are known regarding time evolution. See Sec. B.4.4 for a case study regarding this Hamiltonian.

The Heisenberg spin chain is generated using the procedure `HamiltonianHeisenberg` discussed in Sec. B.9.5.4 and the suite of operators needed to define and characterize it are generated using the procedure `CreateHeisenbergOps` discussed in Sec. B.9.5.2.

B.3.7.2 The Bose-Hubbard Hamiltonian

The Bose-Hubbard Hamiltonian was first introduced by Fisher et. al. as a model to study the phases of bosons with short-ranged repulsive interactions [17]. It became the subject of much interest in the atomic physics community after a paper by Jaksch et. al. demonstrated that an optical lattice containing ultracold bosons is an almost perfect realization of the Bose-Hubbard Hamiltonian [1]. The Hamiltonian is

$$\hat{H}_{\mathrm{BH}} = -J\sum_{i=1}^{L-1}\hat{t}_i + \frac{U}{2}\sum_{i=1}^{L}\hat{n}_i\left(\hat{n}-\hat{1}\right) + V\sum_{\langle i,i'\rangle}\hat{n}_i\hat{n}_{i'} + \sum_{i=1}^{L}\left(\epsilon_i-\mu\right)\hat{n}_i \tag{B.151}$$

where $\hat{a}_i$ destroys a particle on site i, a dagger denotes Hermitian conjugation, $\hat{n}_i \equiv \hat{a}_i^{\dagger}\hat{a}_i$ is the number operator on the i^{th} site, $\hat{t}_i \equiv \left(\hat{a}_{i+1}^{\dagger}\hat{a}_i + \mathrm{h.c.}\right)$ is the tunneling operator, and the notation $\langle \ldots \rangle$ means that the sum is taken over nearest neighbors. J is the energetic cost for a particle to "hop" or "tunnel" from a particular discrete position to its neighboring discrete position, U is the energetic cost for two bosons to occupy the same discrete position, V is the energetic cost for two bosons to occupy neighboring sites, μ is the chemical potential, and ϵ_i is the on-site energy of site i.

The Bose-Hubbard Hamiltonian is generated using the procedure `HamiltonianBoseHubbard` discussed in Sec. B.9.3.4 and the suite of operators needed to define and characterize it are generated using the procedure `CreateFieldOps` discussed in Sec. B.9.3.2.

B.3.7.3 The Hubbard Hamiltonian

The Hubbard hamiltonian is the simplest model Hamiltonian for studying the metal-insulator transition in an electronic system. It is given by

$$\hat{H} = -t \sum_{\langle i,j\rangle,\sigma} \left(\hat{f}^{\dagger}_{i,\sigma}\hat{f}_{j,\sigma} + \hat{f}^{\dagger}_{j,\sigma}\hat{f}_{i,\sigma}\right) + U\sum_{i=1}^{L}\left(\hat{n}_{i\uparrow} - \frac{1}{2}\right)\left(\hat{n}_{i\downarrow} - \frac{1}{2}\right) + V\sum_{\langle i,j\rangle,\sigma} \hat{n}_{i\sigma}\hat{n}_{j,\sigma} + \sum_{i\sigma}\left(\epsilon_i - \mu\right)\hat{n}_{i\sigma} \tag{B.152}$$

where $\hat{f}_{i,\sigma}$ destroys a fermion of spin $\sigma = \{\uparrow,\downarrow\}$ on the i^{th} lattice site, t is the energetic cost for a particle to “hop” or “tunnel” from a particular discrete position to its neighboring discrete position, U is the energetic cost for a spin up and a spin down particle to occupy the same site, and V is the energetic cost for two particles of the same spin to occupy neighboring sites.

The Hubbard Hamiltonian is generated using the procedure `HamiltonianHubbard` discussed in Sec. B.9.4.4 and the suite of operators needed to define and characterize it are generated using the procedure `CreateFermiSOps` discussed in Sec. B.9.4.2.

B.3.7.4 Spin-s Bose-Hubbard Hamiltonian

The low-energy Hamiltonian describing bosonic (pseudo)spin-s atoms in an optical trap interacting by a rotationally invariant, spin-preserving contact potential is, in second quantization, [18]

$$\hat{H} = -J\sum_{i=1}^{L-1}\sum_{\alpha=-s}^{s}\left(\hat{a}^{\dagger}_{i+1,\alpha}\hat{a}_{i,\alpha} + \text{h.c.}\right) + \sum_{S=0}^{2s} g_S \sum_{\alpha=-s}^{s} \hat{O}^{\dagger}_{i,S,\alpha}\hat{O}_{i,S,\alpha} + \sum_{i=1}^{L}\sum_{\alpha=-s}^{s}\epsilon_{i,\alpha}\hat{n}_{i,\alpha} \tag{B.153}$$

$$\hat{O}_{i,S,\alpha} = \sum_{\alpha_1,\alpha_2}\langle S\alpha|s,\alpha_1;s,\alpha_2\rangle\hat{a}_{i,\alpha_1}\hat{a}_{i,\alpha_2}\,,\ g_S = \frac{4\pi\hbar^2 a_S}{m}\,, \tag{B.154}$$

$$\hat{n}_{i,\alpha} = \hat{a}^{\dagger}_{i,\alpha}\hat{a}_{i,\alpha}\,, \tag{B.155}$$

where m is the mass of the atomic species, a_S is the s-wave scattering length in the total spin S channel, $\langle S\alpha|s,\alpha_1;s,\alpha_2\rangle$ is a Clebsch-Gordan coefficient (Sec. B.3.8.1), $\epsilon_{i,\alpha}$ is an external potential applied to the α^{th} spin component on site i, and $\hat{a}_{i,\alpha}$ destroys a particle in the α^{th} spin component on site i. For spin-1, this can be rearranged to

$$\begin{aligned}\hat{H} &= -J\sum_{i=1}^{L-1}\sum_{\alpha=-1}^{1}\left(\hat{a}^{\dagger}_{i+1,\alpha}\hat{a}_{i,\alpha}+\text{h.c.}\right)+\frac{U_o}{2}\sum_{i=1}^{L}\hat{n}_i\left(\hat{n}-\hat{1}\right)\\ &+\frac{U_2}{2}\sum_{i=1}^{L}\left(\hat{\mathbf{S}}_i^2-2\hat{n}_i\right)+\sum_{i=1}^{L}\sum_{\alpha=-1}^{1}\epsilon_{i,\alpha}\hat{n}_{i,\alpha}\,, \qquad \text{(B.156)}\\ U_0 &= \frac{g_0+2g_2}{3}\,,\ U_2=\frac{g_2-g_0}{3}\,, \qquad \text{(B.157)}\\ \hat{n}_i &= \sum_{\alpha=-1}^{1}\hat{n}_{i,\alpha}\,,\ \hat{\mathbf{S}}_i^2=\left(\sum_{\nu}\hat{S}^2_{i,\nu}\right)\,, \qquad \text{(B.158)}\end{aligned}$$

which has a clearer relationship to the ordinary Bose-Hubbard Hamiltonian, Eq. (B.151). We can define the spin operators $\left\{\hat{S}_{i,x},\hat{S}_{i,y},\hat{S}_{i,z}\right\}$ in terms of the creation and destruction operators as

$$\hat{S}_{i,\nu}=\sum_{\alpha,\beta=-s}^{s}\hat{a}^{\dagger}_{i,\alpha}F^{\nu}_{\alpha\beta}\hat{a}_{i,\beta}\,, \qquad \text{(B.159)}$$

where F^{ν} is the $(2s+1)\times(2s+1)$ dimensional matrix representation of the Pauli matrix in the ν^{th} cartesian direction. It may be easily verified that the operators defined as such obey the proper SU(2) commutation rules using the commutation properties of the bosonic operators and the Pauli matrices.

When we introduce a magnetic field **B**, the linear Zeeman effect causes a uniform rotation of the spins about **B**, and so we can neglect this term by transforming to the frame that rotates with the Larmor frequency in spin space. The quadratic Zeeman effect has the interaction term

$$\hat{H}_B = \frac{\mu_B^2}{4E_{\mathrm{hf}}} \int d\mathbf{r} \sum_{m,m'} \hat{\psi}_m^{\dagger}(\mathbf{r}) \left[(\mathbf{B}\cdot\mathbf{F})^2\right]_{mm'} \hat{\psi}_{m'}(\mathbf{r}) \,, \tag{B.160}$$

where μ_B is the magnetic dipole moment of the species in question and E_{hf} is the energetic splitting between the s and $s+1$ hyperfine levels. Assuming $\hat{e}_z$ polarized field and the Wannier basis decomposition $\hat{\psi}_\alpha = \sum_i w_\alpha(\mathbf{r}-\mathbf{r}_i)\,\hat{a}_{i,\alpha}$ we have

$$\begin{aligned} \hat{H}_B &= V_B \sum_{i=1}^{L-1} \sum_{\alpha,\alpha'} \left[\hat{S}_z^2\right]_{\alpha,\alpha'} \hat{a}_{i,\alpha}^{\dagger} \hat{a}_{i,\alpha'} \,, & \text{(B.161)} \\ &\equiv V_B \hat{V}_B \,. & \text{(B.162)} \end{aligned}$$

The spin-s Bose-Hubbard Hamiltonian is generated using the procedure `HamiltonianSpinS` discussed in Sec. B.9.6.5 and the suite of operators needed to define and characterize it are generated using the procedure `CreateSpinSops` discussed in Sec. B.9.6.2. If one wishes to call the spin-1 Hamiltonian with the Hubbard parameters instead of the scaled scattering lengths, Eq. (B.156), the procedure `HamiltonianSpinOne` discussed in Sec. B.9.6.4 will do so.

B.3.7.5 The Molecular Hubbard Hamiltonian

The Molecular Hubbard Hamiltonian (MHH), which describes the essential many body physics of closed-shell ultracold heteronuclear molecules in their absolute ground state in a quasi-one-dimensional optical lattice, is

$$\begin{aligned} \hat{H} = &-\sum_{JJ'M} t_{JJ'M} \sum_{\langle i,i'\rangle} \left(\hat{a}_{i',J'M}^{\dagger} \hat{a}_{iJM} + \mathrm{h.c.}\right) \\ &+ \sum_{JM} E_{JM} \sum_i \hat{n}_{iJM} - \pi \sin(\omega t) \sum_{JM} \Omega_{JM} \sum_i \left(\hat{a}_{iJ,M}^{\dagger} \hat{a}_{iJ+1,M} + \mathrm{h.c.}\right) \\ &+ \frac{1}{2} \sum_{\substack{J_1,J_1',J_2,J_2' \\ M,M'}} U_{dd}^{\substack{J_1,J_1',J_2,J_2' \\ M,M'}} \sum_{\langle i,i'\rangle} \hat{a}_{iJ_1M}^{\dagger} \hat{a}_{iJ_1'M} \hat{a}_{i'J_2M'}^{\dagger} \hat{a}_{i'J_2'M'} \,. \end{aligned} \tag{B.163}$$

A careful derivation of Eq. (B.163) and some results of its time evolution are presented in this paper.

B.3.8 Internal Degrees of Freedom

In this section we demonstrate how Hilbert spaces with internal degrees of freedom are generated by the code, and also discuss how vector coupling constants which are used by the spin-S and molecular Hubbard Hamiltonians are generated numerically.

B.3.8.1 Definitions and Basic properties of Vector-Coupling Coefficients

In the quantum theory of angular momentum, the total angular momentum operator $\hat{\mathbf{J}}$ of a two part system comprised of angular momenta $\hat{\mathbf{J}}_1$ and $\hat{\mathbf{J}}_2$ is given by the vector sum $\hat{\mathbf{J}} = \hat{\mathbf{J}}_1 + \hat{\mathbf{J}}_2$. There are two bases with which we can describe the rotational properties of this system. The first, known as the *uncoupled representation*, is the basis $|j_1 m_1 j_2 m_2\rangle$ which diagonalizes $\hat{\mathbf{J}}_1^2$, $\hat{\mathbf{J}}_2^2$, $\hat{J}_{1z}$, and $\hat{J}_{2z}$, with the fundamental equations

$$\hat{\mathbf{J}}_1^2 |j_1 m_1 j_2 m_2\rangle = j_1 (j_1 + 1) |j_1 m_1 j_2 m_2\rangle \,, \tag{B.164}$$
$$\hat{J}_{1z} |j_1 m_1 j_2 m_2\rangle = m_1 |j_1 m_1 j_2 m_2\rangle \,, \tag{B.165}$$
$$\hat{\mathbf{J}}_2^2 |j_1 m_1 j_2 m_2\rangle = j_2 (j_1 + 1) |j_1 m_1 j_2 m_2\rangle \,, \tag{B.166}$$
$$\hat{J}_{2z} |j_1 m_1 j_2 m_2\rangle = m_2 |j_1 m_1 j_2 m_2\rangle \,. \tag{B.167}$$

This basis describes the system in terms of the rotational properties of its constituents, but obscures the rotational properties of the composite system. The second basis, known as the *coupled representation*, is the basis $|j_1 j_2 j m\rangle$ which diagonalizes $\hat{\mathbf{J}}_1^2$, $\hat{\mathbf{J}}_2^2$, $\hat{\mathbf{J}}^2$, and $\hat{J}_z = \hat{J}_{1z} + \hat{J}_{2z}$, with the fundamental equations

$$\hat{\mathbf{J}}_1^2 |j_1 j_2 j m\rangle = j_1 (j_1 + 1) |j_1 j_2 j m\rangle \,, \tag{B.168}$$
$$\hat{\mathbf{J}}_2^2 |j_1 j_2 j m\rangle = j_2 (j_1 + 1) |j_1 j_2 j m\rangle \,, \tag{B.169}$$
$$\hat{\mathbf{J}}^2 |j_1 j_2 j m\rangle = j (j + 1) |j_1 j_2 j m\rangle \,, \tag{B.170}$$
$$\hat{J}_z |j_1 j_2 j m\rangle = m |j_1 j_2 j m\rangle \,. \tag{B.171}$$

This basis describes the system in terms of its composite rotational properties, and is often denoted $|jm\rangle$ for brevity. The two bases are equivalent, and so can be related via the unitary transformation

$$|j_1m_1j_2m_2\rangle = \sum_{jm}\langle jm|j_1m_1j_2m_2\rangle|jm\rangle \tag{B.172}$$

or the inverse transformation

$$|jm\rangle = \sum_{m_1m_2}\langle j_1m_1j_2m_2|jm\rangle|j_1m_2j_2m_1\rangle \tag{B.173}$$

where the transformation coefficients $\langle jm|j_1m_1j_2m_2\rangle = \langle j_1m_1j_2m_2|jm\rangle$, known as the *Clebsch-Gordan coefficients* (CG coefficients), are chosen to be real. It is expedient to define three functions:

$$T\left(j_1,j_2,j\right) = \left\{\begin{array}{c} 1 \text{ if } \left|j_1-j_2\right| \leq j \leq j_1+j_2 \text{ and } j_1+j_2+j \text{ is an integer} \\ 0 \text{ otherwise}\end{array}\right., \tag{B.174}$$

$$P\left(m,j\right) = \left\{\begin{array}{c} 1 \text{ if } -j \leq m \leq j \text{ and } m+j \text{ is an integer} \\ 0 \text{ otherwise}\end{array}\right., \tag{B.175}$$

$$Q\left(m_1,m_2,m\right) = \left\{\begin{array}{c} 1 \text{ if } m_1+m_2=m \\ 0 \text{ otherwise}\end{array}\right., \tag{B.176}$$

which encapsulate the selection rules required of the arguments of vector coupling coefficients. For a nonzero CG coefficient $\langle j_1m_1j_2m_2|jm\rangle$, we must have $T\left(j_1,j_2,j\right) = P\left(m_1,j_1\right) = P\left(m_2,j_2\right) = P\left(m,j\right) = Q\left(m_1,m_2,m\right) = 1$. The CG coefficients are generated by the function `Clebsch` discussed in Sec. B.9.2.15.

Wigner defined an equivalent, but more symmetric, object known as a *Wigner 3-j symbol* as

$$\left(\begin{array}{ccc} j_1 & j_2 & j \\ m_1 & m_2 & -m\end{array}\right) \equiv (-1)^{j_1-j_2+m}\frac{\langle j_1m_1j_2m_2|jm\rangle}{\sqrt{2j+1}}\,. \tag{B.177}$$

The 3-j symbol is left unchanged by an even permutation of its columns, and is multiplied by the factor $(-1)^{j_1+j_2+j}$ under an odd permutation of its columns. These properties allow for easier analytic manipulation, and so 3-j coefficients are favored over CG coefficients in many applications.

The orthonormality of the basis elements $|jm\rangle$ and $|j_1 m_1 j_2 m_2\rangle$ lead to the Clebsch-Gordan orthogonality relations

$$\sum_{j,m} \langle j_1 m_1 j_2 m_2 | jm\rangle \langle jm | j_1 m_1' j_2 m_2'\rangle = \delta_{m_1,m_1'} \delta_{m_2,m_2'} \,, \tag{B.178}$$

$$\sum_{m_1,m_2} \langle jm | j_1 m_2 j_2 m_2\rangle \langle j_1 m_1 j_2 m_2 | j' m'\rangle = \delta_{j,j'} \delta_{m,m'} \,, \tag{B.179}$$

subject, of course, to the selection rules above. These, in turn, imply the 3-j orthogonality relations

$$\sum_{j,m} (2j+1) \begin{pmatrix} j_1 & j_2 & j \\ m_1 & m_2 & m \end{pmatrix} \begin{pmatrix} j_1 & j_2 & j \\ m_1' & m_2' & m \end{pmatrix} = \delta_{m_1,m_1'} \delta_{m_2,m_2'} \,, \tag{B.180}$$

$$\sum_{m_1,m_2} (2j+1) \begin{pmatrix} j_1 & j_2 & j \\ m_1 & m_2 & m \end{pmatrix} \begin{pmatrix} j_1 & j_2 & j' \\ m_1 & m_2 & m' \end{pmatrix} = \delta_{j,j'} \delta_{m,m'} \,. \tag{B.181}$$

Racah, in an important 1942 paper, gave the explicit 3-j formula [19]

$$\begin{aligned} \begin{pmatrix} a & b & c \\ \alpha & \beta & \gamma \end{pmatrix} &= \triangle\,(abc) \sqrt{(a+\alpha)!\,(a-\alpha)!\,(b+\beta)!\,(b-\beta)!\,(c+\gamma)!\,(c-\gamma)!} \\ &\times \sum_t (-1)^{a-b-\gamma+t} \left[t!\,(c-b+t+\alpha)!\,(c-a+t-\beta)!\,(a+b-c-t)!\right]^{-1} \\ &\times \left[(a-t-\alpha)!\,(b-t+\beta)!\right]^{-1} \,, \end{aligned} \tag{B.182}$$

where we have defined the "triangle coefficient" as

$$\triangle\,(abc) \equiv \sqrt{\frac{(a+b-c)!\,(b+c-a)!\,(c+a-b)!}{(a+b+c+1)!}} \tag{B.183}$$

and the summation includes all values of t such that the factorials are nonnegative. If all of the selection rules are satisfied, then all of the factorials will be integer for integer t. If these are not satisfied, then the coefficient is identically zero. To explicitly find the integer values of t to include in the summation, we investigate when the summed factorials become negative. The first term ($t!$) ensures that the lowest allowed value of t is zero. The next two terms involve addition of t to some constants, thus they may set a lower bound greater than 0. From inspection of the arguments, we see that the lowest t value allowed by these three terms is $\min\{0, c-b+\alpha, c-a-\beta\}$. The remaining terms provide an upper bound for t. By inspection this is $\min\{a+b-c, a-\alpha, b+\beta\}$, which is never negative so long as the triangle inequalities hold. The number of terms in the sum is $\nu+1$ where ν is the smallest of the nine numbers [20]

$$\begin{array}{ccc} a \pm \alpha & b \pm \beta & c \pm \gamma \\ a+b-c & b+c-a & c+a-b \end{array} .$$

Direct use of formula Eq. (B.182) above for numerical computation would result in large roundoff error and possible integer overflow. Thus, we work not with the factorials but with their logarithms, which grow much more slowly. See the listing for `ThreeJ` in Sec. B.9.2.14 for implementation details.

In the coupling of three angular momenta j_1, j_2, and j_3 to form a total angular momentum j, we can first couple j_1 to j_2 as

$$|j_{12}m_{12}\rangle = \sum_{m_1,m_2} |j_1m_1j_2m_2\rangle\langle j_1m_1j_2m_2|j_{12}m_{12}\rangle \tag{B.184}$$

and then couple the resultant angular momentum j_{12} to j_3 to result in total angular momentum j as

$$|\left(\left(j_1j_2j_{12}\right)j_3\right)jm\rangle = \sum_{m_{12},m_3} |j_{12}m_{12}j_3m_3\rangle\langle j_{12}m_{12}j_3m_3|jm\rangle \tag{B.185}$$

where the parentheses in the resultant ket reminds us of the order of coupling. Alternatively, we can first couple j_2 to j_3 as

$$|j_{23}m_{23}\rangle = \sum_{m_2,m_3} |j_2m_2j_3m_3\rangle\langle j_2m_2j_3m_3|j_{23}m_{23}\rangle \tag{B.186}$$

and then couple the resultant angular momentum j_{23} to j_1 to result in total angular momentum j as

$$|\left(j_1\left(j_2j_3j_{23}\right)\right)jm\rangle = \sum_{m_{23},m_1} |j_1m_1j_{23}m_{23}\rangle\langle j_1m_2j_{23}m_{23}|jm\rangle\,. \tag{B.187}$$

Both coupling schemes result in complete orthonormal bases for the coupled representation of the three angular momenta, and so can be related through a unitary transformation

$$|\left(\left(j_1j_2j_{12}\right)j_3\right)jm\rangle = \sum_{j_{23}} |\left(j_1\left(j_2j_3j_{23}\right)\right)jm\rangle\langle\left(j_1\left(j_2j_3j_{23}\right)\right)jm|\left(\left(j_1j_2j_{12}\right)j_3\right)jm\rangle\,. \tag{B.188}$$

The coefficients, which are independent of m, are cast in their most symmetric form as a *Wigner 6-j symbol* defined by

$$\left\{\begin{array}{ccc} j_1 & j_2 & j_{12} \\ j_3 & j & j_{23}\end{array}\right\} \equiv (-1)^{j_1+j_2+j_3+j}\left[\left(2j_{12}+1\right)\left(2j_{23}+1\right)\right]^{-\frac{1}{2}}\langle\left(\left(j_1j_2j_{12}\right)j_3\right)j|\left(j_1\left(j_2j_3j_{23}\right)\right)j\rangle\,. \tag{B.189}$$

For a nonzero 6-j $\left\{\begin{array}{ccc} j_1 & j_2 & j_3 \\ J_1 & J_2 & J_3\end{array}\right\}$ we must have $T\left(j_1,j_2,j_3\right) = T\left(j_1,J_2,J_3\right) = T\left(J_1,j_2,J_3\right) = T\left(J_1,J_2,j_3\right) = 1$. Because the transformation between bases is unitary, we have the orthonormality condition

$$\sum_j \left(2j+1\right)\left(2j''+1\right)\left\{\begin{array}{ccc} j_1 & j_2 & j' \\ j_3 & j_4 & j\end{array}\right\}\left\{\begin{array}{ccc} j_3 & j_2 & j \\ j_1 & j_4 & j''\end{array}\right\} = \delta_{j',j''}\,. \tag{B.190}$$

To compute the Wigner 6-j coefficient we use the Racah formula [19]

$$\left\{ \begin{array}{ccc} j_1 & j_2 & j_3 \\ J_1 & J_2 & J_3 \end{array} \right\} = \triangle\left(j_1 j_2 j_3\right) \triangle\left(j_1 J_2 J_3\right) \triangle\left(J_1 j_2 J_3\right) \triangle\left(J_1 J_2 j_3\right) \sum_t \left(-1\right)^t \left(t+1\right)!$$
$$\times \left[\left(j_1+j_2+J_1+J_2-t\right)! \left(j_2+j_3+J_2+J_3-t\right)! \left(j_3+j_1+J_3+J_1-t\right)!\right]^{-1}$$
$$\times \left[\left(t-j_1-j_2-j_3\right)! \left(t-j_1-J_2-J_3\right)! \left(t-J_1-j_2-J_3\right)! \left(t-J_1-J_2-j_3\right)!\right]^{-1} , \tag{B.191}$$

where $\triangle\left(j_1 j_2 j_3\right)$ was defined in Eq. (B.183) and the summation includes all t such that the factorials are nonnegative. This gives (nonnegative) lower and upper bounds of

$$t_{\text{min}} = \max\left\{j_1+j_2+j_3, j_1+J_2+J_3, J_1+j_2+J_3, J_1+J_2+j_3\right\} , \tag{B.192}$$
$$t_{\text{max}} = \min\left\{j_1+j_2+J_1+J_2, j_2+j_3+J_2+J_3, j_3+j_1+J_3+J_1\right\} , \tag{B.193}$$

respectively. The number of terms in the summation is $1+\sigma$ where σ is the smallest of the twelve numbers [20]

$$\begin{array}{cccc} j_1+j_2-j_3 & j_1+J_2-J_3 & J_1+j_2-J_3 & J_1+J_2-j_3 \\ j_2+j_3-j_1 & J_2+J_3-j_1 & j_2+J_3-J_1 & J_2+j_3-J_1 \\ j_3+j_1-j_2 & J_3+j_1-J_2 & J_3+J_1-j_2 & j_3+J_1-J_2 \end{array} .$$

See the listing for `SixJ` in Sec. B.9.2.16 for implementation details.

The last vector coupling coefficient we consider is the *Wigner 9-j symbol*, which arises in the re-coupling of four angular momenta. It is defined by

$$\left\{ \begin{array}{ccc} j_1 & j_2 & j_{12} \\ j_3 & j_4 & j_{34} \\ j_{12} & j_{24} & j \end{array} \right\} \equiv \frac{\left\langle \left(\left(j_1 j_2 j_{12}\right)\left(j_3 j_4 j_{34}\right)\right) j \right| \left(\left(j_1 j_3 j_{13}\right)\left(j_2 j_4 j_{24}\right)\right) j \rangle}{\left[\left(2j_{12}+1\right)\left(2j_{34}+1\right)\left(2j_{13}+1\right)\left(2j_{24}+1\right)\right]^{\frac{1}{2}}} . \tag{B.194}$$

From the orthonormality of the coupled bases, we have the orthogonality relation

$$\sum_{j_{12} j_{34}} \left(2j_{12}+1\right)\left(2j_{34}+1\right)\left(2j_{13}+1\right)\left(2j_{24}+1\right) \left\{ \begin{array}{ccc} j_1 & j_2 & j_{12} \\ j_3 & j_4 & j_{34} \\ j_{13} & j_{24} & j \end{array} \right\} \left\{ \begin{array}{ccc} j_1 & j_2 & j_{12} \\ j_3 & j_4 & j_{34} \\ j'_{13} & j'_{24} & j \end{array} \right\}$$
$$= \delta_{j_{13},j'_{13}} \delta_{j_{24},j'_{24}} . \tag{B.195}$$

The 9-j is also used to find the reduced matrix elements of the tensor product of two operators in the coupled representation, e.g.

$$\begin{aligned}&\langle\gamma' j_1' j_2' j'|| \left[T^{(k_1)}\otimes U^{(k_2)}\right]^{(k)} ||\gamma j_1 j_2 j\rangle\\&=\sqrt{(2j+1)(2j'+1)(2k+1)}\sum_{\gamma''}\left\{\begin{array}{ccc} j_1' & j_1 & k_1\\ j_2' & j_k & k_2\\ j' & j & k\end{array}\right\}\langle\gamma' j_1'||T^{(k_1)}||\gamma'' j_1\rangle\langle\gamma'' j_2'||U^{(k_2)}||\gamma j_2\rangle\end{aligned} \tag{B.196}$$

where $T^{(k_1)}$ and $U^{(k_2)}$ are tensor operators that act in the j_1 and j_2 subspaces, respectively, and the double vertical bar signifies a reduced matrix element. This formula has been used in SU(2) conserving implementations of DMRG[21], and so may prove useful in future versions of the code.

There is no single-index Racah formula for the Wigner 9-j symbol, but we can compute them from the contraction formula

$$\begin{aligned}\left\{\begin{array}{ccc} j_{11} & j_{12} & j_{13}\\ j_{21} & j_{22} & j_{23}\\ j_{31} & j_{32} & j_{33}\end{array}\right\} &= \sum_t (-1)^t (t+1)\left\{\begin{array}{ccc} j_{11} & j_{21} & j_{31}\\ j_{32} & j_{33} & \frac{t}{2}\end{array}\right\}\\ &\times\left\{\begin{array}{ccc} j_{12} & j_{22} & j_{32}\\ j_{21} & \frac{t}{2} & j_{23}\end{array}\right\}\left\{\begin{array}{ccc} j_{13} & j_{23} & j_{33}\\ \frac{t}{2} & j_{11} & j_{12}\end{array}\right\}.\end{aligned} \tag{B.197}$$

The summation runs over all t such that the 6-js are not trivially zero (i.e. all triangle rules hold). This amounts to the lower and upper bounds $\max\left\{\left|j_{11}-j_{33}\right|,\left|j_{32}-j_{21}\right|,\left|j_{23}-j_{12}\right|\right\}$ and $\min\left\{j_{11}+j_{33},j_{32}+j_{21},j_{23}+j_{12}\right\}$, and also the condition that t increases in increments of two. See the listing for `NineJ` in Sec. B.9.2.17 for implementation details.

B.3.8.2 Fock Space Combinatorics with Spin Degrees of Freedom

Consider a site on a lattice system which can be occupied by at most N bosons, each of which has spin s. We wish to find the local dimension d, i.e. the total number of allowed quantum states per site. If we first consider how many states exist for a fixed number of particles k, the problem becomes the familiar combinatoric problem of how many ways one can place k identical particles into $2s+1$ wells. The answer is simply the number of combinations of k objects and $2s$ partitions, $\begin{pmatrix} k+2s \\ 2s \end{pmatrix}$. If we now sum over all k from 0 to N, we have

$$d=\sum_{k=0}^{N}\begin{pmatrix} k+2s \\ 2s \end{pmatrix}=\begin{pmatrix} N+2s+1 \\ 2s+1 \end{pmatrix} \tag{B.198}$$

from the upper summation identity of binomial coefficients. If we denote the number in the m^{th} spin component as N_m, then we index the Fock space for a fixed total number N as in Table B.3.

$$\begin{array}{cccccc}
N & 0 & 0 & 0 & \dots & 0 \\
N-1 & 1 & 0 & 0 & \dots & 0 \\
N-1 & 0 & 1 & 0 & \dots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
N-1 & 0 & 0 & 0 & \dots & 1 \\
N-2 & 2 & 0 & 0 & \dots & 0 \\
N-2 & 1 & 1 & 0 & \dots & 0 \\
N-2 & 1 & 0 & 1 & \dots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
N-2 & 1 & 0 & 0 & \dots & 1 \\
N-2 & 0 & 2 & 0 & \dots & 0 \\
N-2 & 0 & 1 & 1 & \dots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
N-2 & 0 & 1 & 0 & \dots & 1 \\
N-2 & 0 & 0 & 2 & \dots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
N-2 & 0 & 0 & 0 & \dots & 2 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots
\end{array}$$

Figure B.6: Structure of Fock space algorithm.

The indexing given suggests our method for constructing this Fock space for arbitrary s. We begin with all N particles in the $m=s$ state, of which there is one such arrangement. We then consider $N-1$ particles in the $m=s$ state, of which there are $\begin{pmatrix} 2s \\ 2s-1 \end{pmatrix}$ states- the number of ways to put the single particle into the $2s$ remaining spin components. We now consider $N-2$ particles in the $m=s$ state. There are $\begin{pmatrix} 2s+1 \\ 2s-1 \end{pmatrix}$ such states, which is also $\sum_{i=0}^{2s}\begin{pmatrix} 1+(2s-1)-i \\ (2s-1)-i \end{pmatrix}$, the number of ways of putting a single particle particle into $2s$ wells added to the number of ways of putting a single particle into $2s-1$ wells and so on. This implies that we can break the problem of putting 2 particles into $2s-1$ wells into $2s$ single well problems. Another diagram makes this clearer. Consider Figure B.6, where we demonstrate this substructure. The section colored

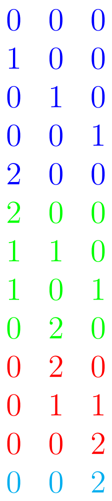
Table B.3: Fock space indexing for a fixed number of particles N.

label	N_s	N_{s-1}	N_{s-2}	N_{s-3}	$\ldots$	N_{-s+1}	N_{-s}
$\begin{pmatrix} 2s-1 \\ 2s-1 \end{pmatrix}$ =1	N	0	0	0	$\ldots$	0	0
2	N-1	1	0	0	$\ldots$	0	0
3	N-1	0	1	0	$\ldots$	0	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
2s	N-1	0	0	0	$\ldots$	1	0
$\begin{pmatrix} 2s-1 \\ 2s-1 \end{pmatrix} + \begin{pmatrix} 2s \\ 2s-1 \end{pmatrix} = 2s+1$	N-1	0	0	0	$\ldots$	0	1
2s+2	N-2	2	0	0	$\ldots$	0	0
2s+3	N-2	1	1	0	$\ldots$	0	0
2s+4	N-2	1	0	1	$\ldots$	0	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$1 + \sum_{i=0}^{2} \begin{pmatrix} i+2s-1 \\ 2s-1 \end{pmatrix}$	N-3	3	0	0	$\ldots$	0	0
$2 + \sum_{i=0}^{2} \begin{pmatrix} i+2s-1 \\ 2s-1 \end{pmatrix}$	N-3	2	1	0	$\ldots$	0	0
$3 + \sum_{i=0}^{2} \begin{pmatrix} i+2s-1 \\ 2s-1 \end{pmatrix}$	N-3	2	0	1	$\ldots$	0	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$1+\sum_{i=0}^{N} \begin{pmatrix} i+2s-1 \\ 2s-1 \end{pmatrix}$	0	N	0	0	$\ldots$	0	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$1+\sum_{i=0}^{N} \begin{pmatrix} i+2s-1 \\ 2s-1 \end{pmatrix} + \begin{pmatrix} 2s-1 \\ 2s-2 \end{pmatrix}$	0	N-1	1	0	$\ldots$	0	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\begin{pmatrix} N+2s \\ 2s \end{pmatrix}$	0	0	0	0	$\ldots$	0	N

in cyan is putting a single particle into $2s-1$ wells, the part colored in magenta is putting a single particle into $2s-2$ wells, and so on. If we now recognize that everything in green is the original problem of putting N particles into M wells with $N=2$ and $M=2s$, we see that the entire algorithm has this recursive substructure.

We take advantage of this recursive structure by making the procedure for generating the Fock space recursive. This algorithm focuses on the m^{th} spin component, which can have up to nmax particles. If the state is $\underbrace{\ldots}_{m-1}$,0,...,0 then it is simply counted, if the state is $\underbrace{\ldots}_{m-1}$,1,0,...,0 then the 1 is moved from its current position to the last spin component, counting each time, and if the state is $\underbrace{\ldots}_{m-1}$,$k$,0,...,0 with $k>1$ we loop over performing the algorithm on the subspace beginning with spin component $m+1$ and 0,1,...,nmax particles in spin component $m+1$. Looping the recursive algorithm over 0,1,...,`maxFilling` particles in the first spin component generates the entire Fock space.

As a concrete example, consider a spin-1 Fock space with `maxFilling`=2, and observe Figure B.7. Calling `onsiteStateListIdof` begins a loop (the blue loop) which initializes `onsiteStateListIdofInner` with 0 particles in the first spin component, focused on the first spin component. This state is $\langle N_1, N_0, N_{-1}\rangle = \langle 0,0,0\rangle$, which is of the form $\langle\underbrace{\ldots}_{m-1}, 0, \ldots, 0\rangle$ with $m=1$, and so it is simply counted, giving $\langle 0,0,0\rangle$ as state 1. The blue loop then initializes `onsiteStateListIdofInner` with 1 particle in the first spin component, focused on the first spin component. This state is $\langle 1,0,0\rangle$, which is of the form $\langle\underbrace{\ldots}_{m-1}, 1, \ldots, 0\rangle$ with $m=1$ and so the inner routine returns $\langle 1,0,0\rangle$ as state 2, $\langle 0,1,0\rangle$ as state 3, and $\langle 0,0,1\rangle$ as state 4. The blue loop now initializes `onsiteStateListIdofInner`



0	0	0
1	0	0
0	1	0
0	0	1
2	0	0
2	0	0
1	1	0
1	0	1
0	2	0
0	2	0
0	1	1
0	0	2
0	0	2

Figure B.7: Fock space algorithm execution for Spin-1.

with 2 particles in the first spin component, focused on the first spin component. This state is $\langle 2,0,0\rangle$, which is of the form $\langle \underbrace{...}_{m-1}, k, ..., 0\rangle$ with $k > 1$ and $m = 1$. `onsiteStateListIdofInner` now calls a loop (the green loop) which initializes `onsiteStateListIdofInner` with 2 particles in the first spin component and zero in the second spin component, focused on the second spin component. The state is $\langle 2,0,0\rangle$, which is of the form $\langle \underbrace{...}_{m-1}, 0, ..., 0\rangle$ with $m = 2$, and so the state is simply counted, giving $\langle 2,0,0\rangle$ as state 5. The green loop now moves one particle from the first spin component into the second spin component. The state is $\langle 1,1,0\rangle$, which is of the form $\langle \underbrace{...}_{m-1}, 1, ..., 0\rangle$ with $m = 2$, and so the inner routine returns $\langle 1,1,0\rangle$ as state 6 and $\langle 1,0,1\rangle$ as state 7. The green loop now initializes `onsiteStateListIdofInner` with 2 particles in the second spin component, focused on the second spin component. This state is $\langle 0,2,0\rangle$, which is of the form $\langle \underbrace{...}_{m-1}, k, ..., 0\rangle$ with $k > 1$ and $m = 2$. `onsiteStateListIdofInner` now calls a loop (the red loop) which initializes `onsiteStateListIdofInner` with no particles in the first spin component, two particles in the second spin component, and zero in the third spin component, focused on the third spin component. This state is $\langle 0,2,0\rangle$, which is of the form $\langle \underbrace{...}_{m-1}, 0, ..., 0\rangle$ with $m = 3$, and so it is simply counted, giving $\langle 0,2,0\rangle$ as state 8. The red loop now moves one particle from the second spin component into the third spin component. The state is $\langle 0,1,1\rangle$, which is of the form $\langle \underbrace{...}_{m-1}, 1, ..., 0\rangle$ with $m = 3$, and so the inner routine returns $\langle 0,1,1\rangle$ as state 9. The red loop now moves one particle from the second spin component into the third spin component. The state is $\langle 0,0,2\rangle$, which is of the form $\langle \underbrace{...}_{m-1}, k, ..., 0\rangle$ with $k > 1$ and $m = 3$. `onsiteStateListIdofInner` now calls a loop (the cyan loop) which initializes `onsiteStateListIdofInner` with no particles in either the first or second spin components and two particles in the third spin component, focused on a fictitious fourth spin component. This state is $\langle 0,0,2\rangle$, which is of the form $\langle \underbrace{...}_{m-1}, 0, ..., 0\rangle$ with $m = 4$, and so it is simply counted, giving $\langle 0,0,2\rangle$ as state 10. This completes the algorithm for $s=1$, `maxFilling`=2. The general procedure should now be clear. The recursive procedure which generates the Fock space for a fixed number of particles in the highest spin component is `onsiteStateListIdofInner`, listed in Sec. B.9.2.5, and the procedure which generates the entire Fock space by calling `onsiteIdofInner` with $N=0,1,\dots$,`maxFilling` particles is `onsiteStateListIdof`, listed in Sec. B.9.2.4.

B.4 Case Studies and Exercises

The purpose of this section is to provide exercises which will orient the user with Open Source TEBD, and also require them to modify the code for specific purposes.

B.4.1 Case Study 1:The Bose-Hubbard Hamiltonian–Ground state Properties

In this section we shall study the ground state properties of the Bose-Hubbard Hamiltonian (BHH) in the absence of a trap:

$$\hat{H}=-J\sum_{\langle i,i'\rangle}\left[\hat{a}_i^{\dagger}\hat{a}_{i'}+\text{h.c.}\right]+\frac{U}{2}\sum_{i=1}^{L}\hat{n}_i\left(\hat{n}-\hat{1}\right)-\sum_{i=1}^{L}\mu\hat{n}_i \tag{B.199}$$

using TEBD. We will begin with an overview of the properties of the Bose-Hubbard model.

First, let us consider the symmetries of the BHH. It is symmetric under the global U(1) phase transformation

$$\hat{a}_i\rightarrow\hat{a}_ie^{i\theta}\,, \tag{B.200}$$

the associated conserved quantity is the total number of bosons

$$\hat{N}=\sum_i\hat{a}_i^{\dagger}\hat{a}_i\,. \tag{B.201}$$

The first term in Eq. (B.199) couples neighboring sites in a way that prefers to break this global symmetry, and this competes with the second term, which is completely local and prefers states which are invariant under Eq. (B.200). Thus, we may reasonably expect a quantum phase transition as a function of J/U from a state in which the U(1) symmetry is unbroken to one in which it is broken. We begin our investigation of this phase transition by considering the limiting cases $J \to 0$ and $U \to 0$.

In the $J \to 0$ limit particles cannot hop from site to site, and so each site is occupied by n bosons, where n is a non-negative integer. The energy becomes a function of n alone:

$$\epsilon\left(n\right) = \frac{U}{2} n\left(n-1\right) - \mu n\,, \tag{B.202}$$

and so we can find the number of bosons per site by minimizing this expression with respect to n. We find that for chemical potentials in the range

$$n-1 \le \mu/U \le n\,,\ \ n \ge 1\,, \tag{B.203}$$

exactly n bosons occupy each site, and the total wavefunction is

$$|\Psi\rangle = \left[\prod_{i=1}^{L} \frac{1}{\sqrt{n!}}\left(\hat{a}_i^{\dagger}\right)^n\right]|00\dots0\rangle\ = |nn\dots n\rangle. \tag{B.204}$$

Investigating the density $\rho \equiv N/L$ (N is the total number of particles, not the number of particles per site) as a function of μ, we see a step-like structure where the chemical potential can be varied in the range according to Eq. (B.203) at integer fillings without changing ρ. Thus, the state represented by the wavefunction of Eq. (B.204) represents a gapped phase known as a *Mott Insulator* (MI).

Let us now consider the effects of small J. If we have a system in the MI phase with chemical potential in the range

$$\mu/U = n - \frac{1}{2} + \alpha\,, \quad -\frac{1}{2} < \alpha < \frac{1}{2} \tag{B.205}$$

then a tunneling energy J that is smaller than either $\left(\frac{1}{2} - \alpha\right) U$ or $\left(\frac{1}{2} + \alpha\right) U$ will not affect the system. We understand this, as the kinetic energy we gain from adding/removing a particle to/from the system is smaller than the potential energy cost of the particle interacting with the other particles in the lattice. Thus, for each integer density ρ we expect a finite area in the μ vs J phase diagram corresponding to the MI phase. The strong suppression of hopping also leads to a localization of density fluctuations, making the states incompressible and justifying the use of the term insulator.

In the limit $U \to 0$ the particles are noninteracting, and so the ground state is the state with all particles in the lowest energy single-particle state. For a system of N particles and periodic boundary conditions this is represented by the coherent state

$$|\Psi\rangle = \frac{1}{\sqrt{N!}} \left[\frac{1}{\sqrt{L!}} \sum_{i=1}^{L} \hat{a}_i^\dagger \right]^N |00\ldots0\rangle\,, \tag{B.206}$$

which represents the *Superfluid* (SF) phase. This state is highly delocalized and displays large number fluctuations, in contrast to the MI state which is highly localized and displays no number fluctuations. The complementary (in Bohr's sense) observable to number is phase, and so the SF phase displays a phase coherence which breaks the U(1) symmetry Eq. (B.200). We will investigate ways to quantify this phase coherence in the exercises.

We now show how to characterize the SF and MI phases using the case study `BoseHubbard_ITP` found in `Bose_Hubbard_Wrapper` via Figure B.8-Figure B.10.

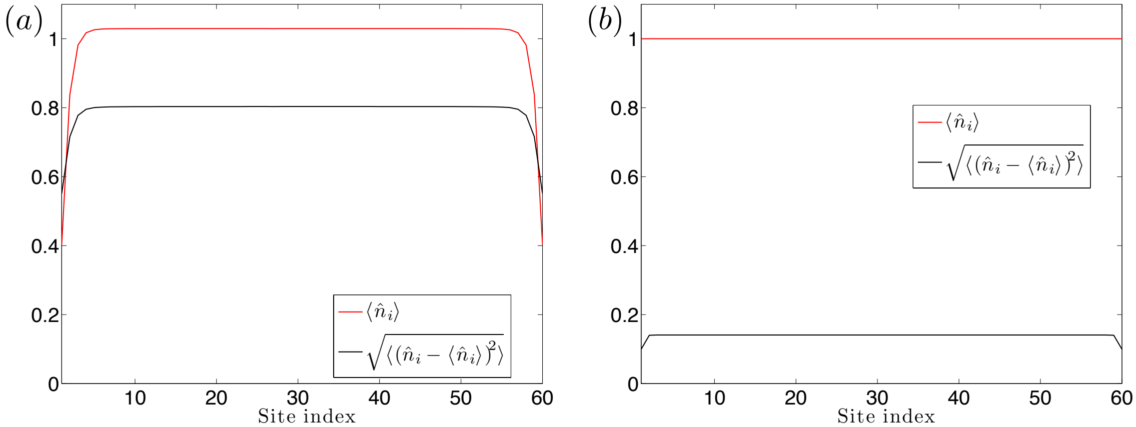


Figure B.8: Numbers and number deviations for the Bose-Hubbard model. (a) $\langle \hat{n} \rangle$ and deviations for a superfluid system. Note the very large fluctuations and the tapering towards the boundaries. (b)$\langle \hat{n} \rangle$ and deviations for a Mott insulating system. Note that the number fluctuations are much smaller than in the superfluid case.

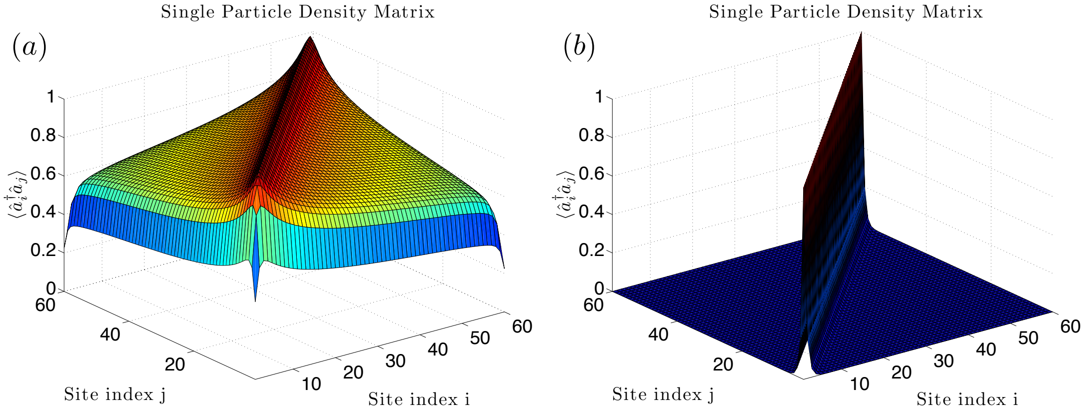


Figure B.9: Single-particle density matrix for the Bose-Hubbard model. (a) Single-particle density matrix for a superfluid system. The strong off-diagonal correlations are indicative of the quasi-off-diagonal long range order that characterizes the superfluid phase. (b) Single-particle density matrix for a Mott insulating system. The off-diagonal correlations decay very rapidly with the site separation, in stark contrast to the superfluid phase.

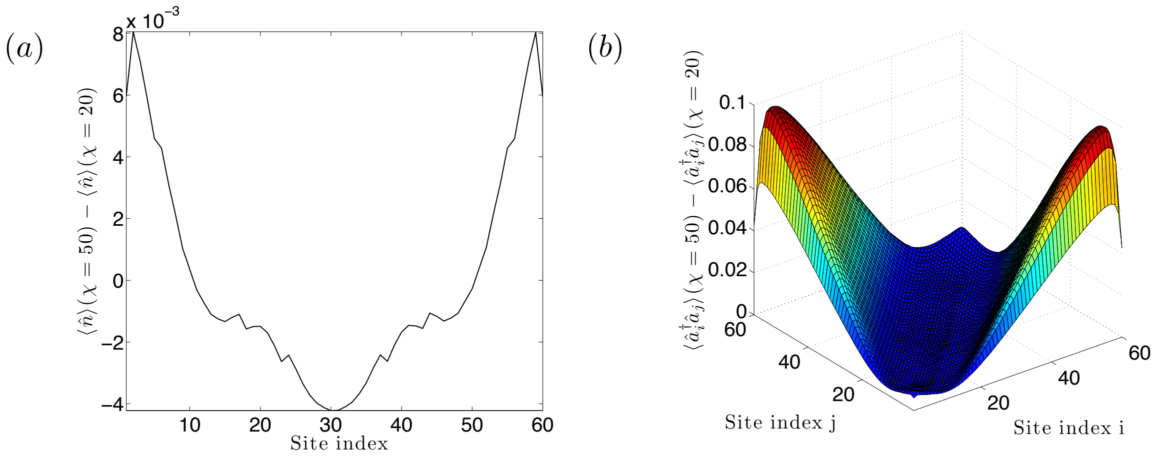


Figure B.10: Differences in number and single-particle density matrix as χ is increased. (a) Difference in the on-site expectation values of numbers as χ is increased. The superfluid case is shown, the Mott case is zero on the scale of this plot. The Mott phase numbers do not change as more entanglement is allowed, while the superfluid phase numbers change slightly as the amount of entanglement allowed increases. (b) Difference in the single-particle density matrix as χ is increased. The Mott phase differences are smaller than 1 part in a billion, but the superfluid phase differences are significant at large distances-up to one part in ten. The moral here is that long-range correlations need much higher χ to converge properly than do single-site observables e. g. number or energy. This is especially true in critical systems.

B.4.1.1 Exercise: ρ as a Function of μ

We can determine the universality class of the SF-MI transition in the following way. Consider the $U \to \infty$ limit of Eq. (B.199), known as the "hard core limit." Now each site can have either 0 or 1 bosons. If we now identify

$$\sigma_j^x = \hat{a}_j + \hat{a}_j^\dagger, \quad \sigma_j^y = -i\left(\hat{a}_j - \hat{a}_j^\dagger\right), \quad \sigma_j^z = 1 - 2\hat{a}_j^\dagger \hat{a}_j, \tag{B.207}$$

as simply the spin-1/2 spin operators in the Schwinger representation, the Hamiltonian can be written as

$$\hat{H} = -\frac{J}{2}\sum_{\langle i,j\rangle}\left(\hat{\sigma}_i^x \hat{\sigma}_j^x + \hat{\sigma}_i^y \hat{\sigma}_j^y\right) + \frac{\mu}{2}\sum_i \sigma_i^z, \tag{B.208}$$

which is the Heisenberg model, Eq. (B.150), with $J_x = J_y = J/2$, $J_z = 0$, and $h = \mu/2$, known as the *XX model*. Using the Jordan-Wigner transformation, we find the free fermion dispersion $E_k = -2J\cos(ka) - \mu$. We can thus express the number expectation in terms of the magnetization of the XX model as

$$\rho \propto \langle \hat{a}_i^\dagger \hat{a}_i \rangle = \frac{1}{2}\left(1 - \langle \hat{\sigma}_i^z \rangle\right) = \langle \hat{f}_i^\dagger \hat{f}_i \rangle = \begin{cases} 0 & \mu \le -2J \\ 1 - (1/\pi)\cos^{-1}(\mu/2J) & |\mu| \le 2J \\ 1 & \mu \ge 2J \end{cases} \tag{B.209}$$

In the scaling region $\mu \sim 2J$, the compressibility $\kappa = \frac{\partial \rho}{\partial \mu}$ thus diverges as $\kappa \sim |\mu - \mu_c|^{1/2}$, the critical exponent of $\frac{1}{2}$ placing this transition in the same universality class as the transitions between fully polarized and partially polarized spin states in the XX model.

Using the number conserving code for some small number of sites (~ 10) and $U/J = 20$, compute the energy of the system for $N = 1, 2, ..., 2L$. We can get the chemical potentials from $\mu(N) = E(N+1) - E(N)$. Plot μ versus the filling $\rho = N/L$. Identify the Mott gap. Is it where you expected? Why is the chemical potential negative below half filling? Does the behavior near the critical point behave with the proper scaling? If not, why?

B.4.1.2 Exercise. Bose Hubbard Hamiltonian in a Trap

Using the `extPot` global variable, define a harmonic trap and include it in the Hamiltonian (see Sec. B.9.3.4 for implementation details). Repeat the analysis of the above exercise. What can you conclude? Define the local compressibility as

$$\kappa_i = \sqrt{\langle \hat{n}^2 \rangle - \langle \hat{n} \rangle^2}\,, \tag{B.210}$$

and study this quantity. How does the physics change as you move across the trap?

B.4.1.3 Exercise. Phase Coherence and Visibility

We define the *visibility* ν as

$$\nu = \frac{I_{\text{max}} - I_{\text{min}}}{I_{\text{max}} + I_{\text{min}}}\,, \tag{B.211}$$

where I is

$$I\left(q\right) = \frac{1}{L} \sum_{j,k=1}^{L} e^{-i(j-k)q} \langle \hat{a}_i^\dagger \hat{a}_j \rangle\,, \tag{B.212}$$

q being one of the allowed quasimomenta: $q = \frac{2\pi k}{L}$, $k = 0, 1, ..., L-1$. We can interpret I either as being the occupation number of the Bloch state with quasimomentum q or simply as the Fourier transform of the boson green's function. As suggested by the name visibility, ν gives an indication of the visibility of matter wave interference fringes arising from the superfluid phase. Compute this quantity for a system deep in the Mott phase $U/J = 20$ and in the superfluid phase $U/J = 1$ and comment on the phase coherence of the system. Also investigate what happens in the trapped case.

B.4.2 Case Study 2:The Bose-Hubbard Hamiltonian–Quench Dynamics and Loschmidt Echo.

In this Case study, we investigate the dynamics a Bose-Hubbard system, initially in the Mott phase ($U/J = 10$), as U/J is quenched linearly to 1 in a time $t_R/2$ and then back to the starting value. We characterize the system by its "Loschmidt echo" $|\langle \Psi(0) | \Psi(t) \rangle|^2$, which provides the most definitive measure for how adiabatic the quench and return process is. Shown in Figure B.11 is this quantity for a six-site, six-particle system and a variety of quench times. We see that for rapid quench times the system does not return to its initial state, but for longer quench times it will. The calculations are performed by the case study `BoseHubbard_Quench` in `Bose_Hubbard_Wrapper`.

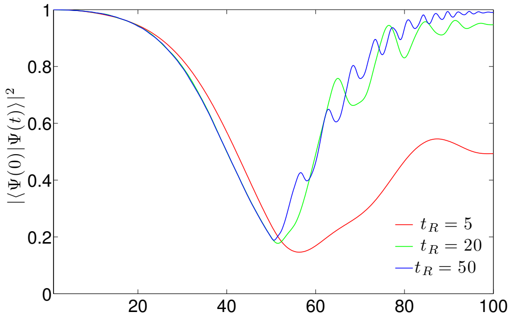


Figure B.11: Loschmidt echo vs. $100t/t_R$ for $L = N = 6$, $\chi = 50$. Note that the more rapid quench has difficulty returning to the initial state, whereas the more adiabatic quenches recover more easily.

B.4.2.1 Exercise. Finite Size Effects During a Quench.

Investigate the behavior of the Loschmidt echo as a function of the ramp time and system size. Be careful that your calculations are well converged (the cumulative truncation error being $< 10^{-9}$ is a rough estimate). Do you need much higher χ or longer t_R as you increase the system size? Why might this be so? Estimate the scaling of t_R with the system size.

B.4.3 Case Study 3:Spinless Fermions

In this section we examine the behavior of a chain of tight-binding spinless fermions. The main program we use to do so is `fermionITP.f90` located in the `Fermi_Wrapper` subdirectory of the `Case_Studies` directory. We use imaginary time to find the ground state of a 50-site system with 29 fermions and open boundary conditions (note that periodic boundary conditions are not supported for fermionic systems), and then measure the on-site numbers and deviations, as well as the single particle density matrix. We compute the single particle density matrix using both the `corr` (phaseless) part of the `measure` derived type and the `Fermicorr` part, which gives the correct Fermi phases to emphasize the drastic difference. The results are shown in Figure B.12-Figure B.13.

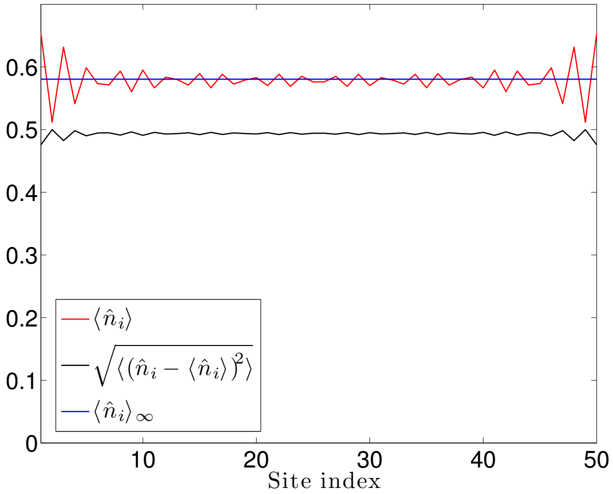


Figure B.12: $\langle\hat{n}\rangle$ and deviations for a system of spinless fermions. The blue line is the prediction for an infinite homogenous system of spinless fermions. Near the boundaries the density oscillates, a phenomenon known as Friedel oscillations.

B.4.3.1 Exercise. Interacting Fermions.

The Pauli principle prohibits spinless fermions from interacting on-site, but they can still interact via longer-range potentials. A model nearest-neighbor density-

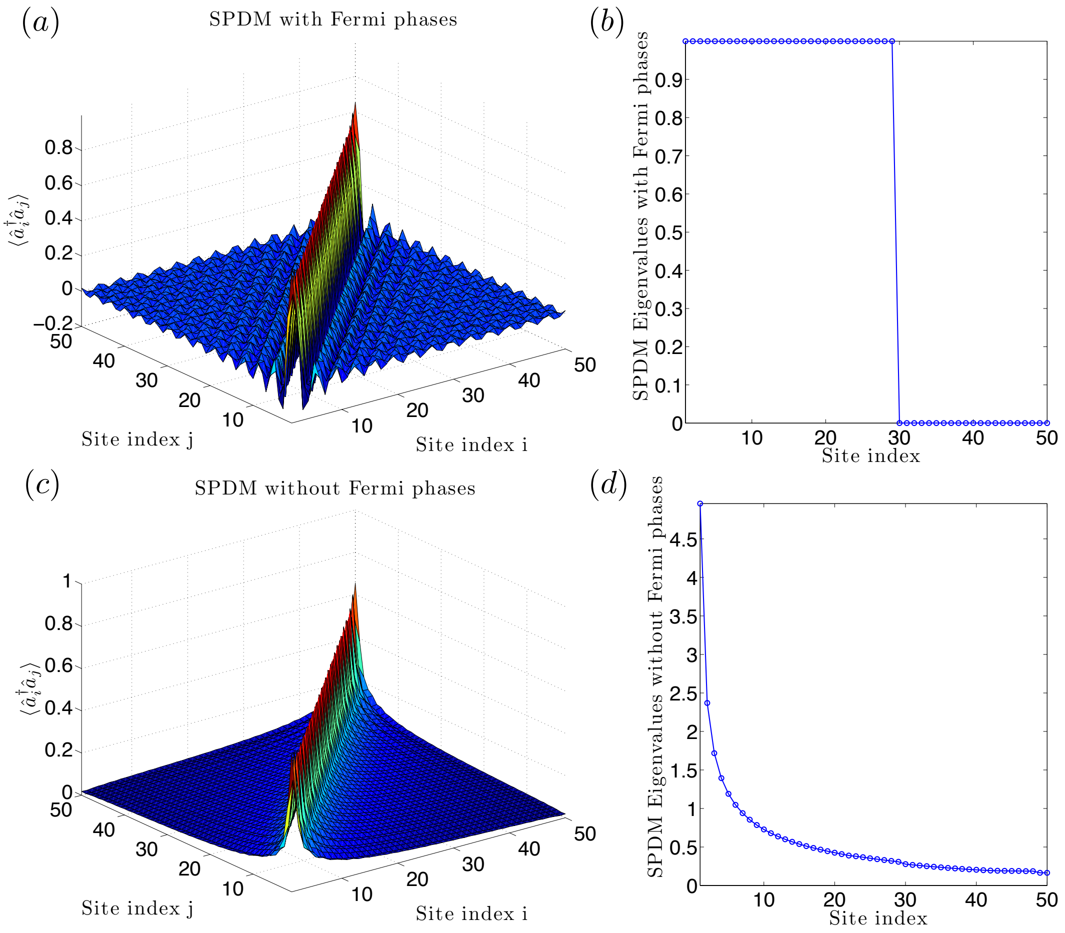


Figure B.13: Comparison of single-particle density matrices and their eigenvalues for noninteracting spinless fermions with and without proper Fermi phases. (a) Single-particle density matrix with Fermi phases for a system of spinless fermions. Note the many nodes present in this function. (b) Eigenvalue spectrum of the single particle density matrix with Fermi phases. These eigenvalues give the occupations of single particle orbitals. In this case they give a Fermi distribution, as was to be expected. (c) Single-particle density matrix without Fermi phases (Wrong!). This function has no nodes, as was the case in the bosonic system. (d) Eigenvalue spectrum of the single particle density matrix without Fermi phases (Wrong!). In this case we see macroscopic occupation of a single particle orbital-the hallmark of Bose condensation.

density interaction $V\sum_{\langle i,j\rangle}\hat{n}_i\hat{n}_j$ is controlled by the parameter `V0` in the `fermionITP.nml` input file. Study what happens as you increase the interactions.

B.4.4 Case Study 4:Dynamics of the XX Spin Chain

Download the paper "Real-time dynamics in spin-1/2 chains with adaptive time-dependent DMRG" available from Prof. Ulrich Schollwöck's website. In this paper they perform a thorough analysis of the sources of error in time-dependent DMRG by using an exactly known result concerning dynamics of a spin chain. Their results apply equally as well for TEBD. The main program `XXDynamics.f90` found in the `Heisenberg_Wrapper` subdirectory of `Case_Studies` performs the real time propagation described in the paper. By changing the parameters in the input file `XXDynamics.nml`, repeat their study of the error as a function of δt and time, and observe the behavior of the runaway time as a function of χ. As exercises, study the same details using the fifth order trotter decomposition instead of the second order decomposition, and investigate the effects of using periodic over open boundary conditions (restricting your simulations to times over which the boundary effects cannot reach the spins in question).

B.5 OpenSourceTEBD_v2.0

B.5.1 Package Layout

The OpenSourceTEBD package is written in a Core/Wrapper structure which allows the sharing of core procedures and variables. The core routines and variables are stored in `VidalCore_v_2.0`, `HamiltonianCore_v_2.0`, and `ToolsCore_v_2.0`, and are enumerated below.

B.5.1.1 Cores

`ToolsCore_v_2.0` contains input/output routines which are useful to any program. Specifically it contains:

1. `io_module.f90`: Contains i/o routines. See Sec. B.9.10 for details on these routines.

2. `timing_module.f90` has been deleted. The Fortran routine `CPU_TIME` (see this website for implementation details) may be used in its place. Examples of its use may also be found in the case study main programs.

`HamiltonianCore_v_2.0` contains routines which define Fock spaces, operators, and Hamiltonians in TEBD form. Specifically it contains:

1. `Hamiltonian_tools_module.f90`: This module contains declarations of all of the operators that are used in the various hamiltonians, the procedures to define initial states, and routines to compute vector-coupling coefficients. For information on the operators, see Sec. B.8.2, and for information on the routines see Sec. B.9.2.

2. `Heisenberg_module.f90`: This module contains routines to define the operators, Fock space, and Hamiltonian associated with the Heisenberg Chain of Sec. B.3.7.1. For details on the routines see Sec. B.9.5.

3. `Bose_hubbard_module.f90`: This module contains routines to define the operators, Fock space, and Hamiltonian associated with the Bose Hubbard Model of Sec. B.3.7.2. For details on the routines see Sec. B.9.3.

4. `Fermi_hubbard_module.f90`: This module contains routines to define the operators, Fock space, and Hamiltonian associated with the Hubbard Model of Sec. B.3.7.3. For details on the routines see Sec. B.9.4.

5. `spinS_module.f90`: This module contains routines to define the operators, Fock space, and Hamiltonian associated with the spin-s Bose Hubbard Model of Sec. B.3.7.4. For details on the routines see Sec. B.9.6.

6. `rotation_module.f90`: This module contains routines to define the operators, Fock space, and Hamiltonian associated with the Molecular Hubbard Model of Sec. B.3.7.5. For details on the routines see Sec. B.9.7.

`VidalCore_v_2.0` contains declarations of the derived types used in the package, allocation and deallocation routines, and the main routines to perform TEBD. Specifically it contains:

1. `TEBDtools_module.f90`: This module contains declarations of the derived types used in the package, allocation and deallocation routines, and functions that are generally useful. For listings of the derived types, see Sec. B.7, and for procedure listings see Sec. B.9.1.

2. `local_operations_module.f90`: This module contains the routines which perform one and two site operations on a state in the Vidal representation both with and without number conservation. See Sec. B.9.8 for details on these routines.

3. `observables_module.f90`: This module contains routines that extract observables from a state in the Vidal representation. See Sec. B.9.9 for details on these routines.

4. `propagation_module.f90`: This module contains routines that propagate a state in the Vidal representation in real or imaginary time. See Sec. B.9.11 for details on these routines.

`ExtensionsCore_v_2.0` contains routines which are under development and not used in a general program. Those that are interested in helping develop or using these routines should contact the author. Specifically it contains:

1. `PDtools_module.f90`: This module generates a Bose-Hubbard or Hubbard phase diagram by parallelizing over parameter space using MPI.
2. `SitesParallelSetup_Module.f90`: This module is a first attempt at intrinsic parallelization. The tensor network is divided among processors, allowing the trotter step to be parallelized.

B.5.1.2 Wrappers

Wrappers obey the following rules:

1. Each wrapper should contain its own readme, describing the purpose and implementation of the program(s).
2. Each wrapper should contain a makefile with primary rules to make all of the main programs in the wrapper. The makefile should reference all modules in the core libraries and any additional ones in the wrapper itself.
3. The makefile should have the option 'make clean' to remove all object files in the wrapper.
4. Each main program in the wrapper should have a `NAMELIST` (.nml) input file.
5. Each wrapper should contain its own copy of `system_parameters.f90` to set its own defaults.
6. If a program is designed to run on a cluster, a qsub or similar script should be included in the wrapper.

The Wrappers contained in the v2.0 release are Case studies found in the directory `Case_Studies`. They are

1. `Heisenberg_Wrapper`, which contains main programs to illustrate the Heisenberg Case studies in Sec. B.4.
2. `Bose_Hubbard_Wrapper`, which contains main programs to illustrate the Bose Hubbard Case studies in Sec. B.4.
3. `Fermi_Wrapper`, which contains main programs to illustrate the spinless fermion Case study in Sec. B.4.

B.6 Frequently Asked Questions

B.6.1 Regarding Fortran90+

1. Q: What does the `_rKind` at the end of real numbers mean?

   A: One of the problems often encountered when porting programs from one platform to another is that the terms "single precision" and "double precision" are not precisely defined. In most cases, single precision refers to a 32 bit long value and double precision to a 64 bit long value, but on some platforms, most notably Cray supercomputers and hardware based on the 64-bit Intel Itanium chip, the bit values are twice this. Fortran 90+ has a built in means to automatically select the proper kind of real value to use based on the required precision and range. It takes the form of an integer valued function `SELECTED_REAL_KIND(p=precision,r=range)` which returns the smallest type of real value that has `precision` decimal digits of accuracy and a range of the exponent (in powers of 10) of `range`. In the code, we refer to the specified real kind with global variable `precis` digits of precision and a range of `range` as being `KIND=rKind`, with the real unit of this kind being `1.0_rKind` and so on.

2. Q:What is a derived type?/What is this `%t` thing that follows Gammas?

   A:In addition to the intrinsic data types: REAL, INTEGER, etc. , Fortran 90+ gives us a means to create our own data types and define operations on them.

Such an object is called a *derived type*. For example, in `TEBDtools_module`, we define the TYPE vector as

```
TYPE vector
REAL(KIND=rKind), POINTER :: v(:)
END TYPE vector
```

so that when we define an object, let's call it `my_vec`, as being TYPE(vector), e.g.

```
TYPE(vector) :: my_vec
```

`my_vec` is now a REAL 1-D pointer. This allows us to dynamically allocate and deallocate such 1-D objects as

```
ALLOCATE(my_vec%v(vecsize))
...
DEALLOCATE(my_vec%v)
```

providing us with a great deal of flexibility, as we often do not know *a priori* what the size of our objects needs to be. Note that the TYPE(vector) object `my_vec` is followed by `%v`, as was specified by calling the general TYPE(vector) "`v(:)`" in the definition above. This additional piece is known as the component selector of that derived type.

Perhaps the greatest utility of derived types for our code is that they allow us to create linked lists of non-scalar objects. For example, consider the four-index local tensors $\Gamma^{[l]i_l}_{\alpha_l\alpha_{l+1}}$. There are occasions when we wish to pass the entire set of Gammas to a procedure, but there are also occasions when we wish to pass a Gamma corresponding to a single site. We facilitate this by defining a TYPE(tensor) as

```
TYPE tensor
COMPLEX(KIND=rKind), POINTER :: t(:,:,:)
END TYPE tensor
```

and then defining a list of TYPE(tensor)s

```
TYPE(tensor), POINTER :: Gammas(:)
```

We now specify how many three-index Gammas we want as

```
ALLOCATE(Gammas(systemSize))
```

and then how large each Gamma is as

```
DO i=1,systemSize
ALLOCATE(Gammas(i)%t(chi,localSize,chi))
END DO
```

`Gammas(l)%t(j,i,k)` refers to the element with site index l, Schmidt indices j and k, and on-site index i. If we wish to pass all of the Gammas into a procedure, then we call the procedure as

```
CALL Procedure(...,Gammas,...)
```

whereas if we wish to pass only the Gamma on site l, we call the procedure as

```
CALL Procedure(...,Gammas(l)%t,...)
```

This flexibility coupled with the dynamic allocation abilities of derived types clearly display their utility. A listing of all derived types and their component selectors is found in Sec. B.7.

3. Q: What are these INTERFACEs doing at the top of a module?

   A: The interface allows us to define a generic procedure, one which chooses between the following MODULE PROCEDURES based on the number, KIND, etc of arguments present when the interface is called. This allows for construction of procedures that can accept real or complex arguments etc.

B.6.2 Regarding Code Output

1. Q: The code says "*** Cannot open file named ..."

   A: One of a few things will cause this. First of all, make sure that the directory beginning the file name exists. Second, be sure that `openUnit` is being called properly. If the file already exists and you call `openUnit(...,'N')`, the program will give this error, as it is looking to open a new file and the file already exists. I have designed the program to do this so that old data is not carelessly overwritten. To overwrite any old data, simply leave the optional character at the end of `openUnit` out (see Sec. B.9.10.5 for implementation details).

Please ask more!

B.7 Derived Type Listings

B.7.1 TEBDtools_module Derived Type Listings

Contents:

```
TYPE vector
TYPE vectorComplex
TYPE matrix
TYPE tensor
TYPE vectorInt
TYPE matrixInt
TYPE matrixReal
TYPE mlocal
TYPE mavg
```

`TYPE mcorr`

`TYPE mcorrf`

`YPE entropy`

B.7.1.1 vector

Description:

The derived type `vector` is a POINTER to a single-index array of REAL(KIND=rKind). The component selector is `v`.

B.7.1.2 vector

Description:

The derived type `vectorComplex` is a POINTER to a single-index array of COMPLEX(KIND=rKind). The component selector is `vc`.

B.7.1.3 matrix

Description:

The derived type `matrix` is a POINTER to a two-index array of COMPLEX(KIND=rKind). The component selector is `m`.

B.7.1.4 tensor

Description:

The derived type `tensor` is a POINTER to a three-index array of COMPLEX(KIND=rKind). The component selector is `t`.

B.7.1.5 vectorInt

Description:

The derived type `vectorInt` is a POINTER to a single-index array of INTEGER. The component selector is `vi`.

B.7.1.6 matrixInt

Description:

The derived type `matrixInt` is a POINTER to a two-index array of INTEGER. The component selector is `mi`.

B.7.1.7 matrixReal

Description:

The derived type `matrixReal` is a POINTER to a two-index array of REAL(KIND=rKind). The component selector is `mr`.

B.7.1.8 mlocal

Description:

The derived type `mlocal` consists of an allocatable COMPLEX(KIND=rKind) rank-two array with component selector `Op` and an allocatable COMPLEX(KIND=rKind) rank-one array with component selector `value`. It is part of the `measure` derived type, see Sec. B.3.6.6.

B.7.1.9 mavg

Description:

The derived type `mavg` consists of an allocatable COMPLEX(KIND=rKind) rank-two array with component selector `Op` and a COMPLEX(KIND=rKind) with component selector `value`. It is part of the `measure` derived type, see Sec. B.3.6.6.

B.7.1.10 mcorr

Description:

The derived type `mcorr` consists of an allocatable COMPLEX(KIND=rKind) rank-two array with component selector `Op` and an allocatable COMPLEX(KIND=rKind) rank-two array with component selector `value`. It is part of the `measure` derived type, see Sec. B.3.6.6.

B.7.1.11 mcorrf

Description:

The derived type `mcorrf` consists of an allocatable COMPLEX(KIND=rKind) rank-two array with component selector `Op` and an allocatable COMPLEX(KIND=rKind) rank-two array with component selector `value`. It is part of the `measure` derived type, see Sec. B.3.6.6.

B.7.1.12 entropy

Description:

The derived type `entropy` consists of a REAL(KIND=rKind) with component selector `qme`, an allocatable COMPLEX(KIND=rKind) rank-one array with component selector `vN`, an allocatable COMPLEX(KIND=rKind) rank-one array with component selector `chain`, and an allocatable COMPLEX(KIND=rKind) rank-two array with component selector `tbvN`. It is part of the `measure` derived type, see Sec. B.3.6.6.

B.7.2 observables_module Derived Type Listings

Contents:

`TYPE measure`

B.7.2.1 measure

Description:

The derived type `measure` consists of a REAL(KIND=rKind) with component selector `en`, LOGICALs with component selectors `localpres`, `avgpres`, `entpres`, `corrpres`, and `Fermicorrpres`, a TYPE(entropy) with component selector `ent`, and pointers to TYPE(mlocal), TYPE(mavg), TYPE(mcorr), and TYPE(mcorrf) with component selectors `local`, `avg`, `corr`, and `Fermicorr`, respectively. The usage of this derived type is discussed in Sec. B.3.6.6.

B.8 Global Variable Listings

B.8.1 system_parameters Global Variable Listings

Contents:

`precis`

`range`

`rKind`

`systemSize`

`maxFilling`

`trotterOrder`

`chiMin`

`chiMax`

`dtITP`

`stepsForJudge`

`maxITPsteps`

`convCriterion1`

`convCriterion2`

`dtRTP`

`totalStep`

`stepsForStore`

`localSize`

`itpDir`

`itpExt`

`rtpDir`

`rtpExt`

`pdDir`

`print_switch`

`ncSwitch`

`ITPreadMPDSwitch`

`ITPwriteMPDSwitch`

`BoundaryCond`

`ITPopenKind`

`statInt`

`fileStatus`

`totNum`

`U2`

`VB`

B.8.1.1 precis

Description:

`precis` is an INTEGER PARAMETER which is the number of decimal digits of precision required for real variables. It is used to define `rKind` through the `SELECTED_REAL_KIND` construct, see Sec. B.6. The default value is 15.

B.8.1.2 range

Description:

`range` is an INTEGER PARAMETER which is the range of the exponent required for real variables in powers of 10. It is used to define `rKind` through the `SELECTED_REAL_KIND` construct, see Sec. B.6. The default value is 30.

B.8.1.3 rKind

Description:

`rKind` is an INTEGER PARAMETER which defines the KIND of REAL variables used in the code through the `SELECTED_REAL_KIND` construct, the global precision `precis`, and the global range `range`, see Sec. B.6.

B.8.1.4 systemSize

Description:

`systemSize` is an INTEGER which defines the total number of lattice sites. The default value is 4.

B.8.1.5 maxFilling

Description:

`maxFilling` is an INTEGER which defines the total number of particles allowed on a single site. The default value is 1.

B.8.1.6 trotterOrder

Description:

`trotterOrder` is an INTEGER which defines the order of the Trotter expansion used for time evolution. The default value is 2. It may only be 2 or 5, see Secs. B.3.5.1 and B.3.5.1.

B.8.1.7 chiMin

Description:

`chiMin` is an INTEGER which defines the entanglement cutoff parameter used in the first iteration of imaginary time propagation. The default value is 2.

B.8.1.8 chiMax

Description:

`chiMax` is an INTEGER which defines the entanglement cutoff parameter used in the final iteration of imaginary time propagation. The default value is 5.

B.8.1.9 dtITP

Description:

`dtITP` is a REAL(KIND=rKIND) which defines the "infinitesimal" imaginary time step. The default value is 0.0001_rKind

B.8.1.10 stepsForJudge

Description:

`stepsForJudge` is an INTEGER which gives the number of imaginary time steps between convergence tests. The default value is 100

B.8.1.11 maxITPsteps

Description:

`maxITPsteps` is an INTEGER which gives the total number of imaginary time steps allowed. The default value is 2000.

B.8.1.12 convCriterion1

Description:

`convCriterion1` is a REAL(KIND=rKIND) which defines the convergence criterion for the first iteration of imaginary time propagation. The first iteration of imaginary time propagation is said to have converged if the difference between all $\lambda_{\alpha}^{[l]}$ at imaginary time τ and imaginary time $\tau+$`stepsForJudge`$*$`dtITP` is less than `convCriterion1`. The default value is 0.00001_rKind.

B.8.1.13 convCriterion2

Description:

`convCriterion2` is a REAL(KIND=rKIND) which defines the convergence criterion for the last iteration of imaginary time propagation. The last iteration of imaginary time propagation is said to have converged if the difference between all $\lambda_{\alpha}^{[l]}$ at imaginary time τ and imaginary time $\tau+$`stepsForJudge`$*$`dtITP` is less than `convCriterion2`. The default value is 0.000001_rKind.

B.8.1.14 dtRTP

Description:

`dtRTP` is a COMPLEX(KIND=rKIND) which defines the “infinitesimal” real time step. The default value is 0.1_rKind

B.8.1.15 totalStep

Description:

`totalStep` is an INTEGER which gives the total number of real time steps allowed. The default value is 10000.

B.8.1.16 stepsForStore

Description:

`stepsForStore` is an INTEGER which gives the number of real time steps between computation and output of a set of observables. The default value is 10

B.8.1.17 localSize

Description:

`localSize` is an INTEGER which defines the size of the local Hilbert space, denoted d in theoretical discussions. Each Hamiltonian has its own local dimension function, for example `BoseHubbardLocalDim` (Sec. B.9.3.1) for the Bose-Hubbard model.

B.8.1.18 itpDir

Description:

`itpDir` is a CHARACTER(32) which gives the directory for output files generated during imaginary time propagation. The default directory is “ITPDATA/”.

B.8.1.19 itpExt

Description:

`itpExt` is a CHARACTER(32) which gives the extension for output files generated during imaginary time propagation. The default extension is “.dat”.

B.8.1.20 rtpDir

Description:

`rtpDir` is a CHARACTER(32) which gives the directory for output files generated during real time propagation. The default directory is “RTPDATA/”.

B.8.1.21 rtpExt

Description:

`rtpExt` is a CHARACTER(32) which gives the extension for output files generated during real time propagation. The default extension is “.dat”.

B.8.1.22 pdDir

Description:

`pdDir` is a CHARACTER(32) which gives the directory for output files generated by the phase diagram extension. The default directory is “PD_DATA/”.

B.8.1.23 print_switch

Description:

`print_switch` is a LOGICAL which controls non-error output to the screen. When `print_switch`=.TRUE. messages are printed to the screen, and when `print_switch`=.FALSE. printing to the screen are suppressed. The default value is .TRUE..

B.8.1.24 ncSwitch

Description:

`ncSwitch` is a LOGICAL which toggles the number conserving option. When `ncSwitch`=.TRUE. the number conserving method is used, and when `ncSwitch`=.FALSE. the number non-conserving method is used. The default value is .FALSE..

B.8.1.25 ITPreadMPDswitch

Description:

`ITPreadMPDswitch` is a LOGICAL which toggles the generation of an initial state for imaginary time propagation. When `ITPreadMPDswitch`=.TRUE. the initial state is read from file (if possible), and when `ITPreadMPDswitch`=.FALSE. the initial state is generated using one of the methods of Sec. B.3.3.2. The default value is .FALSE.. This variable is not used in any of the case studies.

B.8.1.26 ITPwriteMPDswitch

Description:

`ITPwriteMPDswitch` is a LOGICAL which toggles the output of a state after imaginary time propagation. When `ITPwriteMPDswitch`=.TRUE. the initial state is written to file, and when `ITPwriteMPDswitch`=.FALSE. it is not. The default value is .FALSE.. This variable is not used in any of the case studies.

B.8.1.27 BoundaryCond

Description:

`BoundaryCond` is a CHARACTER which toggles the boundary conditions. When `BoundaryCond`='O' the code uses open boundary conditions and when `BoundaryCond`='P' the code uses periodic boundary conditions. The default value is 'O' Periodic boundary conditions are not supported for fermionic systems.

B.8.1.28 ITPopenKind

Description:

`ITPopenKind` is a CHARACTER which toggles the format of output states. When `ITPopenKind`='S' the state is output in human readable scientific notation, and when `ITPopenKind`='B' the state is output in binary. The default value is 'S'. This variable is not used in any of the case studies.

B.8.1.29 statInt

Description:

`statInt` is an INTEGER used as an error flag when dynamically allocating or deallocating memory.

B.8.1.30 fileStatus

Description:

`fileStatus` is an INTEGER used as an error flag when opening files.

B.8.1.31 totNum

Description:

`totNum` is an INTEGER which defines the total number of particles when the number conserving method is being used. The default value is 4.

B.8.2 Hamiltonian_tools_module Global Variable Listings

Contents:

`jTunn`

`U0`

`mu0`

`V0`

`extPot`

`spin`

`spinSize`

`one_op`

`a_op`

`t_op`

`PBphase_op`

`FermiPhase_op`

`Conserv`

`lFac`

`a_opS`

`Sx_opS`

`Sy_opS`

`Sz_opS`

`Ssq_opS`

`VB_opS`

`ntot_opS`

`ttot_opS`

B.8.2.1 jTunn

Description:

`jTunn` is a REAL(KIND=rKind) giving the tunneling energy in the Bose-Hubbard and Hubbard Hamiltonians, Sec. B.3.7.2 and Sec. B.3.7.3.

B.8.2.2 U0

Description:

`U0` is a REAL(KIND=rKind) giving the on-site repulsion energy in the Bose-Hubbard Hamiltonian, Sec. B.3.7.2, the Hubbard Hamiltonian, Sec. B.3.7.3, and the spin-1 Bose-Hubbard model Sec. B.3.7.4.

B.8.2.3 mu0

Description:

`mu0` is a REAL(KIND=rKind) giving the chemical potential in the Bose-Hubbard Hamiltonian, Sec. B.3.7.2, the Hubbard Hamiltonian, Sec. B.3.7.3, and the spin-s Bose-Hubbard model Sec. B.3.7.4.

B.8.2.4 V0

Description:

`V0` is a REAL(KIND=rKind) giving the nearest neighbor repulsion energy in the (extended) Bose-Hubbard Hamiltonian, Sec. B.3.7.2.

B.8.2.5 extPot

Description:

`extPot` is an ALLOCATABLE REAL(KIND=rKind), DIMENSION(:) giving the external potential in the Bose-Hubbard Hamiltonian, Sec. B.3.7.2, or the Hubbard Hamiltonian, Sec. B.3.7.3.

B.8.2.6 spin

Description:

`spin` is a REAL(KIND=rKind) which defines the spin of the constituent particles in the spin-s Bose-Hubbard Hamiltonian (Sec. B.3.7.4) or the Hubbard Hamiltonian, Sec. B.3.7.3.

B.8.2.7 spinSize

Description:

`spinSize` is an INTEGER which defines the number of allowed states per particle of the constituent particles in the spin-s Bose-Hubbard Hamiltonian (Sec. B.3.7.4) or the Hubbard Hamiltonian, Sec. B.3.7.3. It is simply a convenient integer shorthand for 2`spin`+1.

B.8.2.8 one_op

Description:

`one_op` is the one-site unit operator, $\hat{1}$, used in systems with and without internal degrees of freedom. It is constructed in `CreateHeisenbergOps`, Sec. B.9.5.2; `CreateFieldOps`, Sec. B.9.3.2; `CreateSpinSops`, Sec. B.9.6.2; `CreateRotationops`, Sec. B.9.7.7; and `CreateRotationopsMzero`, Sec. B.9.7.5.

KIND:

TYPE(matrixReal)

B.8.2.9 a_op

Description:

`a_op` is the one-site destruction operator for systems without internal degrees of freedom, $\hat{a}_i$. It is constructed in `CreateFieldOps`, Sec. B.9.3.2.

KIND:

TYPE(matrixReal)

B.8.2.10 t_op

Description:

`t_op` is the two-site tunneling operator for systems without internal degrees of freedom, defined as $\hat{t} = \left(\hat{a}^{\dagger}_{i+1}\hat{a}_i + \text{h.c.}\right)$. It is constructed in `CreateFieldOps`, Sec. B.9.3.2.

KIND:

TYPE(matrixReal)

B.8.2.11 PBphase_op

Description:

`PBphase_op` is the Pegg-Barnett Hermitian phase operator, defined in number representation as[22]

$$\hat{\theta} = \theta_o + \frac{(d-1)\,\pi}{d} + \frac{2\pi}{d}\sum_{n\neq n'}\frac{\exp\left[i\,(n'-n)\,\theta_o\right]}{\exp\left[i\,(n'-n)\,2\pi/d\right]-1}|n'\rangle\langle n|\,, \tag{B.213}$$

where we set the reference phase θ_o to zero. It is constructed in `CreateFieldOps`, Sec. B.9.3.2.

KIND:

TYPE(matrix)

B.8.2.12 FermiPhase_op

Description:

`FermiPhase_op` is $(-1)^{\hat{n}}$, where $\hat{n}$ is the number operator for a single site. It is used to construct operators which anticommute on different sites for fermionic codes, and is constructed in `CreateFermiSOps`, B.9.4.2.

KIND:

TYPE(matrixReal)

B.8.2.13 Conserv

Description:

`Conserv` is a vector that stores a the number associated with an on-site state, used in number-conserving code with internal degrees of freedom, see Sec. B.3.4.3. It's i^{th} component is the quantum number for on-site state i. It is constructed in `CreateHeisenbergOps`, Sec. B.9.5.2;`CreateSpinSops`, Sec. B.9.6.2; `CreateRotationops`, Sec. B.9.7.7; and `CreateRotationopsMzero`, Sec. B.9.7.5

KIND:

TYPE(vectorInt)

B.8.2.14 lFac

Description:

`lFac` is an array, indexed from 1 to 200, whose i^{th} component is $\log\left[(i-1)!\right]$. It is initialized in `SetupLogFac`, Sec. B.9.2.7.

KIND:

REAL(KIND=rKind), DIMENSION(:)

B.8.2.15 a_opS

Description:

`a_opS` is the one-site destruction operator for systems with internal degrees of freedom, $\hat{a}_{i,\alpha}$. It is constructed in `CreateHeisenbergOps`, Sec. B.9.5.2; `CreateSpinSops`, Sec. B.9.6.2 and `CreateRotationops`, Sec. B.9.7.7.

KIND:

TYPE(matrixReal), POINTER, DIMENSION(:)

B.8.2.16 Sx_opS

Description:

`Sx_opS` is the one-site spin operator in the x direction, $\hat{S}_x$, defined as in Eq. (B.159). It is constructed in `CreateHeisenbergOps`, Sec. B.9.5.2 and `CreateSpinSops`, Sec. B.9.6.2.

KIND:

TYPE(matrixReal)

B.8.2.17 Sy_opS

Description:

`Sy_opS` is the one-site spin operator in the y direction, $\hat{S}_y$, defined as in Eq. (B.159). It is constructed in `CreateHeisenbergOps`, Sec. B.9.5.2 and `CreateSpinSops`, Sec. B.9.6.2. Note that this operator is (necessarily) complex, unlike `Sx_opS` and `Sz_opS`.

KIND:

TYPE(matrix)

B.8.2.18 Sz_opS

Description:

`Sz_opS` is the one-site spin operator in the z direction, $\hat{S}_z$, defined as in Eq. (B.159). It is constructed in `CreateHeisenbergOps`, Sec. B.9.5.2 and `CreateSpinSops`, Sec. B.9.6.2.

KIND:

TYPE(matrixReal)

B.8.2.19 Ssq_opS

Description:

`Ssq_opS` is the one-site total spin operator, defined as $\hat{S}_z^2+\hat{S}_y^2+\hat{S}_z^2$. It is constructed in `CreateHeisenbergOps`, Sec. B.9.5.2 and `CreateSpinSops`, Sec. B.9.6.2.

KIND:

TYPE(matrixReal)

B.8.2.20 VB_opS

Description:

`VB_opS` is the quadratic Zeeman operator, defined as in Eq. (B.161). It is constructed in `CreateSpinSops`, Sec. B.9.6.2.

KIND:

TYPE(matrixReal)

B.8.2.21 ntot_opS

Description:

`ntot_opS` is the total one-site number operator for systems with spin degrees of freedom, defined as $\hat{n}_i = \sum_{\alpha} \hat{a}^{\dagger}_{i,\alpha}\hat{a}_{i,\alpha}$. It is constructed in Sec. B.9.5.2 and `CreateSpinSops`, Sec. B.9.6.2.

B.8.2.22 ttot_opS

Description:

`ttot_opS` is the total two-site tunneling operator for systems with spin degrees of freedom, defined as $\hat{t}_i = \sum_{\alpha}\left(\hat{a}^{\dagger}_{i+1,\alpha}\hat{a}_{i,\alpha} + \text{h.c.}\right)$. It is constructed in `CreateSpinSops`, Sec. B.9.6.2.

B.8.3 Heisenberg_module Global Variable Listings

Contents:

`Jx`

`Jy`

`Jz`

`magH`

B.8.3.1 Jx

Description:

`Jx` is a REAL(KIND=rKind) giving the coupling of the x-component of spin in the Heisenberg spin chain, Sec. B.3.7.1.

B.8.3.2 Jy

Description:

`Jy` is a REAL(KIND=rKind) giving the coupling of the y-component of spin in the Heisenberg spin chain, Sec. B.3.7.1.

B.8.3.3 Jz

Description:

`Jz` is a REAL(KIND=rKind) giving the coupling of the z-component of spin in the Heisenberg spin chain, Sec. B.3.7.1.

B.8.3.4 magH

Description:

`magH` is a REAL(KIND=rKind) giving the strength of the magnetic field in the Heisenberg spin chain, Sec. B.3.7.1.

B.8.4 spinS_module Global Variable Listings

Contents:

`U2`

`VB`

B.8.4.1 U2

Description:

`U2` is a REAL(KIND=rKind) giving the spin-dependent on-site repulsion energy in the spin-1 Bose-Hubbard model Sec. B.3.7.4.

B.8.4.2 VB

Description:

`VB` is a REAL(KIND=rKind) giving the quadratic Zeeman strength in the spin-1 Bose-Hubbard model Sec. B.3.7.4.

B.8.5 rotation_module Global Variable Listings

Contents:

```
rotLevel

rotSize

Jcut

qDC

qAC

rotConst

dip

eDC

eAC

omega

detuning

alphaBar

deltaAlpha

Udipdip

ERecoil

LattHeight

EDC_opR

EAC_opR

ttot_opR

dipdip_opR

INTEGER :: LWORK, LIWORK, LRWORK

INTEGER, ALLOCATABLE :: IWORK(:)

REAL(KIND=rKind), ALLOCATABLE :: W(:), RWork(:)

COMPLEX(KIND=rKind) ,ALLOCATABLE :: Work(:)
```

B.8.5.1 rotLevel

Description:

`rotLevel` is an INTEGER PARAMETER which defines the largest number of rotational quanta allowed per particle of the constituent particles in the Molecular Hubbard Hamiltonian (Sec. B.3.7.5).

B.8.5.2 rotSize

Description:

`rotSize` is an INTEGER which defines the number of allowed states per particle of the constituent particles in the Molecular Hubbard Hamiltonian (Sec. B.3.7.5). It is simply a convenient shorthand for (1+`rotLevel`)^2.

B.8.5.3 Jcut

Description:

`Jcut` is an INTEGER PARAMETER which defines the largest number of rotational quanta used to construct the dressed basis for the Molecular Hubbard Hamiltonian (Sec. B.3.7.5).

B.8.5.4 qDC

Description:

`qDC` is an INTEGER PARAMETER which defines the space-fixed spherical polarization for the DC electric field used in the Molecular Hubbard Hamiltonian (Sec. B.3.7.5).

B.8.5.5 qAC

Description:

`qAC` is an INTEGER PARAMETER which defines the space-fixed spherical polarization for the AC electric field used in the Molecular Hubbard Hamiltonian (Sec. B.3.7.5).

B.8.5.6 rotConst

Description:

`rotConst` is a REAL(KIND=rKind) PARAMETER which defines the rotational constant of the molecular species considered in the Molecular Hubbard Hamiltonian (Sec. B.3.7.5).

B.8.5.7 dip

Description:

`dip` is a REAL(KIND=rKind) PARAMETER which defines the permanent dipole moment of the molecular species considered in the Molecular Hubbard Hamiltonian (Sec. B.3.7.5).

B.8.5.8 eDC

Description:

`eDC` is a REAL(KIND=rKind) which defines the strength of the DC electric field used in the Molecular Hubbard Hamiltonian (Sec. B.3.7.5).

B.8.5.9 eAC

Description:

`eAC` is a REAL(KIND=rKind) which defines the strength of the AC electric field used in the Molecular Hubbard Hamiltonian (Sec. B.3.7.5).

B.8.5.10 omega

Description:

`omega` is a REAL(KIND=rKind) which defines the frequency of the AC electric field used in the Molecular Hubbard Hamiltonian (Sec. B.3.7.5).

B.8.5.11 detuning

Description:

`omega` is a REAL(KIND=rKind) which defines the detuning from resonance of the AC electric field used in the Molecular Hubbard Hamiltonian (Sec. B.3.7.5).

B.8.5.12 alphaBar

Description:

`alphaBar` is a REAL(KIND=rKind) PARAMETER which defines the average polarizability of the molecular species considered in the Molecular Hubbard Hamiltonian (Sec. B.3.7.5).

B.8.5.13 deltaAlpha

Description:

`deltaAlpha` is a REAL(KIND=rKind) PARAMETER which defines the polarizability anisotropy of the molecular species considered in the Molecular Hubbard Hamiltonian (Sec. B.3.7.5).

B.8.5.14 Udipdip

Description:

`Udipdip` is a REAL(KIND=rKind) PARAMETER which defines the energy scale of the dipole-dipole term of the Molecular Hubbard Hamiltonian (Sec. B.3.7.5).

B.8.5.15 ERecoil

Description:

`ERecoil` is a REAL(KIND=rKind) PARAMETER which defines the recoil energy of an optical lattice.

B.8.5.16 LattHeight

Description:

`LattHeight` is a REAL(KIND=rKind) PARAMETER which defines the ersatz lattice height used in the scaled Molecular Hubbard Hamiltonian (Sec. B.3.7.5).

B.8.5.17 EDC_opR

Description:

`EDC_opR` is the Rotational+DC Hamiltonian operator for the MHH. It is constructed in `CreateRotationops`, Sec. B.9.7.7 and `CreateRotationopsMzero`, Sec. B.9.7.5.

KIND:

TYPE(matrixReal)

B.8.5.18 EAC_opR

Description:

`EAC_opR` is the AC Hamiltonian operator for the MHH. It is constructed in `CreateRotationops`, Sec. B.9.7.7 and `CreateRotationopsMzero`, Sec. B.9.7.5.

KIND:

TYPE(matrixReal)

B.8.5.19 ttot_opR

Description:

`ttot_opR` is the total two-site tunneling operator for systems with rotational degrees of freedom. It is constructed in `CreateRotationops`, Sec. B.9.7.7 and `CreateRotationopsMzero`, Sec. B.9.7.5.

KIND:

TYPE(matrixReal)

B.8.5.20 dipdip_opR

Description:

`dipdip_opR` is the dipole-dipole Hamiltonian operator. It is constructed in `CreateRotationops`, Sec. B.9.7.7 and `CreateRotationopsMzero`, Sec. B.9.7.5.

KIND:

TYPE(matrixReal)

Descriptions:

The remaining global variables in rotation_module are used in the LAPACK routine DSYEV to diagonalize a real symmetric matrix, and have the same names as their counterparts in the LAPACK manpages. See the webpage for DSYEV.

B.8.6 local_operations_module Global Variable Listings

Contents:

```
CHARACTER(1) :: jobu_SVD, jobvt_SVD
INTEGER :: matrixSizeSM_SVD, workSizeSM_SVD, matrixSizeLG_SVD, workSizeLG_SVD
INTEGER :: matrixSize_SVD, workSize_SVD, info_SVD, matrixSizeL_SVD, matrixSizeT_SVD
REAL(KIND=rKind), ALLOCATABLE :: rworkSM_SVD(:)
COMPLEX(KIND=rKind), ALLOCATABLE :: workSM_SVD(:)
REAL(KIND=rKind), ALLOCATABLE :: rworkLG_SVD(:)
COMPLEX(KIND=rKind), ALLOCATABLE :: workLG_SVD(:)
REAL(KIND=rKind), ALLOCATABLE :: rwork_SVD(:)
COMPLEX(KIND=rKind), ALLOCATABLE :: work_SVD(:)
```

Descriptions:

The global variables in `local_operations_module` all correspond to inputs of the LAPACK routine ZGESVD which performs the singular value decomposition on a general, possibly rectangular, COMPLEX double precision (64-bit) matrix. The implementation details are found on this website. The correspondence between our global variables and the argument names given is given in Table B.4. The variables not in this table are the sizes of variables in this table, and can be clearly determined from the procedures `SVDInit` discussed in Sec. B.9.8.1 (Number non-conserving) or `SVDInitNC` discussed in Sec. B.9.8.16 (Number conserving).

B.9 Procedure Listings

B.9.1 TEBDtools_module Procedure Listings

Contents:

Table B.4: Correspondence between global variables and ZGESVD arguments

Global variable name	`ZGESVD` Argument name
`jobu_SVD`	`JOBU`
`jobvt_SVD`	`JOBVT`
`matrixSizeLG_SVD`	`M, N, LDA, LDU, LDVT, LDWORK`
`workSizeLG_SVD`	`WORK`
`info_SVD`	`INFO`
`rworkLG_SVD`	`WORK2`

```
INTERFACE matrix_exponential

        MODULE PROCEDURE matrix_exponential_r, matrix_exponential_c

INTERFACE tensorProduct

        MODULE PROCEDURE tensorProd_r, tensorProd_c

                                                 tensorProd_rc,tensorProd_cr

                                                 tensorProd_rs, tensorProd_cs

                                                 tensorProd_rcs,tensorProd_crs

INTERFACE kronDelta

        MODULE PROCEDURE kronDelta_r, kronDelta_c

INTERFACE TraceMatmul

        MODULE PROCEDURE TraceMatmul_r,TraceMatmul_c,

                                                 TraceMatmul_rc,TraceMatmul_cr

                                                 TraceMatmul_rf,TraceMatmul_cf,

                                                 TraceMatmul_rcf,TraceMatmul_crf

FUNCTION Factorial

FUNCTION BinomialCoef

SUBROUTINE AllocateGamLam

SUBROUTINE CopyGamLam

SUBROUTINE DeallocateGamLam

SUBROUTINE AllocateLabel

SUBROUTINE CopyLabel

SUBROUTINE DeallocateLabel

SUBROUTINE AllocateIndexLR

SUBROUTINE DeallocateIndexLR
```

```
SUBROUTINE AllocateBlockTheta

SUBROUTINE DeallocateBlockTheta

SUBROUTINE AllocateUSV

SUBROUTINE DeallocateUSV

SUBROUTINE AllocateSSflat

SUBROUTINE AllocateOps

SUBROUTINE DeallocateOps

SUBROUTINE AllocateProp

SUBROUTINE DeallocateProp

INTERFACE ConstructPropagators

        MODULE PROCEDURE OLDConstructPropagators, NEWConstructPropagators
```

B.9.1.1 matrix_exponential

Description:

`matrix_exponential(A,Exp_A,tau,n)` computes $\exp\left[-\mathrm{tau}*\mathrm{A}\right]$ for a real n× n matrix A and real tau and $\exp\left[-i*\mathrm{tau}*\mathrm{A}\right]$ for complex n×n matrix A and complex tau, storing this in Exp_A. The routine assumes A is symmetric if real and Hermitian if complex. The routine diagonalizes A, computes the matrix with exponentials of the eigenvalues on the diagonal, and then transforms back to the original basis of A using the matrix with A's eigenvectors as columns. This procedure uses the LAPACK routines DSYEV and ZHEEVD for real and complex matrices, respectively.

Type:

SUBROUTINE

Arguments:

A is REAL(KIND=rKind), DIMENSION(n,n) or COMPLEX(KIND=rKind), DIMENSION(n,n)

tau is REAL(KIND=rKind) or COMPLEX(KIND=rKind) and must match the type of A

n is INTEGER

Exp_A is REAL(KIND=rKind), DIMENSION(n,n) or COMPLEX(KIND=rKind), DIMENSION(n,n) and must match the type of A and tau.

B.9.1.2 tensorProd

Description:

`tensorProd(A,B)` returns the tensor product of matrices A and B, and `tensorProd(AXB,A,B)` computes the tensor product of matrices A and B and stores it in AXB. The procedure accepts real or complex A or B.

Types:

SUBROUTINE

REAL(KIND=rKind) or COMPLEX(KIND=rKind), DIMENSION(:,:) FUNCTION

Syntaxes:

result=tensorProd(A,B)

CALL tensorProd(AXB,A,B)

Arguments:

A is REAL(KIND=rKind), DIMENSION(:,:) or COMPLEX(KIND=rKind), DIMENSION(:,:) and is INTENT(IN)

B is REAL(KIND=rKind), DIMENSION(:,:) or COMPLEX(KIND=rKind), DIMENSION(:,:) and is INTENT(IN)

AXB is REAL(KIND=rKind), DIMENSION(:,:) if both A and B are REAL(KIND=rKind), and COMPLEX(KIND=rKind), DIMENSION(:,:) otherwise.

B.9.1.3 kronDelta

Description:

`kronDelta(vec1,vec2, dim)` returns 1 if the vectors vec1 and vec2 (each of size dim) are identical and zero otherwise. The procedure accepts real or complex vec1 or vec2.

Type:

INTEGER FUNCTION

Arguments:

vec1 is REAL(KIND=rKind), DIMENSION(:) or COMPLEX(KIND=rKind), DIMENSION(:) and is INTENT(IN)
vec2 is REAL(KIND=rKind), DIMENSION(:) or COMPLEX(KIND=rKind), DIMENSION(:) is INTENT(IN), and must match the type of vec1
dim is INTEGER and INTENT(IN)

B.9.1.4 TraceMatmul

Description:

`TraceMatmul(A,B)` returns the trace of the matrix resulting from matrix-matrix multiplication of A and B, and
`TraceMatmul(res,A,B)` computes the trace of the matrix resulting from matrix-matrix multiplication of A and B, and stores this in res. The procedure accepts real or complex A or B.

Types:

REAL(KIND=rKind) or COMPLEX(KIND=rKind) FUNCTION
SUBROUTINE

Syntaxes:

result=TraceMatmul(A,B)

CALL TraceMatmul(res,A,B)

Arguments:

A is REAL(KIND=rKind), DIMENSION(:,:) or COMPLEX(KIND=rKind), DIMENSION(:,:) and is INTENT(IN)

B is REAL(KIND=rKind), DIMENSION(:,:) or COMPLEX(KIND=rKind), DIMENSION(:,:) and is INTENT(IN)

res is REAL(KIND=rKind) if both A and B are REAL(KIND=rKind), and COMPLEX(KIND=rKind) otherwise.

B.9.1.5 Factorial

Description:

`Factorial(n)` returns the factorial of the integer n.

Type:

INTEGER FUNCTION

Arguments:

n is INTEGER and INTENT(IN)

B.9.1.6 BinomialCoef

Description:

`BinomialCoef(n,m)` returns the binomial coefficient $\begin{pmatrix} \mathrm{n} \\ \mathrm{m} \end{pmatrix}$.

Type:

REAL(KIND=rKind) FUNCTION

Arguments:

n, m are INTEGER and INTENT(IN)

B.9.1.7 AllocateGamLam

Description:

`AllocateGamLam(Gammas,Lambdas,chi)` allocates the lists of local tensors Gammas and Lambdas with entanglement cutoff chi and local dimension `localSize`.

Type:

SUBROUTINE

Arguments:

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

chi is INTEGER and of INTENT(IN)

B.9.1.8 CopyGamLam

Description:

`CopyGamLam(GammasCopy, LambdasCopy, GammasOrig, LambdasOrig)` copies the values from GammasOrig to GammasCopy, and from LamdasOrig to LambdasCopy.

Type:

SUBROUTINE

Arguments:

GammasCopy, GammasOrig are TYPE(tensor), DIMENSION(:), POINTER

LambdasCopy, LambdasOrig are TYPE(vector), DIMENSION(:), POINTER

B.9.1.9 DeallocateGamLam

Description:

`DeallocateGamLam(Gammas, Lambdas)` deallocates the lists of local tensors Gammas and Lambdas.

Type:

SUBROUTINE

Arguments:

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

B.9.1.10 AllocateLabel

Description:

`AllocateLabel(LabelLeft,LabelRight,chi)` allocates the lists of number conserving vectors LabelLeft and LabelRight with entanglement cutoff chi. See Sec. B.3.4.3 for more details on these objects.

Type:

SUBROUTINE

Arguments:

LabelLeft, LabelRight are TYPE(vectorInt), DIMENSION(:), POINTER

chi is INTEGER and of INTENT(IN)

B.9.1.11 CopyLabel

Description:

`CopyLabel(LablCopy, LabRCopy, LabLOrig, LabROrig)` copies the list of number conserving vectors from LabROrig to LabRCopy, and from LabLOrig to LabLCopy.

Type:

SUBROUTINE

Arguments:

LabLCopy, LabRCopy, LabLOrig, LabROrig are TYPE(vectorInt), DIMENSION(:), POINTER

B.9.1.12 DeallocateLabel

Description:

`DeallocateLabel(LabelLeft, LabelRight)` deallocates the lists of number conserving vectors LabelLeft and LabelRight.

Type:

SUBROUTINE

Arguments:

LabelLeft, LabelRight are TYPE(vectorInt), DIMENSION(:), POINTER

B.9.1.13 AllocateIndexLR

Description:

`AllocateIndexLR(indL, indR, BlockSize)` allocates the lists of indices for the Block diagonal Theta used in number conserving codes. BlockSize gives the size of each block in the Block diagonal Theta. See Sec. B.3.4.3 for more details.

Type:

SUBROUTINE

Arguments:

indL, indR are TYPE(matrixInt), DIMENSION(:), POINTER

BlockSize is an INTEGER, DIMENSION(:,:) and is INTENT(IN)

B.9.1.14 DeallocateIndexLR

Description:

`DeallocateIndexLR(indL, indR)` deallocates the lists of indices for the Block diagonal Theta used in number conserving codes. See Sec. B.3.4.3 for more details.

Type:

SUBROUTINE

Arguments:

indL, indR are TYPE(matrixInt), DIMENSION(:), POINTER

B.9.1.15 AllocateBlockTheta

Description:

`AllocateBlockTheta(BlockTheta, BlockSize)` allocates the Block diagonal Theta used in number conserving codes. BlockSize gives the size of each block in the Block diagonal Theta. See Sec. B.3.4.3 for more details.

Type:

SUBROUTINE

Arguments:

BlockTheta is TYPE(matrix), DIMENSION(:), POINTER

BlockSize is INTEGER, DIMENSION(:,:) and is INTENT(IN)

B.9.1.16 DeallocateBlockTheta

Description:

`DeallocateBlockTheta(BlockTheta)` deallocates the Block diagonal Theta used in number conserving codes. See Sec. B.3.4.3 for more details.

Type:

SUBROUTINE

Arguments:

BlockTheta is TYPE(matrix), DIMENSION(:), POINTER

B.9.1.17 AllocateUSV

Description:

`AllocateUSV(US, SS, VS, BlockSize)` allocates the SVD arrays US, SS, and VS used in number conserving codes. BlockSize gives the size of each block in the Block diagonal Theta. See Sec. B.3.4.3 for more details.

Type:

SUBROUTINE

Arguments:

US, VS are TYPE(matrix), DIMENSION(:), POINTER

SS is TYPE(vector), DIMENSION(:), POINTER

BlockSize is an INTEGER, DIMENSION(:,:) and is INTENT(IN)

B.9.1.18 DeallocateUSV

Description:

`DeallocateUSV(US, SS, VS)` deallocates the SVD arrays US, SS, and VS used in number conserving codes. See Sec. B.3.4.3 for more details.

Type:

SUBROUTINE

Arguments:

US, VS are TYPE(matrix), DIMENSION(:), POINTER

SS is TYPE(vector), DIMENSION(:), POINTER

B.9.1.19 AllocateSSflat

Description:

`AllocateSSflat(ssflat, BlockSize)` allocates the array ssflat, which is used to combine all of the singular values from the individual blocks of the block diagonal Theta into one long array. It is used in number conserving codes. BlockSize gives the size of each block in the Block diagonal Theta. See Sec. B.3.4.3 for more details.

Type:

SUBROUTINE

Arguments:

ssflat is TYPE(vector)

BlockSize is an INTEGER, DIMENSION(:,:) and is INTENT(IN)

B.9.1.20 AllocateOps

Description:

`AllocateOps(Ops,numops,opsize)` allocates a numops length list of opsize×opsize TYPE(matrix) named Ops.

Type:

SUBROUTINE

Arguments:

Ops is TYPE(matrix), DIMENSION(:), POINTER

numops, opsize are INTEGER, INTENT(IN)

B.9.1.21 DeallocateOps

Description:

`DeallocateOps(Ops,numops)` deallocates a numops length list of TYPE(matrix) named Ops.

Type:

SUBROUTINE

Arguments:

Ops is TYPE(matrix), DIMENSION(:), POINTER

numops is INTEGER, INTENT(IN)

B.9.1.22 AllocateProp

Description:

`AllocateProp(U)` allocates the propagator TYPE(matrix) U based on the boundary conditions and order of trotter expansion. This routine supersedes the usage of `AllocateOps` in v1.0.

Type:

SUBROUTINE

Arguments:

U is TYPE(matrix), DIMENSION(:), POINTER

B.9.1.23 DellocateProp

Description:

`DeallocateProp(U)` deallocates the propagator TYPE(matrix) U based on the boundary conditions and order of trotter expansion. This routine supersedes the usage of `AllocateOps` in v1.0.

Type:

SUBROUTINE

Arguments:

U is TYPE(matrix), DIMENSION(:), POINTER

B.9.1.24 ConstructPropagators

Description:

`ConstructPropagators(H, U, dtodd, dteven)` creates a `systemSize` length list of second order trotter propagators U from the list of two-site Hamiltonians H with odd site time step dtodd and even site time step dteven. Explicitly, U(i) is $\exp\left[-i*\mathrm{dtodd}*\mathrm{H}_i\right]$ for i odd and $\exp\left[-i*\mathrm{dteven}*\mathrm{H}_i\right]$ for i even. `ConstructPropagators(H, U, dt)` constructs a general propagator based on the boundary conditions and order of trotter expansion, see Sec. B.3.5 for details. The old syntax is being kept only for consistency with v1.0 and is pending deletion in the next version.

Type:

SUBROUTINE

Syntaxes:

CALL ConstructPropagators(H, U, dtodd, dteven)

CALL ConstructPropagators(H, U, dt)

Arguments:

H, U are TYPE(matrix), DIMENSION(:), POINTER

dtodd, dteven, dt are COMPLEX(KIND=rKind), INTENT(IN)

B.9.2 Hamiltonian_tools_module Procedure Listings

Contents:

`SUBROUTINE ProductStateMPD`

`SUBROUTINE ProductStateLabels`

`SUBROUTINE AllStates`

`SUBROUTINE onsiteStateListIdof`

```
RECURSIVE SUBROUTINE onsiteIdofInner
SUBROUTINE InitialSetNC
SUBROUTINE SetupLogFac
FUNCTION LogTriCoef
FUNCTION TriTest
FUNCTION IntTest
FUNCTION MTest
FUNCTION tIndTJ
FUNCTION tIndSJ
FUNCTION ThreeJ
FUNCTION Clebsch
FUNCTION SixJ
FUNCTION NineJ
INTERFACE HamiOneSite
MODULE PROCEDURE HamiOneSite_r, HamiOneSite_c
INTERFACE HamiLeft
MODULE PROCEDURE HamiLeft_r, HamiLeft_c
INTERFACE HamiRight
MODULE PROCEDURE HamiRight_r, HamiRight_c
```

B.9.2.1 ProductStateMPD

Description:

`ProductStateMPD(Gammas, Lambdas, carray)` constructs the Vidal decomposition of a product state whose coefficients are stored in carray, and then imprints this decomposition on the lists of local tensors Gammas and Lambdas as in Eqs. (B.59) and (B.60). The (i, l) element of carray is the coefficient of the i^{th} on-site state located at site l.

Type:

SUBROUTINE

Arguments:

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

carray is COMPLEX(KIND=rKind), Dimension(:,:) and is INTENT(IN)

B.9.2.2 ProductStateLabels

Description:

`ProductStateLabels` initializes the lists of number conserving vectors based on the MPS stored in carray. This routine should be used in conjunction with `ProductStateMPD` (Sec. B.9.2.1) when the number conserving routines are to be used. intDegFree is an OPTIONAL argument that specifies the presence of internal degrees of freedom.

Type:

SUBROUTINE

Syntaxes:

CALL ProductStateLabels(LabelLeft, LabelRight, carray)

CALL ProductStateLabels(LabelLeft, LabelRight, carray, intDegFree)

Arguments:

LabelLeft, LabelRight are TYPE(vectorInt), DIMENSION(:), POINTER

carray is COMPLEX(KIND=rKind), Dimension(:,:) and is INTENT(IN)

intDegFree is an INTEGER, INTENT(IN) and is OPTIONAL

B.9.2.3 AllStates

Description:

`AllStates(Gammas, Lambdas)` initializes the local tensor lists Gammas and Lambdas as a product state that contains all possible states in the same amount, see Eqs. (B.64) and (B.65). It is used as the inital state for imaginary time propagation in the absence of symmetries.

Type:

SUBROUTINE

Arguments:

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

B.9.2.4 onsiteStateListIdof

Description:

`onsiteStateListIdof(list, idofSize)` indexes the Fock space with up to `maxFilling` particles per site and idofSize internal states per particle and stores the index and number in each component in the variable list. This routine calls `onsiteStateListIdofInner` below. Both routines are described in detail in Sec. B.3.8.2.

Type:

SUBROUTINE

Arguments:

list is COMPLEX(KIND=rKind), DIMENSION(:,:), INTENT(OUT) idofSize is INTEGER, INTENT(INOUT)

B.9.2.5 onsiteIdofInner

Description:

`onsiteIdofInner(list, nmax, counter, m, n, idofSize)` is a recurisve routine that indexes the Fock space with up to nmax particles per site and idofSize internal states per particle and stores the indexing in the variable list. The routine is described in detail in Sec. B.3.8.2.

Type:

RECURSIVE SUBROUTINE

Arguments:

list is COMPLEX(KIND=rKind), DIMENSION(:,:), INTENT(INOUT)

nmax, m, counter are INTEGER, INTENT(INOUT)

n is INTEGER, DIMENSION(:), INTENT(INOUT)

idofSize is INTEGER, INTENT(INOUT)

B.9.2.6 InitialSetNC

Description:

`InitialSetNC` initializes the local tensor lists Gammas and Lambdas and lists of number conserving vectors LabelLeft and LabelRight as the "wedding cake" product state that is an eigenstate of total number, see Sec. B.3.3.2. It is used as the inital state for imaginary time propagation when number is conserved. intDegFree is an OPTIONAL argument that specifies the presence of internal degrees of freedom.

Type:

SUBROUTINE

Syntaxes:

CALL InitialSetNC(Gammas, Lambdas, LabelLeft, LabelRight)

CALL InitialSetNC(Gammas, Lambdas, LabelLeft, LabelRight, intDegFree)

Arguments:

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

LabelLeft, LabelRight are TYPE(vectorInt), DIMENSION(:), POINTER

intDegFree is an INTEGER, INTENT(IN) and is OPTIONAL

B.9.2.7 SetupLogFac

Description:

`SetupLogFac()` initializes the global vector `lFac` such that `lFac(n)`$=\log\left[(n-1)!\right]$ for $1 \leq n \leq 200$. This routine should be run once before any of the vector coupling coefficient routines are run.

Type:

SUBROUTINE

Arguments:

None

B.9.2.8 LogTriCoef

Description:

`LogTriCoef(X1,X2,X3)` returns the logarithm of the triangle coefficient $\triangle$(X1,X2,X3) defined in Eq. (B.183). `SetupLogFac()` should be run before using this routine.

Type:

REAL(KIND=rKind) FUNCTION

Arguments:

X1,X2,X3 are REAL(KIND=rKind) and INTENT(IN)

B.9.2.9 TriTest

Description:

`TriTest(J1,J2,J3)` returns 1 if the triad J1, J2, J3 satisfies the triangle inequality $|\mathrm{J1}-\mathrm{J2}|\leq \mathrm{J3}\leq \mathrm{J1}+\mathrm{J2}$ and zero if it doesn't.

Type:

REAL(KIND=rKind) FUNCTION

Arguments:

J1,J2,J3 are REAL(KIND=rKind) and INTENT(IN)

B.9.2.10 IntTest

Description:

`IntTest(J1,J2,J3)` returns 1 if J1+J2+J3 is integer and zero if it isn't. We define the real number x as "integer" if it satisfies $\lfloor x \rfloor = \lceil x \rceil$.

Type:

REAL(KIND=rKind) FUNCTION

Arguments:

J1,J2,J3 are REAL(KIND=rKind) and INTENT(IN)

B.9.2.11 MTest

Description:

`MTest(M1,J1)` returns 1 if -J1≤M1≤J1 and M1+J1 is integer, and zero otherwise. We define the real number x as "integer" if it satisfies $\lfloor x \rfloor = \lceil x \rceil$.

Type:

REAL(KIND=rKind) FUNCTION

Arguments:

M1, J1 are REAL(KIND=rKind) and INTENT(IN)

B.9.2.12 tIndTJ

Description:

`tIndTJ(X1,X2,X3,Y1,Y2,Y3)` returns the logarithm of the coefficient preceding the summation in the Racah formula for the Wigner 3-j coefficient, Eq. (B.182). `SetupLogFac()` should be run before using this routine.

Type:

REAL(KIND=rKind) FUNCTION

Arguments:

X1,X2,X3,Y1,Y2,Y3 are REAL(KIND=rKind) and INTENT(IN)

B.9.2.13 tIndSJ

Description:

`tIndSJ(X1,X2,X3,Y1,Y2,Y3)` returns the logarithm of the coefficient preceding the summation in the Racah formula for the Wigner 6-j coefficient, Eq. (B.191). `SetupLogFac()` should be run before using this routine.

Type:

REAL(KIND=rKind) FUNCTION

Arguments:

X1,X2,X3,Y1,Y2,Y3 are REAL(KIND=rKind) and INTENT(IN)

B.9.2.14 ThreeJ

Description:

`ThreeJ(J1D,M1D,J2D,M2D,J3D,M3D)` calculates the Wigner 3-j coefficient $\begin{pmatrix} \mathrm{J1D}/2 & \mathrm{J2D}/2 & \mathrm{J3D}/2 \\ \mathrm{M1D}/2 & \mathrm{M2D}/2 & \mathrm{M3D}/2 \end{pmatrix}$ using the Racah formula, Eq. (B.182). Note that the arguments of this procedure are integers twice the value of the inputs to the 3-j symbol. `SetupLogFac()` should be run before using this routine.

Type:

REAL(KIND=rKind) FUNCTION

Arguments:

J1D,M1D,J2D,M2D,J3D,M3D are INTEGER and INTENT(IN)

B.9.2.15 Clebsch

Description:

`Clebsch(J1D,M1D,J2D,M2D,JD,MD)` calculates the Clebsch-Gordan coefficient $\langle \mathrm{J1D}/2, \mathrm{M1D}/2, \mathrm{J2D}/2, \mathrm{M2D}/2 | \mathrm{JD}/2, \mathrm{MD}/2 \rangle$ using the corresponding 3-j as in Eq. (B.177). Note that the arguments of this procedure are integers twice the value of the inputs to the Clebsch-Gordan coefficient. `SetupLogFac()` should be run before using this routine.

Type:

REAL(KIND=rKind) FUNCTION

Arguments:

J1D,M1D,J2D,M2D,JD,MD are INTEGER and INTENT(IN)

Return Value:

$\langle \mathrm{J1D}/2, \mathrm{M1D}/2, \mathrm{J2D}/2, \mathrm{M2D}/2 | \mathrm{JD}/2, \mathrm{MD}/2 \rangle$

B.9.2.16 SixJ

Description:

`SixJ(J11D,J21D,J12D,J22D,J13D,J23D)` calculates the Wigner 6-j coefficient

$\begin{Bmatrix} \mathrm{J11D}/2 & \mathrm{J12D}/2 & \mathrm{J13D}/2 \\ \mathrm{J21D}/2 & \mathrm{J22D}/2 & \mathrm{J23D}/2 \end{Bmatrix}$ using the Racah formula, Eq. (B.191). Note that the arguments of this procedure are integers twice the value of the inputs to the 6-j symbol. `SetupLogFac()` should be run before using this routine.

Type:

REAL(KIND=rKind) FUNCTION

Arguments:

J11D,J21D,J12D,J22D,J13D,J23D are INTEGER and INTENT(IN)

B.9.2.17 NineJ

Description:

`NineJ(J11D,J21D,J31D,J12D,J22D,J32D,J13D,J23D,J33D)` calculates the Wigner 9-j coefficient
$\begin{Bmatrix} \mathrm{J11D}/2 & \mathrm{J12D}/2 & \mathrm{J13D}/2 \\ \mathrm{J21D}/2 & \mathrm{J22D}/2 & \mathrm{J23D}/2 \\ \mathrm{J31D}/2 & \mathrm{J32D}/2 & \mathrm{J33D}/2 \end{Bmatrix}$ using the contraction formula, Eq. (B.197). Note that the arguments of this procedure are integers twice the value of the inputs to the 9-j symbol. `SetupLogFac()` should be run before using this routine.

Type:

REAL(KIND=rKind) FUNCTION

Arguments:

J11D,J21D,J31D,J12D,J22D,J32D,J13D,J23D,J33D are INTEGER and INTENT(IN)

B.9.2.18 HamiOneSite

Description:

`HamiOneSite(Op)` returns $\frac{1}{2}\left[\hat{1}\otimes\hat{\mathrm{Op}}+\hat{\mathrm{Op}}\otimes\hat{1}\right]$, used to add a one-site term to a Hamiltonian. The procedure accepts TYPE(matrix) or TYPE(matrixReal) Op, and returns COMPLEX(KIND=rKind) or REAL(KIND=rKind) values accordingly.

Type:

COMPLEX(KIND=rKind) or REAL(KIND=rKind) FUNCTION

Arguments:

Op is TYPE(matrix), INTENT(IN) or TYPE(matrixReal), INTENT(IN)

B.9.2.19 HamiLeft

Description:

`HamiLeft(Op)` returns $\frac{1}{2}\hat{\mathrm{Op}}\otimes\hat{1}$, used to add a one-site term to the first site of an open-boundary Hamiltonian. The procedure accepts TYPE(matrix) or TYPE(matrixReal) Op, and returns COMPLEX(KIND=rKind) or REAL(KIND=rKind) values accordingly.

Type:

COMPLEX(KIND=rKind) or REAL(KIND=rKind) FUNCTION

Arguments:

Op is TYPE(matrix), INTENT(IN) or TYPE(matrixReal), INTENT(IN)

B.9.2.20 HamiRight

Description:

`HamiRight(Op)` returns $\frac{1}{2}\hat{1}\otimes\hat{\mathrm{Op}}$, used to add a one-site term to the last site of an open-boundary Hamiltonian. The procedure accepts TYPE(matrix) or TYPE(matrixReal) Op, and returns COMPLEX(KIND=rKind) or REAL(KIND=rKind) values accordingly.

Type:

COMPLEX(KIND=rKind) or REAL(KIND=rKind) FUNCTION

Arguments:

Op is TYPE(matrix), INTENT(IN) or TYPE(matrixReal), INTENT(IN)

B.9.3 Bose_hubbard_module Procedure Listings

Contents:

```
FUNCTION BoseHubbardLocalDim
SUBROUTINE CreateFieldOps
SUBROUTINE DestroyFieldOps
INTERFACE HamiltonianBoseHubbard
MODULE PROCEDURE HamiltonianBoseHubbardScalar, HamiltonianBoseHubbardUVector,
HamiltonianBoseHubbardJVector, HamiltonianBoseHubbardJUVector
SUBROUTINE SetupBHName
```

B.9.3.1 BoseHubbardLocalDim

Description:

`BoseHubbardLocalDim()` returns the local dimension of a spinless Bose-Hubbard system truncated at `maxFilling` particles per site. It is simply `maxFilling`+1, the extra

1 accounting for the vacuum state.

Type:

INTEGER FUNCTION

Arguments:

None

B.9.3.2 CreateFieldOps

Description:

`CreateFieldOps()` allocates and defines the suite of operators that define and characterize the Bose-Hubbard Hamiltonian, Eq. (B.151). These include the one-site operators a_op, which destroys a particle on-site, one_op, the single site unity operator, and PBphase_op, the Pegg-Barnett phase operator. Also included is the two-site operator t_op, the tunneling operator. All of these operators are discussed in Sec. B.8.2.

Type:

SUBROUTINE

Arguments:

None

B.9.3.3 DestroyFieldOps

Description:

`DestroyFieldOps()` deallocates the suite of operators that define and characterize the Bose-Hubbard Hamiltonian, Eq. (B.151).

Type:

SUBROUTINE

Arguments:

None

B.9.3.4 HamiltonianBoseHubbard

Description:

`HamiltonianBoseHubbard` creates the Bose-Hubbard Hamiltonian, Eq. (B.151), characterized by tunneling energy jTunn, on-site interaction energy U0, chemical potential mu0, and nearest-neighbor interaction energy V0 in TEBD form, and stores this in the list H. The OPTIONAL argument extPot is a site-indexed array of the onsite energies ϵ_i as defined in Eq. (B.151), allowing for the additional of an arbitrary external potential. The arguments jTunn and U0 may be either scalars or site-indexed vectors.

Type:

SUBROUTINE

Syntaxes:

CALL HamiltonianBoseHubbard(H, jTunn, U0, mu0, V0)

CALL HamiltonianBoseHubbard(H, jTunn, U0, mu0, V0, extPot)

Arguments:

mu0, V0 are REAL(KIND=rKind) and INTENT(IN)

jTunn, U0 may be either REAL(KIND=rKind), INTENT(IN) or REAL(KIND=rKind), DIMENSION(:), INTENT(IN) extPot is REAL(KIND=rKind), DIMENSION(:), INTENT(IN), OPTIONAL

H is TYPE(matrix), DIMENSION(:), POINTER

B.9.3.5 SetupBHName

Description:

`SetupBHName(baseName,diRectory)` creates a file name (in the directory diRectory) containing the Bose-Hubbard parameters defined when called, and stores it in the character string baseName.

Type:

SUBROUTINE

Arguments:

baseName is CHARACTER(len=*), INTENT(INOUT)

diRectory is CHARACTER(len=*), INTENT(IN)

B.9.4 Fermi_hubbard_module Procedure Listings

Contents:

```
FUNCTION spinSFermiLocalDim
SUBROUTINE CreateFermiSOps
SUBROUTINE DestroyFermiSOps
INTERFACE HamiltonianHubbard
SUBROUTINE SetupFHName
```

B.9.4.1 spinSFermiLocalDim

Description:

`spinSFermiLocalDim()` returns the local dimension of a spin-S Hubbard system truncated at `maxFilling` particles per site.

Type:

INTEGER FUNCTION

Arguments:

None

B.9.4.2 CreateFermiSOps

Description:

`CreateFermiSOps()` allocates and defines the suite of operators that define and characterize the Hubbard Hamiltonian, Eq. (B.152). These include the list of one-site operators a_opS(:), which destroys a particle on-site in a particular spin component, one_op, the single site unity operator, Sx_opS, Sy_opS, Sz_opS, and Ssq_opS, the spin operators, and FermiPhase_op (Sec. B.8.2.12), which is used in constructing two-site correlation functions with Fermi phases (Sec. B.3.6.3). Also included is the two-site operator ttot_opS, the tunneling operator summed over all spin components. All of these operators are discussed in Sec. B.8.2.

Type:

SUBROUTINE

Arguments:

None

B.9.4.3 DestroyFermiSOps

Description:

`DestroyFermiSOps()` deallocates the suite of operators that define and characterize the Hubbard Hamiltonian, Eq. (B.152).

Type:

SUBROUTINE

Arguments:

None

B.9.4.4 HamiltonianHubbard

Description:

`HamiltonianHubbard` creates the Hubbard Hamiltonian, Eq. (B.151), characterized by tunneling energy jTunn, on-site interaction energy U0, chemical potential mu0, and nearest-neighbor interaction energy V0 in TEBD form, and stores this in the list H. The OPTIONAL argument extPot is a site-indexed array of the onsite energies ϵ_i as defined in Eq. (B.152), allowing for the additional of an arbitrary external potential.

Type:

SUBROUTINE

Syntaxes:

CALL HamiltonianHubbard(H, jTunn, U0, mu0)

CALL HamiltonianHubbard(H, jTunn, U0, mu0, V0=V0)

CALL HamiltonianHubbard(H, jTunn, U0, mu0, extPot=extPot)

CALL HamiltonianHubbard(H, jTunn, U0, mu0, V0=V0, extPot=extPot)

Arguments:

jTunn, U0, mu0 are REAL(KIND=rKind) and INTENT(IN)

V0 is REAL(KIND=rKind), INTENT(IN), OPTIONAL

extPot is REAL(KIND=rKind), DIMENSION(:), INTENT(IN), OPTIONAL

H is TYPE(matrix), DIMENSION(:), POINTER

B.9.4.5 SetupFHName

Description:

`SetupFHName(baseName,diRectory)` creates a file name (in the directory diRectory) containing the Hubbard parameters defined when called, and stores it in the character string baseName.

Type:

SUBROUTINE

Arguments:

baseName is CHARACTER(len=*), INTENT(INOUT)

diRectory is CHARACTER(len=*), INTENT(IN)

B.9.5 Heisenberg_module Procedure Listings

Contents:

```
FUNCTION HeisenbergLocalDim
SUBROUTINE CreateHeisenbergOps
SUBROUTINE DestroyHeisenbergOps
SUBROUTINE HamiltonianHeisenberg
SUBROUTINE SetupHeisenbergName
```

B.9.5.1 HeisenbergLocalDim

Description:

`HeisenbergLocalDim()` returns the local dimension of a spin=`spin` Heisenberg chain truncated at `maxFilling`=1 particles per site. It is simply `spinSize`+1, the extra 1 accounting for the vacuum state.

Type:

INTEGER FUNCTION

Arguments:

None

B.9.5.2 CreateHeisenbergOps

Description:

`CreateHeisenbergOps()` allocates and defines the suite of operators that define and characterize the Heisenberg spin chain, Eq. (B.150). These include the list of destruction operators indexed by spin component a_opS(:), the unity operator one_op, the spin operators Sx_opS, Sy_opS, Sz_opS, and Ssq_opS, and the total number operator ntot_opS, all of which are discussed in Sec. B.8.2.

Type:

SUBROUTINE

Arguments:

None

B.9.5.3 DestroyHeisenbergOps

Description:

`DestroyHeisenbergOps()` deallocates the suite of operators that define and characterize the Heisenberg spin chain, Eq. (B.150).

Type:

SUBROUTINE

Arguments:

None

B.9.5.4 HamiltonianHeisenberg

Description:

`HamiltonianHeisenberg` creates the Heisenberg spin chain Hamiltonian, Eq. (B.150), characterized by spin couplings Jx, Jy, and Jz in TEBD form, and stores this in the list H. The OPTIONAL argument magField adds a magnetic field, defined as h in Eq. (B.150).

Type:

SUBROUTINE

Syntaxes:

CALL HamiltonianHeisenberg(H , Jx, Jy, Jz)

CALL HamiltonianHeisenberg(H , Jx, Jy, Jz, magField)

Arguments:

Jx, Jy, Jz are REAL(KIND=rKind) and INTENT(IN)

magField is REAL(KIND=rKind), INTENT(IN), OPTIONAL

H is TYPE(matrix), DIMENSION(:), POINTER

B.9.5.5 SetupHeisenbergName

Description:

`SetupHeisenbergName(baseName,diRectory)` creates a file name (in the directory diRectory) containing the Heisenberg parameters defined when called, and stores it in the character string baseName.

Type:

SUBROUTINE

Arguments:

baseName is CHARACTER(len=*), INTENT(INOUT)

diRectory is CHARACTER(len=*), INTENT(IN)

B.9.6 spinS_module Procedure Listings

Contents:

```
FUNCTION spinSLocalDim
SUBROUTINE CreateSpinSops
SUBROUTINE DestroySpinSops
SUBROUTINE HamiltonianSpinOne
SUBROUTINE HamiltonianSpinS
SUBROUTINE SetupBHSpinOneName
```

B.9.6.1 SpinSLocalDim

Description:

`SpinSLocalDim()` returns the local dimension of a spin=`spin` system with at most `maxFilling` particles per site using Eq. (B.198).

Type:

INTEGER FUNCTION

Arguments:

None

B.9.6.2 CreateSpinSops

Description:

`CreateSpinSops()` allocates and initializes the operators which define and characterize the spin-s Bose-Hubbard Hamiltonian, Eq. (B.153). These include the list of destruction operators indexed by spin component a_opS(:), the unity operator one_op, the spin operators Sx_opS, Sy_opS, Sz_opS, and Ssq_opS, the Zeeman operator VB_opS, the total number operator ntot_opS, and and the total tunneling operator ttot_opS, all of which are discussed in Sec. B.8.2.

Type:

SUBROUTINE

Arguments:

None

B.9.6.3 DestroySpinSops

Description:

`DestroySpinSops()` deallocates the operators which define and characterize the spin-s Bose-Hubbard Hamiltonian, Eq. (B.153).

Type:

SUBROUTINE

Arguments:

None

B.9.6.4 HamiltonianSpinOne

Description:

`HamiltonianSpinOne(H , J, U0, U2, mu0, VB)` creates the spin-1 Bose-Hubbard Hamiltonian, Eq. (B.156).

Type:

SUBROUTINE

Arguments:

H is TYPE(matrix), DIMENSION(:), POINTER

J, U0, U2, mu0, VB are REAL(KIND=rKind) and INTENT(IN)

B.9.6.5 HamiltonianSpinS

Description:

`HamiltonianSpinS(H , J, gS, mu0, VB)` creates the spin-s Bose-Hubbard Hamiltonian, Eq. (B.153), where gS is a vector of the g_S.

Type:

SUBROUTINE

Arguments:

H is TYPE(matrix), DIMENSION(:), POINTER

J, mu0, VB are REAL(KIND=rKind) and INTENT(IN)

gS is REAL(KIND=rKind), DIMENSION(:) and INTENT(IN)

B.9.6.6 SetupBHSpinOneName

Description:

`SetupBHSpinOneName(baseName,diRectory)` creates a file name (in the directory diRectory) containing the spin-1 Bose-Hubbard parameters defined when called, and stores it in the character string baseName.

Type:

SUBROUTINE

Arguments:

baseName is CHARACTER(len=*), INTENT(INOUT)

diRectory is CHARACTER(len=*), INTENT(IN)

B.9.7 rotation_module Procedure Listings

Contents:

`SUBROUTINE AllocateDsyev`

`SUBROUTINE DeAllocateDsyev`

`FUNCTION rotationLocalDimMzero`

`SUBROUTINE DiagDsyev`

`SUBROUTINE CreateRotationopsMzero`

`FUNCTION rotationLocalDim`

`SUBROUTINE CreateRotationops`

`SUBROUTINE HamiltonianRotationTI`

`SUBROUTINE HamiltonianRotationTD`

`SUBROUTINE DestroyRotationops`

`SUBROUTINE SetupRotName`

B.9.7.1 AllocateDsyev

Description:

`AllocateDsyev(dimen)` allocates the variables needed to diagonalize the rotational + DC Hamiltonians on the extended (highest field-free state=J_{cut} manifold) using the LAPACK routine DSYEV. These variables are discussed in Sec. B.8.5. dimen is the dimension of the Hamiltonian.

Type:

SUBROUTINE

Arguments:

dimen is INTEGER, INTENT(IN)

B.9.7.2 DeallocateDsyev

Description:

`DeallocateDsyev()` deallocates the variables needed to diagonalize the rotational + DC Hamiltonians on the extended (highest field-free state=`Jcut` manifold) using the

LAPACK routine DSYEV.

Type:

SUBROUTINE

Arguments:

None

B.9.7.3 rotationLocalDimMzero

Description:

`rotationLocalDimMzero()` returns the local dimension of a system with highest allowed number of rotational quanta `rotLevel` and at most `maxFilling` particles per site under the constraint that $M = 0$.

Type:

INTEGER FUNCTION

Arguments:

None

B.9.7.4 DiagDsyev

Description:

`DiagDsyev(dimen, Mat,OutMat,OutEig,dipoles)` diagonalizes the dimen×dimen rotational + DC Hamiltonians Mat on the extended (highest field-free state=J_{cut} manifold) using the LAPACK routine DSYEV under the constraint that $M = 0$, and returns the full set of eigenvectors as OutMat, the eigenvalues as OutEig, and the induced dipoles for the lowest `rotLevel` levels as dipoles.

Type:

SUBROUTINE

Arguments:

dimen is INTEGER, INTENT(IN)

Mat is REAL(KIND=rKind), INTENT(INOUT)

OutMat and dipoles are REAL(KIND=rKind), DIMENSION(:,:), INTENT(OUT)

OutEig is REAL(KIND=rKind), DIMENSION(:), INTENT(OUT)

B.9.7.5 CreateRotationopsMzero

Description:

`CreateRotationopsMzero()` allocates and initializes the operators which define and characterize the Molecular Hubbard Hamiltonian, Eq. (B.163), under the constraint that $M = 0$. These include the list of destruction operators indexed by rotational component a_opS(:), the unity operator one_op, the DC and AC electric field operators EDC_opR and EAC_opR, the dipole-dipole operator dipdip_opR, and and the total tunneling operator ttot_opR, all of which are described in sections B.3.7.5 and B.8.2.

Type:

SUBROUTINE

Arguments:

None

B.9.7.6 rotationLocalDim

Description:

`rotationLocalDim()` returns the local dimension of a system with highest allowed number of rotational quanta `rotLavel` and at most `maxFilling` particles per site.

Type:

INTEGER FUNCTION

Arguments:

None

B.9.7.7 CreateRotationops

Description:

`CreateRotationops()` allocates and initializes the operators which define and characterize the Molecular Hubbard Hamiltonian, Eq. (B.163). These include the list of destruction operators indexed by rotational component a_opS(:), the unity operator one_op, the DC and AC electric field operators EDC_opR and EAC_opR, the dipole-dipole operator dipdip_opR, and and the total tunneling operator ttot_opR, all of which are described in sections B.3.7.5 and B.8.2. The tunneling and dipole-dipole operators are not currently supported.

Type:

SUBROUTINE

Arguments:

None

B.9.7.8 HamiltonianRotationTI

Description:

`HamiltonianRotationTI(H)` creates the time-independent Molecular Hubbard Hamiltonian, Eq. (B.163) with all $\Omega_{JM}=0$. It is used for finding the static ground state via imaginary time propagation.

Type:

SUBROUTINE

Arguments:

H is TYPE(matrix), DIMENSION(:), POINTER

B.9.7.9 HamiltonianRotationTD

Description:

`HamiltonianRotationTD(H,time)` creates the time-dependent Molecular Hubbard Hamiltonian, Eq. (B.163), evaluated at t=time. It is used for real time propagation.

Type:

SUBROUTINE

Arguments:

H is TYPE(matrix), DIMENSION(:), POINTER

time is REAL(KIND=rKind), INTENT(IN)

B.9.7.10 DestroyRotationops

Description:

`DestroyRotationops()` deallocates the operators which define and characterize the

Molecular Hubbard Hamiltonian, Eq. (B.163). Note that this routine is used for both $M = 0$ codes and normal codes.

Type:

SUBROUTINE

Arguments:

None

B.9.7.11 SetupRotName

Description:

`SetupRotName(baseName,diRectory)` creates a file name (in the directory diRectory) containing the Molecular-Hubbard parameters defined when called, and stores it in the character string baseName.

Type:

SUBROUTINE

Arguments:

baseName is CHARACTER(len=*), INTENT(INOUT)

diRectory is CHARACTER(len=*), INTENT(IN)

B.9.8 local_operations_module Procedure Listings

Contents:

```
SUBROUTINE SVDInit
SUBROUTINE SVDFinish
```

```
INTERFACE OneSiteOp
        MODULE PROCEDURE OneSiteOp_r,OneSiteOp_c
SUBROUTINE FormTheta
INTERFACE ThetaOperation
MODULE PROCEDURE ThetaOperation_r, ThetaOperation_c
SUBROUTINE ReshapeTheta
SUBROUTINE SVDTruncation
SUBROUTINE FormLambda1
SUBROUTINE FormGamma1
SUBROUTINE FormGamma2
INTERFACE TwoSiteOp
        MODULE PROCEDURE TwoSiteOp_r,TwoSiteOp_c
SUBROUTINE CanonicalForm
SUBROUTINE SVD
SUBROUTINE SwapTheta
SUBROUTINE Swapping
SUBROUTINE SVDInitNC
SUBROUTINE SVDFinishNC
SUBROUTINE FormThetaNC
INTERFACE ThetaOperationNC
        MODULE PROCEDURE ThetaOperationNC_r,ThetaOperationNC_c
SUBROUTINE RenormThetaNC
SUBROUTINE SwapThetaNC
SUBROUTINE minmaxNLR
SUBROUTINE SizeOfBlocks
SUBROUTINE IndexLeft
SUBROUTINE IndexRight
SUBROUTINE FormBlockTheta
SUBROUTINE SVDNC
SUBROUTINE FlattenSS
SUBROUTINE Ordering
SUBROUTINE JudgePosition
SUBROUTINE UpdateLabelLeft
```

```
SUBROUTINE UpdateLabelRight
SUBROUTINE FormLambdaNC
SUBROUTINE FormGamma1NC
SUBROUTINE FormGamma2NC
INTERFACE TwoSiteOpNC
        MODULE PROCEDURE TwoSiteOpNC_r, TwoSiteOpNC_c
SUBROUTINE SwappingNC
SUBROUTINE SpecialState
```

B.9.8.1 SVDInit

Description:

`SVDInit(chi)` allocates the workspace needed to perform a singular value decomposition via the LAPACK routine ZGESVD.

Type:

SUBROUTINE

Arguments:

chi is INTEGER, INTENT(IN)

B.9.8.2 SVDFinish

Description:

`SVDFinish()` deallocates the workspace needed to perform a singular value decomposition via the LAPACK routine ZGESVD.

Type:

SUBROUTINE

Arguments:

None

B.9.8.3 OneSiteOp

Description:

`OneSiteOp(Op1,Gamma)` performs the one-site operation Op1 on the site whose local tensor is Gamma. The routine updates Gamma as discussed in Sec. B.3.4.1

Type:

SUBROUTINE

Arguments:

Op1 is either REAL(KIND=rKind) or COMPLEX(KIND=rKind), DIMENSION(:,:), INTENT(IN)

Gamma is COMPLEX(KIND=rKind), DIMENSION(:,:,:), INTENT(INOUT)

B.9.8.4 FormTheta

Description:

`FormTheta(Theta, Lambda0, Gamma1, Lambda1, Gamma2, Lambda2)` generates the four-index tensor Θ as defined in Sec. B.3.4.2, Eq. (B.75). For a bipartite splitting at link $l+1$, Lambda0 is $\lambda^{[l]}$, Gamma1 is $\Gamma^{[l]}$, Lambda1 is $\lambda^{[l+1]}$, Gamma2 is $\Gamma^{[l+1]}$, and Lambda2 is $\lambda^{[l+1]}$.

Type:

SUBROUTINE

Arguments:

Lambda0, Lambda1, Lambda2 are REAL(KIND=rKind), DIMENSION(:), INTENT(IN)

Gamma1, Gamma2 are COMPLEX(KIND=rKind), DIMENSION(:,:,:), INTENT(IN)

Theta is COMPLEX(KIND=rKind), DIMENSION(:,:,:,:), INTENT(OUT)

B.9.8.5 ThetaOperation

Description:

`ThetaOperation(Op2,Theta)` multiplies Θ by the two-site operator Op2 (in the Kronecker representation). This is step (2) of the procedure discussed in Sec. B.3.4.2.

Type:

SUBROUTINE

Arguments:

Op2 is either REAL(KIND=rKind) or COMPLEX(KIND=rKind), DIMENSION(:,:), INTENT(IN)

Theta is COMPLEX(KIND=rKind), DIMENSION(:,:,:,:), INTENT(INOUT)

B.9.8.6 ReshapeTheta

Description:

`ReshapeTheta(Theta,ThetaRS)` renormalizes Θ such that the sum of the squares of all elements is 1, and then reshapes the $\chi \times d \times d \times \chi$ four-tensor into a two-tensor as

$$\Theta'_{(i-1)\chi+\alpha,(j-1)\chi+\beta} = \Theta_{\alpha,i,j,\beta}\,. \tag{B.214}$$

This is step (3) of the procedure discussed in Sec. B.3.4.2.

Type:

SUBROUTINE

Arguments:

Theta is COMPLEX(KIND=rKind), DIMENSION(:,:,:,:), INTENT(IN)

ThetaRS is COMPLEX(KIND=rKind), DIMENSION(:,:), INTENT(OUT)

B.9.8.7 SVDTruncation

Description:

`SVDTruncation(link, MatrixIn, S, U, V)` performs a singular value decomposition on the matrix MatrixIn, storing the χ largest singular values in the vector S and the left and right unitary matrices in U and V, respectively. When MatrixIn is the reshaped Theta from above, this is step (4) of the procedure discussed in Sec. B.3.4.2, and is shown explicitly in Eq. (B.80).

Type:

SUBROUTINE

Arguments:

link is INTEGER, INTENT(IN)

MatrixIn is COMPLEX(KIND=rKind), DIMENSION(:,:), INTENT(INOUT)

S is REAL(KIND=rKind), DIMENSION(:), INTENT(OUT)

U is COMPLEX(KIND=rKind), DIMENSION(:,:), INTENT(OUT)

V is COMPLEX(KIND=rKind), DIMENSION(:,:), INTENT(OUT)

B.9.8.8 FormLambda1

Description:

`FormLambda1(Lambda1, truncerr, S, chi1)` updates the local tensor Lambda1= $\lambda^{[l+1]}$ using the singular values stored in S to account for the action of a two-site operation. The truncation error resulting from discarding the $(d-1)\,\chi$ smallest singular values is returned in truncerr. This is part of step (5) of the procedure discussed in Sec. B.3.4.2, and is shown explicitly in Eq. (B.81).

Type:

SUBROUTINE

Arguments:

S is REAL(KIND=rKind), DIMENSION(:), INTENT(IN)

Lambda1 is REAL(KIND=rKind), DIMENSION(:), INTENT(INOUT)

truncerr is REAL(KIND=rKind), INTENT(OUT)

chi1 is INTEGER, INTENT(IN)

B.9.8.9 FormGamma1

Description:

`FormGamma1(Lambda0,Gamma1,U,chi0,chi1)` updates the local tensor Gamma1= $\Gamma^{[l]}$ using the matrix U obtained from the singular value decomposition to account for the action of a two-site operation. This is part of step (5) of the procedure discussed in Sec. B.3.4.2, and is shown explicitly in Eq. (B.82).

Type:

SUBROUTINE

Arguments:

U is REAL(KIND=rKind), DIMENSION(:,:), INTENT(IN)

Lambda0 is REAL(KIND=rKind), DIMENSION(:), INTENT(IN)

Gamma1 is COMPLEX(KIND=rKind), DIMENSION(:,:,:), INTENT(INOUT)

chi0, chi1 are INTEGER, INTENT(IN)

B.9.8.10 FormGamma2

Description:

`FormGamma2(Gamma2,Lambda2,V,chi1,chi2)` updates the local tensor Gamma2= $\Gamma^{[l+1]}$ using the matrix V obtained from the singular value decomposition to account for the action of a two-site operation. This is part of step (5) of the procedure discussed in Sec. B.3.4.2, and is shown explicitly in Eq. (B.83).

Type:

SUBROUTINE

Arguments:

V is REAL(KIND=rKind), DIMENSION(:,:), INTENT(IN)

Lambda2 is REAL(KIND=rKind), DIMENSION(:), INTENT(IN)

Gamma2 is COMPLEX(KIND=rKind), DIMENSION(:,:,:), INTENT(INOUT)

chi1, chi2 are INTEGER, INTENT(IN)

B.9.8.11 TwoSiteOp

Description:

`TwoSiteOp(link,Op2,Gammas,Lambdas,truncerr)` operates the two-site operator Op2 on the sites separated by link. The truncation error resulting from discarding the $(d-1)\,\chi$ smallest singular values is recorded in truncerr. This procedure performs steps (1)-(5) of the procedure discussed in Sec. B.3.4.2.

Type:

SUBROUTINE

Arguments:

link is INTEGER, INTENT(IN)

Op2 is either REAL(KIND=rKind) or COMPLEX(KIND=rKind), DIMENSION(:,:), INTENT(IN)

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

truncerr is REAL(KIND=rKind), INTENT(OUT)

B.9.8.12 CanonicalForm

Description:

`CanonicalForm(Lambda0,Gamma1,Lambda1,Gamma2,Lambda2)` puts the bipartite splitting at link $l+1$ defined by Lambda1=$\lambda^{[l+1]}$ into canonical form as defined in[14]. The essential idea is that we change a splitting of our tensor network into subsystems A and B into a Schmidt decomposition, i.e.

$$|\Psi\rangle = \sum_{\alpha} |\phi_{\alpha}^{[A]}\rangle|\phi_{\alpha}^{[B]}\rangle \overrightarrow{\scriptstyle\mathrm{CF}} \sum_{\alpha} \lambda_{\alpha}|\Phi_{\alpha}^{[A]}\rangle|\Phi_{\alpha}^{[B]}\rangle\,. \tag{B.215}$$

This is useful when we need to re-orthogonalize the Schmidt bases during imaginary time evolution or after swapping routines. See Sec. B.3.4.4 for explicit details.

Type:

SUBROUTINE

Arguments:

Lambda0, Lambda1, Lambda2 are REAL(KIND=rKind), DIMENSION(:), INTENT(INOUT)

Gamma1, Gamma2 are COMPLEX(KIND=rKind), DIMENSION(:,:,:), INTENT(INOUT)

B.9.8.13 SVD

Description:

`SVD(MatrixIn, U, S, VT)` performs a singular value decomposition on the matrix MatrixIn, storing the singular values in the vector S and the left and (transposed) right unitary matrices in U and VT, respectively.

Type:

SUBROUTINE

Arguments:

MatrixIn is COMPLEX(KIND=rKind), DIMENSION(:,:), INTENT(INOUT)

S is REAL(KIND=rKind), DIMENSION(:), INTENT(OUT)

U is COMPLEX(KIND=rKind), DIMENSION(:,:), INTENT(OUT)

VT is COMPLEX(KIND=rKind), DIMENSION(:,:), INTENT(OUT)

B.9.8.14 SwapTheta

Description:

`SwapTheta(Theta,ThetaSW)` swaps the local indices on the four-tensor Θ and then reshapes this into the two-tensor Θ' as

$$\Theta'_{(j-1)\chi+\alpha,(i-1)\chi+\beta} = \Theta_{\alpha,i,j,\beta}\,. \tag{B.216}$$

This procedure is used in routines for periodic boundary conditions, see Sec. B.3.4.4.

Type:

SUBROUTINE

Arguments:

Theta is COMPLEX(KIND=rKind), DIMENSION(:,:,:,:), INTENT(IN)

ThetaSW is COMPLEX(KIND=rKind), DIMENSION(:,:), INTENT(OUT)

B.9.8.15 Swapping

Description:

`Swapping(link,Gammas,Lambdas)` swaps the local indices on the four-tensor Θ and then puts this new splitting into canonical form. This procedure is used in routines for periodic boundary conditions, see Sec. B.3.4.4.

Type:

SUBROUTINE

Arguments:

link is INTEGER, INTENT(IN)

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

B.9.8.16 SVDInitNC

Description:

`SVDInitNC(k,BlockSize)` allocates the workspace needed to perform singular value decompositions on each block of the block diagonal Θ via the LAPACK routine ZGESVD.

Type:

SUBROUTINE

Arguments:

k is INTEGER, INTENT(IN)

BlockSize is INTEGER, DIMENSION(:,:), INTENT(IN)

B.9.8.17 SVDFinishNC

Description:

`SVDFinishNC()` deallocates the workspace needed to perform singular value decompositions on each block of the block diagonal Θ via the LAPACK routine ZGESVD.

Type:

SUBROUTINE

Arguments:

None

B.9.8.18 FormThetaNC

Description:

`FormThetaNC` generates the four-index tensor Θ=Theta consistent with number conservation as defined in Sec. B.3.4.3. For a bipartite splitting at link $l+1$, Lambda0 is $\lambda^{[l]}$, Gamma1 is $\Gamma^{[l]}$, Lambda1 is $\lambda^{[l+1]}$, Gamma2 is $\Gamma^{[l+1]}$, and Lambda2 is $\lambda^{[l+1]}$.

The OPTIONAL argument intDegFree, which can be any integer value, specifies the presence of internal degrees of freedom. This procedure is step (1) of the number conserving two-site operation discussed in Sec. B.3.4.3.

Type:

SUBROUTINE

Syntaxes:

CALL FormThetaNC(link, Theta, Gammas, Lambdas, LabelLeft, LabelRight)

CALL FormThetaNC(link, Theta, Gammas, Lambdas, LabelLeft, LabelRight, intDegFree)

Arguments:

link is INTEGER, INTENT(IN) Gammas is TYPE(tensor), DIMENSION(:), POINTER
Lambdas is TYPE(vector), DIMENSION(:), POINTER
LabelLeft and LabelRight are TYPE(vectorInt), DIMENSION(:), POINTER
Theta is COMPLEX(KIND=rKind), DIMENSION(:,:,:,:), INTENT(OUT)
intDegFree is INTEGER, INTENT(IN), OPTIONAL

B.9.8.19 ThetaOperationNC

Description:

`ThetaOperationNC` multiplies Θ by the two-site operator Op2 (in the Kronecker representation) consistent with number conservation. The OPTIONAL argument intDegFree, which can be any integer value, specifies the presence of internal degrees of freedom. This procedure is step (2) of the number conserving two-site operation discussed in Sec. B.3.4.3.

Type:

SUBROUTINE

Syntaxes:

CALL ThetaOperationNC(link, Op2, Theta, LabelLeft, LabelRight)

CALL ThetaOperationNC(link, Op2, Theta, LabelLeft, LabelRight, intDegFree)

Arguments:

link is INTEGER, INTENT(IN)

Op2 is either REAL(KIND=rKind) or COMPLEX(KIND=rKind), DIMENSION(:,:), INTENT(IN)

Theta is COMPLEX(KIND=rKind), DIMENSION(:,:,:,:), INTENT(INOUT)

LabelLeft and LabelRight are TYPE(vectorInt), DIMENSION(:), POINTER

intDegFree is INTEGER, INTENT(IN), OPTIONAL

B.9.8.20 RenormThetaNC

Description:

`RenormThetaNC` renormalizes Θ such that the sum of the squares of all elements is 1. The OPTIONAL argument intDegFree, which can be any integer value, specifies the presence of internal degrees of freedom. This procedure is step (3) of the number conserving two-site operation discussed in Sec. B.3.4.3.

Type:

SUBROUTINE

Syntaxes:

CALL RenormThetaNC(link, Theta, LabelLeft, LabelRight)

CALL RenormThetaNC(link, Theta, LabelLeft, LabelRight, intDegFree)

Arguments:

link is INTEGER, INTENT(IN)

Theta is COMPLEX(KIND=rKind), DIMENSION(:,:,:,:), INTENT(INOUT)

LabelLeft and LabelRight are TYPE(vectorInt), DIMENSION(:), POINTER

intDegFree is INTEGER, INTENT(IN), OPTIONAL

B.9.8.21 SwapThetaNC

Description:

`SwapThetaNC` swaps the local indices on the four-tensor Θ. The OPTIONAL argument intDegFree, which can be any integer value, specifies the presence of internal degrees of freedom. This procedure is used in routines for periodic boundary conditions, see Sec. B.3.4.4.

Type:

SUBROUTINE

Syntaxes:

CALL SwapThetaNC(link, Theta, LabelLeft, LabelRight)

CALL SwapThetaNC(link, Theta, LabelLeft, LabelRight, intDegFree)

Arguments:

link is INTEGER, INTENT(IN)

Theta is COMPLEX(KIND=rKind), DIMENSION(:,:,:,:), INTENT(INOUT)

LabelLeft and LabelRight are TYPE(vectorInt), DIMENSION(:), POINTER

intDegFree is INTEGER, INTENT(IN), OPTIONAL

B.9.8.22 minmaxNLR

Description:

`minmaxNLR(link, LabelLeft, LabelRight, minNL, maxNL, minNR, maxNR)` finds the minimum and maximum values of the number on the left of link, $N_L(\alpha_{l-1}) + N_S(i_l)$, and the minimum and maximum values of the number on the right of link, $N_R(\alpha_{l+1}) + N_S(i_{l+1})$, and stores these in minNL, max NL, minNR, and maxNR, respectively. This procedure is step (4) of the number conserving two-site operation discussed in Sec. B.3.4.3.

Type:

SUBROUTINE

Arguments:

link is INTEGER, INTENT(IN)

LabelLeft and LabelRight are TYPE(vectorInt), DIMENSION(:), POINTER

minNL, maxNL, minNR, maxNR are INTEGER, INTENT(OUT)

B.9.8.23 SizeOfBlocks

Description:

`SizeOfBlocks` finds the number of blocks of fixed number on the left of link that exist and their sizes. This information is stored in BlockSize such that BlockSize(i,1) is the number on the left in site i, BlockSize(i,2) is the number of left arrangements (Schmidt vector plus onsite vector) that have the fixed number i on the left, and BlockSize(i,3) is the number of right arrangements (Schmidt vector plus onsite vector) that have the fixed number i on the left. The OPTIONAL argument intDegFree, which can be any integer value, specifies the presence of internal degrees of freedom. This procedure is step (5) of the number conserving two-site operation discussed in Sec. B.3.4.3.

Type:

SUBROUTINE

Syntaxes:

CALL SizeOfBlocks(link, BlockSize, minNL, maxNL, minNR, maxNR, LabelLeft, LabelRight)

CALL SizeOfBlocks(link, BlockSize, minNL, maxNL, minNR, maxNR, LabelLeft, LabelRight, intDegFree)

Arguments:

link is INTEGER, INTENT(IN)

BlockSize is INTEGER, DIMENSION(:,:), INTENT(IN)

minNL, maxNL, minNR, maxNR are INTEGER, INTENT(IN)

LabelLeft and LabelRight are TYPE(vectorInt), DIMENSION(:), POINTER

intDegFree is INTEGER, INTENT(IN), OPTIONAL

B.9.8.24 IndexLeft

Description:

`IndexLeft` indexes the states corresponding to allowed configurations of fixed number on the left of link and stores this information such that indL(Number on Left-minNL+1)%mi(index,1)=alpha and indL(Number on Left-minNL+1)%mi(index,1)=i, where index indexes the allowed configurations in decreasing order of reduced density matrix eigenvalue, alpha is the left Schmidt index, and i is the onsite index for the site left of the bipartite splitting. The OPTIONAL argument intDegFree, which can be any integer value, specifies the presence of internal degrees of freedom. This procedure is half of step (6) of the number conserving two-site operation discussed in Sec. B.3.4.3.

Type:

SUBROUTINE

Syntaxes:

CALL IndexLeft(link, indL, minNL, maxNL, LabelLeft)

CALL IndexLeft(link, indL, minNL, maxNL, LabelLeft, intDegFree)

Arguments:

link is INTEGER, INTENT(IN)

indL is TYPE(matrixInt), DIMENSION(:), POINTER

minNL and maxNL are INTEGER, INTENT(IN)

LabelLeft is TYPE(vectorInt), DIMENSION(:), POINTER

intDegFree is INTEGER, INTENT(IN), OPTIONAL

B.9.8.25 IndexRight

Description:

`IndexRight` indexes the states corresponding to allowed configurations of fixed number on the left and stores this information such that indR(Total Number-Number on Right-minNR+1)%mi(index,1)=alpha and indR(Total Number-Number on Right-minNR+1)%mi(index,1)=i, where index indexes the allowed configurations in decreasing order of reduced density matrix eigenvalue, alpha is the right Schmidt index, and i is the onsite index for the site to the right of the bipartite splitting. The OP-

TIONAL argument intDegFree, which can be any integer value, specifies the presence of internal degrees of freedom. This procedure is half of step (6) of the number conserving two-site operation discussed in Sec. B.3.4.3.

Type:

SUBROUTINE

Syntaxes:

CALL IndexRight(link, indR, minNL, maxNL, minNR, maxNR, LabelRight)

CALL IndexRight(link, indR, minNL, maxNL, minNR, maxNR, LabelRight,intDegFree)

Arguments:

link is INTEGER, INTENT(IN)

indR is TYPE(matrixInt), DIMENSION(:), POINTER

minNR and maxNR are INTEGER, INTENT(IN)

LabelRight is TYPE(vectorInt), DIMENSION(:), POINTER

intDegFree is INTEGER, INTENT(IN), OPTIONAL

B.9.8.26 FormBlockTheta

Description:

`FormBlockTheta(BlockTheta, indL, indR, BlockSize, Theta)` creates BlockTheta, a list of the Θs which each constitute a fixed number on the left, from Theta. This procedure is step (7) of the number conserving two-site operation discussed in Sec. B.3.4.3.

Type:

SUBROUTINE

Arguments:

BlockTheta is TYPE(matrix), DIMENSION(:), POINTER

indL and indR are TYPE(matrixInt), DIMENSION(:), POINTER

BlockSize is INTEGER, DIMENSION(:,:), INTENT(IN)

Theta is COMPLEX(KIND=rKind), DIMENSION(:,:,:,:), INTENT(INOUT)

B.9.8.27 SVDNC

Description:

`SVDNC(US, SS, VS, BlockTheta, BlockSize)` performs a singular value decomposition on each of the blocks in BlockTheta, storing the singular values of the k^{th} block in the k^{th} vector in the list of vectors SS and the k^{th} left and (transposed) right unitary matrices in the k^{th} matrix of the list of matrices U and VT, respectively. This procedure is step (8) of the number conserving two-site operation discussed in Sec. B.3.4.3.

Type:

SUBROUTINE

Arguments:

BlockTheta, US, and VS are TYPE(matrix), DIMENSION(:), POINTER

SS is TYPE(vector), DIMENSION(:), POINTER

BlockSize is INTEGER, DIMENSION(:,:), INTENT(IN)

B.9.8.28 FlattenSS

Description:

`FlattenSS(SS, ssfl, BlockSize)` collects all of the singular values from the blocks of BlockTheta into ssfl. This procedure is part of step (9) of the number conserving two-site operation discussed in Sec. B.3.4.3.

Type:

SUBROUTINE

Arguments:

SS is TYPE(vector), DIMENSION(:), POINTER

ssfl is REAL(KIND=rKind), INTENT(OUT)

BlockSize is INTEGER, DIMENSION(:,:), INTENT(IN)

B.9.8.29 Ordering

Description:

`Ordering(RealArray, order)` creates an array order such that order(i)=index of the i^{th} largest singular value in the array RealArray. This procedure is part of step (9) of the number conserving two-site operation discussed in Sec. B.3.4.3.

Type:

SUBROUTINE

Arguments:

RealArray is REAL(KIND=rKind), INTENT(IN)

order is INTEGER, DIMENSION(:), INTENT(OUT)

B.9.8.30 JudgePosition

Description:

`JudgePosition(Position, order, BlockSize)` creates an object Position such that Position(beta,1)=k and Position(beta,2)=i, with beta being the index of the singular value (in nondecreasing fashion), k being the block index (with e. g. 1 meaning the least number on the left) and i being the index of the state within the block k. In other words, Position finds the map between the singular value index beta and the block and state indices k and i. This procedure is part of step (9) of the number conserving two-site operation discussed in Sec. B.3.4.3.

Type:

SUBROUTINE

Arguments:

Position is INTEGER, DIMENSION(:,:), INTENT(OUT)

order is INTEGER, DIMENSION(:), INTENT(IN)

BlockSize is INTEGER, DIMENSION(:,:), INTENT(IN)

B.9.8.31 UpdateLabelLeft

Description:

`UpdateLabelLeft(link, LabelLeft, minNL, Position, ssfl, order)` updates LabelLeft such that

LabelLeft(link+1)%vi(beta) is the number in the beta$^{\text{th}}$ Schmidt vector (in the ordering scheme specified by the procedure `Ordering` above) to the left of link+1. This procedure is half of step (10) of the number conserving two-site operation discussed in Sec. B.3.4.3.

Type:

SUBROUTINE

Arguments:

link is INTEGER, INTENT(IN)

LabelLeft is TYPE(vectorInt), DIMENSION(:), POINTER

minNL is INTEGER, INTENT(IN)

Position is INTEGER, DIMENSION(:,:), INTENT(OUT)

ssfl is REAL(KIND=rKind), INTENT(OUT)

order is INTEGER, DIMENSION(:), INTENT(IN)

B.9.8.32 UpdateLabelRight

Description:

`UpdateLabelRight(link, LabelLeft, LabelRight)` updates LabelRight such that LabelRight(link+1)%vi(beta) is the number in the beta$^{\text{th}}$ Schmidt vector (in the ordering scheme specified by the procedure `Ordering` above) to the right of link+1. This procedure is half of step (10) of the number conserving two-site operation discussed in Sec. B.3.4.3.

Type:

SUBROUTINE

Arguments:

link is INTEGER, INTENT(IN)

LabelLeft and LabelRight are TYPE(vectorInt), DIMENSION(:), POINTER

B.9.8.33 FormLambdaNC

Description:

`FormLambdaNC(Lambda1, truncerr, ssfl, order)` updates the local tensor Lambda1= $\lambda^{[l+1]}$ using the singular values stored in ssfl to account for the action of a two-site operation. The truncation error resulting from discarding the $(d-1)\,\chi$ smallest singular values is returned in truncerr. This procedure is part of step (11) of the number conserving two-site operation discussed in Sec. B.3.4.3.

Type:

SUBROUTINE

Arguments:

ssfl is REAL(KIND=rKind), DIMENSION(:), INTENT(IN)

Lambda1 is REAL(KIND=rKind), DIMENSION(:), INTENT(INOUT)

truncerr is REAL(KIND=rKind), INTENT(OUT)

order is INTEGER, DIMENSION(:), INTENT(IN)

B.9.8.34 FormGamma1NC

Description:

`FormGamma1NC(Lambda0, Gamma1, US, indL, order, Position, BlockSize)` updates the local tensor Gamma1= $\Gamma^{[l]}$ using the matrices US obtained from the singular value decompositions of the blocks of Theta to account for the action of a two-site operation. This procedure is part of step (11) of the number conserving two-site operation discussed in Sec. B.3.4.3.

Type:

SUBROUTINE

Arguments:

Lambda0 is REAL(KIND=rKind), DIMENSION(:), INTENT(IN)

Gamma1 is COMPLEX(KIND=rKind), DIMENSION(:,:,:), INTENT(INOUT)

US is TYPE(matrix), DIMENSION(:), POINTER

indL is TYPE(matrixInt), DIMENSION(:), POINTER

order is INTEGER, DIMENSION(:), INTENT(IN)

Position is INTEGER, DIMENSION(:,:), INTENT(OUT)

BlockSize is INTEGER, DIMENSION(:,:), INTENT(IN)

B.9.8.35 FormGamma2NC

Description:

`FormGamma2NC(Gamma2, Lambda2, VS, indR, order, Position, BlockSize)` updates the local tensor Gamma2= $\Gamma^{[l+2]}$ using the matrices VS obtained from the singular value decompositions of the blocks of Theta to account for the action of a two-site operation. This procedure is part of step (11) of the number conserving two-site operation discussed in Sec. B.3.4.3.

Type:

SUBROUTINE

Arguments:

Gamma2 is COMPLEX(KIND=rKind), DIMENSION(:,:,:), INTENT(INOUT)

Lambda2 is REAL(KIND=rKind), DIMENSION(:), INTENT(IN)

VS is TYPE(matrix), DIMENSION(:), POINTER indR is TYPE(matrixInt), DIMENSION(:), POINTER

order is INTEGER, DIMENSION(:), INTENT(IN)

Position is INTEGER, DIMENSION(:,:), INTENT(OUT)

BlockSize is INTEGER, DIMENSION(:,:), INTENT(IN)

B.9.8.36 TwoSiteOpNC

Description:

`TwoSiteOpNC` operates the two-site operator Op2 on the sites separated by link consistent with number conservation. The truncation error resulting from discarding the $(d-1)\,\chi$ smallest singular values is recorded in truncerr. The OPTIONAL argument intDegFree, which can be any integer value, specifies the presence of internal degrees of freedom. This procedure performs steps (1)-(11) of the procedure discussed in Sec. B.3.4.3.

Type:

SUBROUTINE

Syntaxes:

CALL TwoSiteOpNC(link, Op2, Gammas, Lambdas, LabelLeft, LabelRight, truncerr)

CALL TwoSiteOpNC(link, Op2, Gammas, Lambdas, LabelLeft, LabelRight, truncerr,intDegFree)

Arguments:

link is INTEGER, INTENT(IN)

Op2 is either REAL(KIND=rKind) or COMPLEX(KIND=rKind), DIMENSION(:,:), INTENT(IN)

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

truncerr is REAL(KIND=rKind), INTENT(OUT)

LabelLeft and LabelRight are TYPE(vectorInt), DIMENSION(:), POINTER

intDegFree is INTEGER, INTENT(IN), OPTIONAL

B.9.8.37 SwappingNC

Description:

`SwappingNC` swaps the local indices on the four-tensor Θ and then puts this new splitting into canonical form, consistent with number conservation. The OPTIONAL argument intDegFree, which can be any integer value, specifies the presence of internal degrees of freedom. This procedure is used in routines for periodic boundary conditions, see Sec. B.3.4.4.

Type:

SUBROUTINE

Syntaxes:

CALL SwappingNC(link, Gammas, Lambdas, LabelLeft, LabelRight, truncerr)

CALL SwappingNC(link, Gammas, Lambdas, LabelLeft, LabelRight, truncerr,intDegFree)

Arguments:

link is INTEGER, INTENT(IN)

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

LabelLeft and LabelRight are TYPE(vectorInt), DIMENSION(:), POINTER

intDegFree is INTEGER, INTENT(IN), OPTIONAL

B.9.8.38 SpecialState

Description:

`SpecialState(Gammas, Lambdas, stateChar)` stores in Gammas and Lambdas the analytically known Vidal decomposition of the W, GHZ, or cluster state based on whether statechar='W', 'GHZ', or 'Cluster', respectively. The bond dimension χ of the Gammas and Lambdas must be at least two for these states, and the Cluster state algorithm only allows hard core bosons (i.e. `maxFilling` must be 1).

Type:

SUBROUTINE

Syntaxes:

CALL SpecialState(Gammas, Lambdas, 'W')

CALL SpecialState(Gammas, Lambdas, 'GHZ')

CALL SpecialState(Gammas, Lambdas, 'Cluster')

Arguments:

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

stateChar is CHARACTER(len=*), INTENT(IN)

B.9.9 observables_module Procedure Listings

Contents:

```
SUBROUTINE FormSingleSiteRho

SUBROUTINE SingleSiteDensityMatrix

INTERFACE OneSiteExpVal

        MODULE PROCEDURE OneSiteExpVal_mr,OneSiteExpVal_m,

                                           OneSiteExpVal_r,OneSiteExpVal_c
```

```
INTERFACE OneSiteVar
        MODULE PROCEDURE OneSiteVar_mr, OneSiteVar_m,
                                                OneSiteVar_r, OneSiteVar_c
SUBROUTINE GKernal
SUBROUTINE GNext
SUBROUTINE GContraction
INTERFACE TwoSiteExpValG
        MODULE PROCEDURE TwoSiteExpValG_r, TwoSiteExpValG_c,
                                                TwoSiteExpValG_rc, TwoSiteExpValG_cr
SUBROUTINE ThetaKernal
SUBROUTINE ThetaNext
SUBROUTINE TwoSiteRho
INTERFACE TwoSiteExpVal
        MODULE PROCEDURE TwoSiteExpVal_r, TwoSiteExpVal_c,
                                                TwoSiteExpVal_rc, TwoSiteExpVal_cr
FUNCTION InnerProduct
SUBROUTINE OnSiteNumber
SUBROUTINE TotalNumber
INTERFACE TotalOneSite
MODULE PROCEDURE TotalOneSite_mr, TotalOneSite_m,
                                        TotalOneSite_r, TotalOneSite_c
SUBROUTINE LocalNumDev
SUBROUTINE LocalEnergy
SUBROUTINE LocalSpin
SUBROUTINE TotalEnergy
SUBROUTINE Qdepletion
FUNCTION MeyerQmeasure
FUNCTION ChainEntropy
SUBROUTINE LocalEntropyDist
SUBROUTINE TwoBodyEntropyDist
SUBROUTINE PBphaseDist
SUBROUTINE ZetaKernelNC
SUBROUTINE ZetaNextNC
```

```
SUBROUTINE TwoPointRhoNC
INTERFACE TwoPointExpValNC
        MODULE PROCEDURE TwoPointExpValNC_r, TwoPointExpValNC_c,
                                         TwoPointExpValNC_rc, TwoPointExpValNC_cr
SUBROUTINE TotalEnergyNC
SUBROUTINE LocalEnergyNC
SUBROUTINE AllocateMeasures
SUBROUTINE DeallocateMeasures
SUBROUTINE EvaluateMeasures
```

B.9.9.1 FormSingleSiteRho

Description:

`FormSingleSiteRho(rho1, Lambda0, Gamma1, Lambda1)` calculates the single-site reduced density matrix $\hat{\rho}_l$ where Gamma1=$\Gamma^{[l]}$. The algorithm is discussed in Sec. B.3.6.1.

Type:

SUBROUTINE

Arguments:

rho1 is COMPLEX(KIND=rKind), DIMENSION(:,:)

Gamma1 is COMPLEX(KIND=rKind), DIMENSION(:,:,:)

Lambda0 and Lambda1 are REAL(KIND=rKind), DIMENSION(:)

B.9.9.2 SingleSiteDensityMatrix

Description:

`SingleSiteDensityMatrix(rho,Gammas,Lambdas)` calculates the single-site reduced density matrices and stores them in rho such that rho(l)%m(i,j)=$(\hat{\rho}_l)_{ij}$. The algorithm is discussed in Sec. B.3.6.1.

Type:

SUBROUTINE

Arguments:

rho is TYPE(matrix), DIMENSION(:), INTENT(OUT)

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

B.9.9.3 OneSiteExpVal

Description:

`OneSiteExpVal(expList,Op, Gammas, Lambdas)` calculates the expectation values of the one-site operator Op at each site, and stores these in expList such that expList(i)=Tr $\left(\hat{\mathrm{Op}}\hat{\rho}_i\right)$ in the notation of sections B.3.6.1 and B.3.6.2. Op can be either REAL(KIND=rKind), COMPLEX(KIND=rKind), TYPE(matrix), or TYPE(matrixReal). Details on the algorithm can be found in Sec. B.3.6.1.

Type:

SUBROUTINE

Arguments:

expList is REAL(KIND=rKind), DIMENSION(:), INTENT(OUT) or COMPLEX(KIND=rKind), DIMENSION(:), INTENT(OUT)

Op is REAL(KIND=rKind) DIMENSION(:,:), INTENT(IN); COMPLEX(KIND=rKind), DIMENSION(:,:), INTENT(IN); TYPE(matrix); or TYPE(matrixReal) and must match the type of expList (i.e. expList should be complex if Op is).

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

B.9.9.4 OneSiteVar

Description:

`OneSiteVar(varList,Op, Gammas, Lambdas)` calculates the variance of the one-site operator Op at each site, and stores these in expList such that varList(i)=$\mathrm{Tr}\left(\hat{\mathrm{Op}}^2\hat{\rho}_i\right)-\left[\mathrm{Tr}\left(\hat{\mathrm{Op}}\hat{\rho}_i\right)\right]^2$ in the notation of sections B.3.6.1 and B.3.6.2. Op can be either REAL(KIND=rKind), COMPLEX(KIND=rKind), TYPE(matrix), or TYPE(matrixReal). Details on the algorithm can be found in Sec. B.3.6.1.

Type:

SUBROUTINE

Arguments:

varList is REAL(KIND=rKind), DIMENSION(:), INTENT(OUT) or COMPLEX(KIND=rKind), DIMENSION(:), INTENT(OUT)

Op is REAL(KIND=rKind) DIMENSION(:,:), INTENT(IN); COMPLEX(KIND=rKind), DIMENSION(:,:), INTENT(IN); TYPE(matrix); or TYPE(matrixReal) and must match the type of expList (i.e. expList should be complex if Op is).

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

B.9.9.5 GKernal

Description:

`GKernal(gee, Gamma, GammaP,Lambda)` constructs $G^{[k]}_{\alpha_k\beta_k}$ as defined in Eq. (B.142) and stores it in gee. It is the first step in calculating a two-site observable that is a tensor product of one-site observables as discussed in Sec. B.3.6.4.

Type:

SUBROUTINE

Arguments:

gee is COMPLEX(KIND=rKind), DIMENSION(:,:), INTENT(INOUT)

Gamma, GammaP are COMPLEX(KIND=rKind), DIMENSION(:,:,:), INTENT(IN)

Lambda is REAL(KIND=rKind), DIMENSION(:), INTENT(IN)

B.9.9.6 GNext

Description:

`GNext(gee, Gamma, GammaP,Lambda)` constructs $G^{[k-1]}_{\alpha_{k-1}\beta_{k-1}}$ as defined in Eq. (B.143) and stores it in gee. It is the recursive step in calculating a two-site observable that is a tensor product of one-site observables as discussed in Sec. B.3.6.4.

Type:

SUBROUTINE

Arguments:

gee is COMPLEX(KIND=rKind), DIMENSION(:,:), INTENT(INOUT)

Gamma, GammaP are COMPLEX(KIND=rKind), DIMENSION(:,:,:), INTENT(IN)

Lambda is REAL(KIND=rKind), DIMENSION(:), INTENT(IN)

B.9.9.7 GContraction

Description:

`GContraction(obsv,gee, Gamma, GammaP,Lambda1,Lambda2)` constructs the final contraction involved in computing a two-site observable that is a tensor product of one-site observables as in Eq. (B.144) and stores it in obsv. It is the final step in calculating a two-site observable that is a tensor product of one-site observables as discussed in Sec. B.3.6.4.

Type:

SUBROUTINE

Arguments:

gee is COMPLEX(KIND=rKind), DIMENSION(:,:), INTENT(INOUT)

Gamma, GammaP are COMPLEX(KIND=rKind), DIMENSION(:,:,:), INTENT(IN)

Lambda1, Lambda2 are REAL(KIND=rKind), DIMENSION(:), INTENT(IN)

B.9.9.8 TwoSiteExpValG

Description:

`TwoSiteExpValG(observable, Op1, Op2, Gammas, Lambdas)` calculates the expectation values of the two-site operator Op1⊗Op2, and stores these in observable such that observable(i,j)=Tr $\left(\mathrm{Op1}\,\hat{\otimes}\,\mathrm{Op2}\hat{\rho}_{ij}\right)$ in the notation of sections B.3.6.1 and B.3.6.2. Op2 and Op1 can be either REAL(KIND=rKind), COMPLEX(KIND=rKind), TYPE(matrix), or TYPE(matrixReal). The OPTIONAL argument phaseStat specifies that a Fermi phase should be included in the calculation, see Sec. B.3.6.3. Details on the algorithm can be found in Sec. B.3.6.2.

Type:

SUBROUTINE

Syntaxes:

CALL TwoSiteExpValG(observable, Op1, Op2, Gammas, Lambdas)

CALL TwoSiteExpValG(observable, Op1, Op2, Gammas, Lambdas,phaseStat)

Arguments:

observable is COMPLEX(KIND=rKind), DIMENSION(:,:), INTENT(OUT)

Op1 is REAL(KIND=rKind) DIMENSION(:,:), INTENT(IN); COMPLEX(KIND=rKind), DIMENSION(:,:), INTENT(IN); TYPE(matrix); or TYPE(matrixReal)

Op2 is REAL(KIND=rKind) DIMENSION(:,:), INTENT(IN); COMPLEX(KIND=rKind), DIMENSION(:,:), INTENT(IN); TYPE(matrix); or TYPE(matrixReal)

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

phaseStat is INTEGER, INTENT(IN), OPTIONAL

B.9.9.9 ThetaKernal

Description:

`ThetaKernal(Theta,Lambda1,Gamma,Lambda2)` constructs $\Theta^{i_k i'_k}_{\alpha_{k-1}\alpha'_{k-1}}$ as defined in Eq. (B.126) where Gamma=$\Gamma^{[k]}$. It is the first step in calculating the two-site reduced density matrix as discussed in Sec. B.3.6.2.

Type:

SUBROUTINE

Arguments:

Theta is COMPLEX(KIND=rKind), DIMENSION(:,:,:,:), INTENT(OUT)

Gamma1 is COMPLEX(KIND=rKind), DIMENSION(:,:,:), INTENT(IN)

Lambda1 and Lambda2 are REAL(KIND=rKind), DIMENSION(:), INTENT(IN)

B.9.9.10 ThetaNext

Description:

`ThetaNext(Theta, Gamma, GammaP, Lambda)` constructs $\Theta^{i_k i'_k}_{\alpha_{k-2}\alpha'_{k-2}}$ from $\Theta^{i_k i'_k}_{\alpha_{k-1}\alpha'_{k-1}}$ as defined in Eq. (B.127) where Gamma=$\Gamma^{[k-1]}$. It is the second step in calculating the two-site reduced density matrix as discussed in Sec. B.3.6.2.

Type:

SUBROUTINE

Arguments:

Theta is COMPLEX(KIND=rKind), DIMENSION(:,:,:,:), INTENT(INOUT)

Gamma and GammaP are COMPLEX(KIND=rKind), DIMENSION(:,:,:)

Lambda is REAL(KIND=rKind), DIMENSION(:)

B.9.9.11 TwoSiteRho

Description:

`TwoSiteRho(rho2, Theta, Gamma, GammaP, Lambda)` constructs $\hat{\rho}_{lk}$ from $\Theta^{i_k i'_k}_{\alpha_l \alpha'_l}$ as defined in Eq. (B.128) where Gamma=$\Gamma^{[l]}$. It is the final step in calculating the two-site reduced density matrix as discussed in Sec. B.3.6.2.

Type:

SUBROUTINE

Arguments:

rho2 is COMPLEX(KIND=rKind), DIMENSION(:,:), INTENT(OUT)

Theta is COMPLEX(KIND=rKind), DIMENSION(:,:,:,:), INTENT(IN)

Gamma and GammaP are COMPLEX(KIND=rKind), DIMENSION(:,:,:)

Lambda is REAL(KIND=rKind), DIMENSION(:)

B.9.9.12 TwoSiteExpVal

Description:

`TwoSiteExpVal(observable, Op1, Op2, Gammas, Lambdas)` calculates the expectation values of the two-site operator Op2, and stores these in observable such that observable(i,j)=Tr $\left(\hat{\mathrm{Op2}}\hat{\rho}_{ij}\right)$, $i \neq j$, observable(i,j)=Tr $\left(\hat{\mathrm{Op1}}\hat{\rho}_{i}\right)$, $i = j$ in the notation of sections B.3.6.1 and B.3.6.2. Op2 and Op1 can be either REAL(KIND=rKind), COMPLEX(KIND=rKind), TYPE(matrix), or TYPE(matrixReal). The OPTIONAL argument phaseStat specifies that a Fermi phase should be included in the calculation, see Sec. B.3.6.3. Details on the algorithm can be found in Sec. B.3.6.2.

Type:

SUBROUTINE

Syntaxes:

CALL TwoSiteExpVal(observable, Op1, Op2, Gammas, Lambdas)

CALL TwoSiteExpVal(observable, Op1, Op2, Gammas, Lambdas,phaseStat)

Arguments:

observable is COMPLEX(KIND=rKind), DIMENSION(:,:), INTENT(OUT)

Op1 is REAL(KIND=rKind) DIMENSION(:,:), INTENT(IN); COMPLEX(KIND=rKind), DIMENSION(:,:), INTENT(IN); TYPE(matrix); or TYPE(matrixReal)

Op2 is REAL(KIND=rKind) DIMENSION(:,:), INTENT(IN); COMPLEX(KIND=rKind), DIMENSION(:,:), INTENT(IN); TYPE(matrix); or TYPE(matrixReal)

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

phaseStat is INTEGER, INTENT(IN), OPTIONAL

B.9.9.13 InnerProduct

Description:

`InnerProduct(GammasL, LambdasL, GammasR, LambdasR)` returns the overlap of the wavefunction defined by the local tensors GammasL and LambdasL with the wavefunction defined by the local tensors GammasR and LambdasR, $\langle\Psi_L|\Psi_R\rangle$. It can be used to compute the Fidelity (also known as the Loschmidt echo) $|\langle\psi\left(0\right)|\psi\left(t\right)\rangle|^2$, or simply to check normalization. See Sec. B.3.6.5 for a discussion of the algorithm.

Type:

COMPLEX(KIND=rKind) FUNCTION

Arguments:

GammasL and GammasR are TYPE(tensor), DIMENSION(:), POINTER

LambdasL and LambdasR are TYPE(vector), DIMENSION(:), POINTER

B.9.9.14 OnSiteNumber

Description:

`OnSiteNumber(number, Gammas, Lambdas,siteIndex, comPonent)` calculates the total number on the site siteindex, and returns this in number such that number=$\langle \hat{n}_{\mathrm{siteIndex}} \rangle$ for systems without degrees of freedom and number=$\langle \hat{n}_{\mathrm{siteIndex,comPonent}} \rangle$ if the OPTIONAL argument comPonent specifying internal degrees of freedom is present.

Type:

SUBROUTINE

Syntaxes:

CALL OnSiteNumber(number, Gammas, Lambdas,siteIndex)

CALL OnSiteNumber(number, Gammas, Lambdas,siteIndex, comPonent)

Arguments:

number is REAL(KIND=rKind)

comPonent is INTEGER, INTENT(IN), OPTIONAL

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

siteIndex is INTEGER, INTENT(IN)

B.9.9.15 TotalNumber

Description:

`TotalNumber(number, Gammas, Lambdas, comPonent)` calculates the site-averaged total number across all sites, and returns this in number such that number=$\langle \hat{n} \rangle / L$ for systems without degrees of freedom and number=$\langle \hat{n}_{\mathrm{comPonent}} \rangle / L$ if the OPTIONAL argument comPonent specifying internal degrees of freedom is present.

Type:

SUBROUTINE

Syntaxes:

CALL TotalNumber(number, Gammas, Lambdas)

CALL TotalNumber(number, Gammas, Lambdas, comPonent)

Arguments:

number is REAL(KIND=rKind)

comPonent is INTEGER, INTENT(IN), OPTIONAL

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

B.9.9.16 TotalOneSite

Description:

`TotalOneSite(total, Op, Gammas, Lambdas)` calculates the value of Op at each site and stores the values in total such that total(i)=Tr(Op$\hat{\rho}_i$). Op can be REAL(KIND=rKind), COMPLEX(KIND=rKind), TYPE(matrix), or TYPE(matrixReal).

Type:

SUBROUTINE

Arguments:

total is REAL(KIND=rKind) or COMPLEX(KIND=rKind), DIMENSION(:), INTENT(INOUT).

Op is REAL(KIND=rKind) DIMENSION(:,:), INTENT(IN); COMPLEX(KIND=rKind), DIMENSION(:,:), INTENT(IN); TYPE(matrix); or TYPE(matrixReal)

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

B.9.9.17 LocalNumDev

Description:

`LocalNumDev(numbers, deviations, Gammas, Lambdas, comPonent)` calculates the total number and number standard deviation on the site siteindex, and returns these in number and deviations such that number=$\langle \hat{n}_{\mathrm{siteIndex}} \rangle$, deviations=$\sqrt{\langle \hat{n}^2_{\mathrm{siteIndex}} \rangle - \langle \hat{n}_{\mathrm{siteIndex}} \rangle^2}$ for systems without degrees of freedom and number=$\langle \hat{n}_{\mathrm{siteIndex,comPonent}} \rangle$, deviations=$\sqrt{\langle \hat{n}^2_{\mathrm{siteIndex,comPonent}} \rangle - \langle \hat{n}_{\mathrm{siteIndex,comPonent}} \rangle^2}$ if the OPTIONAL argument comPonent specifying internal degrees of freedom is present.

Type:

SUBROUTINE

Syntaxes:

LocalNumDev(numbers, deviations, Gammas, Lambdas)

LocalNumDev(numbers, deviations, Gammas, Lambdas, comPonent)

Arguments:

number and deviations are REAL(KIND=rKind), DIMENSION(:)

comPonent is INTEGER, INTENT(IN), OPTIONAL

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

B.9.9.18 LocalEnergy

Description:

`LocalEnergy(energy, H, Gammas, Lambdas)` calculates the energy associated with each two-site block and stores it in energy such that energy(i)=$\langle \hat{H}_{i,i+1} \rangle$. If periodic boundary conditions are used, energy(systemSize)=$\langle \hat{H}_{L,1} \rangle$.

Type:

SUBROUTINE

Arguments:

energy is REAL(KIND=rKind), DIMENSION(:), INTENT(OUT)

H is TYPE(matrix), DIMENSION(:), POINTER

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

B.9.9.19 LocalSpin

Description:

`LocalSpin(spinVec, Gammas, Lambdas)` calculates the expectation value of the total spin squared $\langle i_1 i_2 | \left(\hat{\mathbf{S}}_1 + \hat{\mathbf{S}}_2 \right)^2 | i_1 i_2 \rangle$ for each contiguous pair of lattice sites i_1 and i_2, and returns this value in spinVec.

Type:

SUBROUTINE

Arguments:

spinVec is REAL(KIND=rKind), DIMENSION(:), INTENT(OUT)

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

B.9.9.20 TotalEnergy

Description:

`TotalEnergy(energy, H, Gammas, Lambdas)` calculates the total energy of the system and stores it in energy.

Type:

SUBROUTINE

Arguments:

energy is REAL(KIND=rKind), INTENT(OUT)

H is TYPE(matrix), DIMENSION(:), POINTER

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

B.9.9.21 Qdepletion

Description:

`Qdepletion(depletion, rho, population)` calculates the quantum depletion, defined as $D \equiv 1 - N_0/N$, where N_0 is the number of particles in the most highly occupied natural orbital (eigenvector of the single-particle density matrix) and N is the total number of particles (here denoted "population")[7]. We find this numerically by performing an SVD on the single-particle density matrix $(\rho_{\mathrm{sp}})_{ij} \equiv \langle \hat{a}_j^{\dagger}\hat{a}_i \rangle$; N_0 is the largest singular value of this matrix. This routine can also return the condensate wave function, the system's most highly occupied natural orbital, which is specified by the largest eigenvector of the single-particle density matrix. This is returned in the OPTIONAL argument CW.

Type:

SUBROUTINE

Syntaxes:

CALL Qdepletion(depletion, rho, population)

CALL Qdepletion(depletion, rho, population, CW)

Arguments:

depletion is REAL(KIND=rKind), INTENT(OUT)

rho is COMPLEX(KIND=rKind), DIMENSION(:,:), INTENT(IN)

population is REAL(KIND=rKind), INTENT(IN)

CW is COMPLEX(KIND=rKind), DIMENSION(:), INTENT(OUT), OPTIONAL

B.9.9.22 MeyerQmeasure

Description:

`MeyerQmeasure(Gammas, Lambdas)` returns the Meyer Q-measure [23–25]

$$Q \equiv \frac{d}{d-1}\left[1-\frac{1}{L}\sum_{k=1}^{L}\mathrm{Tr}\left(\rho_k^2\right)\right]\,, \tag{B.217}$$

where ρ_k is the single-site density matrix obtained by tracing over all but the k^{th} lattice site, and the factor outside of the bracket is a normalization factor (d is the on-site dimension). This gives an average measure of the entanglement of a single site with the rest of the system. The Q-measure can also be interpreted as the average local impurity (recall that the $\mathrm{Tr}(\hat{\rho}^2) = 1$ if and only if $\hat{\rho}$ is a pure state).

Type:

REAL(KIND=rKind) FUNCTION

Arguments:

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

B.9.9.23 ChainEntropy

Description:

`ChainEntropy(link, Lambdas)` returns the entropy of entanglement of the sites to the left of link with the sites to the right of link in the MPS approximation, defined to be

$$S_l \equiv -\sum_{\alpha}^{\chi} \left(\lambda_{\alpha}^{[l]}\right)^2 \log_d \left[\left(\lambda_{\alpha}^{[l]}\right)^2\right] . \tag{B.218}$$

Because of the finite entanglement of an MPS representation, this object is bounded by $\log_d \chi$.

Type:

REAL(KIND=rKind) FUNCTION

Arguments:

link is INTEGER, INTENT(IN)

Lambdas is TYPE(vector), DIMENSION(:), POINTER

B.9.9.24 LocalEntropyDist

Description:

If the OPTIONAL argument tsalliSq is present, then `LocalEntropyDist` calculates the single site Tsallis entropy

$$S_{q,k} \equiv \frac{1}{1-q}\left[\mathrm{Tr}\left(\hat{\rho}_k^q\right)-1\right]\ , \tag{B.219}$$

for each site k, where q=tsalliSq. Otherwise, `LocalEntropyDist` computes the single-site von Neumann entropy

$$S_{\mathrm{vN},k} \equiv -\mathrm{Tr}\left[\hat{\rho}_k \log_d\left(\hat{\rho}_k\right)\right] \tag{B.220}$$

for each site k. The resulting values are stored in the array entDist. The procedure computes the matrix logarithm (power) by diagonalizing the single-site density matrix and then taking the logarithms (powers) of the eigenvalues.

Type:

SUBROUTINE

Syntaxes:

CALL LocalEntropyDist(entDist, Gammas, Lambdas)

CALL LocalEntropyDist(entDist, Gammas, Lambdas, tsalliSq)

Arguments:

entDist is REAL(KIND=rKind), DIMENSION(:), INTENT(OUT)

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

tsallisQ is REAL(KIND=rKind), INTENT(IN), OPTIONAL

B.9.9.25 TwoBodyEntropyDist

Description:

If the OPTIONAL argument tsalliSq is present, then `TwoBodyEntropyDist` calculates the two-site Tsallis entropy

$$S_{q;i,j} \equiv \frac{1}{1-q}\left[\mathrm{Tr}\left(\hat{\rho}_{i,j}^{q}\right)-1\right]\ , \tag{B.221}$$

for each distinct pair of sites i and j, where q=tsalliSq. Otherwise, `TwoBodyEntropyDist` computes the two-site von Neumann entropy

$$S_{\mathrm{vN};i,j} \equiv -\mathrm{Tr}\left[\hat{\rho}_{i,j}\log_{d}\left(\hat{\rho}_{i,j}\right)\right] \tag{B.222}$$

for each distinct pair of sites i and j. The resulting values are stored in the matrix entDist. The procedure computes the matrix logarithm (power) by diagonalizing the single-site density matrix and then taking the logarithms (powers) of the eigenvalues.

Type:

SUBROUTINE

Syntaxes:

CALL TwoBodyEntropyDist(entDist, Gammas, Lambdas)

CALL TwoBodyEntropyDist(entDist, Gammas, Lambdas, tsalliSq)

Arguments:

entDist is REAL(KIND=rKind), DIMENSION(:,:), INTENT(OUT)

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

tsallisQ is REAL(KIND=rKind), INTENT(IN), OPTIONAL

B.9.9.26 PBphaseDist

Description:

`PBphaseDist(phaseDist, Gammas, Lambdas)` calculates the expectation of the Pegg-Barnett phase operator, Eq. (B.213) at each lattice site and stores these in the array phaseDist. This is currently only supported for codes without internal degrees of freedom.

Type:

SUBROUTINE

Arguments:

phaseDist is REAL(KIND=rKind), DIMENSION(:), INTENT(OUT)

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

B.9.9.27 ZetaKernelNC

Description:

`ZetaKernelNC(Zeta, Lambda0, Gamma1, Lambda1, LabelL0, LabelL1)` constructs $\zeta^{i_k i'_k}_{\alpha_{k-1}\alpha'_{k-1}}$, which is identical to Θ as defined in Eq. (B.126) except that the number conservation constraint $N_L\left(\alpha_{k-1}\right) + N_S\left(i_k\right) = N_S\left(i'_k\right) + N_R\left(\alpha'_{k-1}\right)$ is enforced. The site k is identified from the input as Gamma=$\Gamma^{[k]}$. It is the first step in calculating the two-site reduced density matrix for number conservation, analogous to `ThetaKernal` as discussed in Sec. B.3.6.2. This routine currently does not support internal degrees of freedom.

Type:

SUBROUTINE

Arguments:

Zeta is COMPLEX(KIND=rKind), DIMENSION(:,:,:,:), INTENT(OUT)

Gamma1 is COMPLEX(KIND=rKind), DIMENSION(:,:,:), INTENT(IN)

Lambda0 and Lambda1 are REAL(KIND=rKind), DIMENSION(:), INTENT(IN)

LabelL0 and LabelL1 are INTEGER, DIMENSION(:), INTENT(IN)

B.9.9.28 ZetaNextNC

Description:

`ZetaNextNC(Zeta, Lambda0, Gamma1, LabelL0, LabelL1)` constructs $\zeta^{i_k i'_k}_{\alpha_{k-2}\alpha'_{k-2}}$ which is identical to Θ defined in Eq. (B.127) except that the number conservation constraint $N_L\left(\alpha_{k-1}\right)+N_S\left(i_k\right)=N_S\left(i'_k\right)+N_R\left(\alpha'_{k-1}\right)$ is enforced. The site $k-1$ is identified from the input as Gamma=$\Gamma^{[k-1]}$. It is the second step in calculating the two-site reduced density matrix for number conservation, analogous to `ThetaNext` as discussed in Sec. B.3.6.2. This routine currently does not support internal degrees of freedom.

Type:

SUBROUTINE

Arguments:

Zeta is COMPLEX(KIND=rKind), DIMENSION(:,:,:,:), INTENT(INOUT)

Gamma1 is COMPLEX(KIND=rKind), DIMENSION(:,:,:)

Lambda0 is REAL(KIND=rKind), DIMENSION(:)

LabelL0 and LabelL1 are INTEGER, DIMENSION(:), INTENT(IN)

B.9.9.29 TwoPointRhoNC

Description:

`TwoPointRhoNC(rho2, Zeta, Lambda0, Gamma1, LabelL0, LabelL1)` constructs $\hat{\rho}_{lk}$ subject to number conservation from $\zeta^{i_k i'_k}_{\alpha_l \alpha'_l}$ just as the number non-conserving $\hat{\rho}_{lk}$ is constructed from $\Theta^{i_k i'_k}_{\alpha_l \alpha'_l}$ as in Eq. (B.128) where Gamma=$\Gamma^{[l]}$. This routine currently does not support internal degrees of freedom.

Type:

SUBROUTINE

Arguments:

rho2 is COMPLEX(KIND=rKind), DIMENSION(:,:), INTENT(OUT)

Zeta is COMPLEX(KIND=rKind), DIMENSION(:,:,:,:), INTENT(IN)

Gamma1 is COMPLEX(KIND=rKind), DIMENSION(:,:,:)

Lambda0 is REAL(KIND=rKind), DIMENSION(:)

LabelL0 and LabelL1 are INTEGER, DIMENSION(:), INTENT(IN)

B.9.9.30 TwoSiteExpValNC

Description:

`TwoSiteExpValNC(observable, Op1, Op2, Lambdas, Gammas, LabelLeft)` calculates the expectation values of the two-site operator Op2 consistent with number conservation, and stores these in observable such that observable(i,j)=Tr $\left(\hat{\mathrm{Op}}2\hat{\rho}_{ij}\right)$, $i \neq j$, observable(i,j)=Tr $\left(\hat{\mathrm{Op}}1\hat{\rho}_{i}\right)$, $i = j$ in the notation of sections B.3.6.1 and B.3.6.2. Op2 and Op1 can be either REAL(KIND=rKind) or COMPLEX(KIND=rKind). Details on the algorithm can be found in Sec. B.3.6.2.

Type:

SUBROUTINE

Arguments:

observable is COMPLEX(KIND=rKind), DIMENSION(:,:), INTENT(OUT)

Op1 is REAL(KIND=rKind) or COMPLEX(KIND=rKind), DIMENSION(:,:), INTENT(IN)

Op2 is REAL(KIND=rKind) or COMPLEX(KIND=rKind), DIMENSION(:,:), INTENT(IN)

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

LabelLeft is TYPE(vectorInt), DIMENSION(:), POINTER

B.9.9.31 TotalEnergyNC

Description:

`TotalEnergyNC(energy, H, Gammas, Lambdas, LabelLeft)` calculates the total energy of the system consistent with number conservation and stores it in energy. This routine currently does not support internal degrees of freedom.

Type:

SUBROUTINE

Arguments:

energy is REAL(KIND=rKind), INTENT(OUT)

H is TYPE(matrix), DIMENSION(:), POINTER

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

LabelLeft is TYPE(vectorInt), DIMENSION(:), POINTER

B.9.9.32 LocalEnergyNC

Description:

`LocalEnergyNC(energy, H, Gammas, Lambdas, LabelLeft)` calculates the energy associated with each two-site block consistent with number conservation and stores it in energy such that energy(i)=$\langle \hat{H}_{i,i+1} \rangle$. This routine currently does not support internal degrees of freedom.

Type:

SUBROUTINE

Arguments:

energy is REAL(KIND=rKind), DIMENSION(:), INTENT(OUT)

H is TYPE(matrix), DIMENSION(:), POINTER

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

LabelLeft is TYPE(vectorInt), DIMENSION(:), POINTER

B.9.9.33 AllocateMeasures

Description:

`AllocateMeasures(Measures,numLocal, numAvg, numCorr, numFermiCorr, numEnt)` allocates the `measure` derived type Measures to hold numLocal local observables, numAvg average measures etc. numEnt can take on the values 0, 1, or 2. 0 specifies that only the Q-measure is computed; 1 specifies that the Q measure, the von Neumann entropy at each site, and the Chain entropy of each bond are computed, and 2 specifies that the Q measure, the von Neumann entropy at each site, the Chain entropy of each bond, and the von Neumann entropy of each pair of sites are computed. For more information on the `measure` derived type, see Sec.sec:OSTEBD:measure.

Type:

SUBROUTINE

Arguments:

Measures is TYPE(measure)

numLocal, numAvg, numCorr, numFermiCorr, numEnt are INTEGER, INTENT(IN)

B.9.9.34 DeallocateMeasures

Description:

`DeallocateMeasures(Measures)` deallocates the `measure` derived type Measures. For more information on the `measure` derived type, see Sec.sec:OSTEBD:measure.

Type:

SUBROUTINE

Arguments:

Measures is TYPE(measure)

B.9.9.35 EvaluateMeasures

Description:

`EvaluateMeasures(Measures,Gammas, Lambdas, H)` evaluates the suite of measures defined by the `measure` derived type Measures and returns them in Measures. For more information on the `measure` derived type, see Sec.sec:OSTEBD:measure.

Type:

SUBROUTINE

Arguments:

Measures is TYPE(measure)

H is TYPE(matrix), DIMENSION(:), POINTER

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

B.9.10 io_module Procedure Listings

Contents:

```
SUBROUTINE createFileName
INTERFACE appendBaseName
        MODULE PROCEDURE appendBaseName_r, appendBaseName_i, appendBaseName_c
SUBROUTINE copyName
FUNCTION CheckName
SUBROUTINE openUnit
SUBROUTINE RecordLambdas
SUBROUTINE RecordGammas
SUBROUTINE RecordLabel
SUBROUTINE readGammaLambda
SUBROUTINE readGammaLambdaLabels
INTERFACE RecordOp
        MODULE PROCEDURE RecordOp_m,RecordOp_mr,
                                          RecordOp_c,RecordOp_r
INTERFACE RecordOpList
```

```
        MODULE PROCEDURE RecordOpList_m,RecordOpList_mr
INTERFACE RecordOneSiteOb
        MODULE PROCEDURE RecordOneSiteOb_r,RecordOneSiteOb_c
INTERFACE RecordTwoSiteOb
        MODULE PROCEDURE RecordTwoSiteOb_r,RecordTwoSiteOb_c
```

B.9.10.1 createFileName

Description:

`createFileName(basename,diRectory)` initializes the character string basename with the character string diRectory.

Type:

SUBROUTINE

Arguments:

baseName is CHARACTER(len=*), INTENT(INOUT)

diRectory is CHARACTER(len=*), INTENT(IN)

B.9.10.2 appendBaseName

Description:

`appendBaseName` has three options for appending data to a given baseLen-length character string basename. The first option is `appendBaseName(basename,partName,partDigs,partValue)` with partValue a REAL quantity, which will append the character string partName, and then the REAL value partValue to partDigs decimal places to basename. The next option is `appendBaseName(basename,partName,partDigs,partValue)` with partValue an INTEGER quantity, which will append the character string partName, and then the partDigs-length INTEGER partValue to basename. The final option is `appendBaseName_c(basename,partName)`, which will append the character string partName to basename. This function is useful for dynamically naming output files based on parameters used in the code.

Type:

SUBROUTINE

Syntaxes:

partValue=REAL or INTEGER:

CALL appendBaseName(basename,partName,partDigs,partValue)

partValue=CHARACTER: CALL appendBaseName(basename,partName)

Arguments:

For appending REAL or INTEGER data:

baseName is CHARACTER(len=*), INTENT(INOUT)

partname is CHARACTER(len=*), INTENT(IN)

partDigs is INTEGER, INTENT(IN)

partValue is REAL(KIND=rKind), INTENT(IN) or INTEGER, INTENT(IN)

For appending a character string:

baseName is CHARACTER(len=*), INTENT(INOUT)

partname is CHARACTER(len=*), INTENT(IN)

B.9.10.3 copyName

Description:

`copyName(name1,name2)` copies the character string name1 to the character string name2.

Type:

SUBROUTINE

Arguments:

name1 is CHARACTER(len=*), INTENT(IN)

name2 is CHARACTER(len=*), INTENT(OUT)

B.9.10.4 CheckName

Description:

`CheckName(baseName)` returns TRUE if the file with the name basename exists and FALSE otherwise.

Type:

LOGICAL FUNCTION

Arguments:

baseName is CHARACTER(len=*), INTENT(IN)

B.9.10.5 openUnit

Description:

`openUnit(fileName,myUnit)` opens the file with name fileName, giving it the UNIT myUnit. `openUnit(fileName,myUnit, openKind)` produces different results based on the value of the OPTIONAL character openKind. If openKind='N', it opens the file with STATUS='NEW', write capability only; if openKind='O', it opens the file with STATUS='OLD', read and write capability; and if openKind='A', it opens the file with write capability only, appending new data to the existing data. This allows for strict checking before files are overwritten.

Type:

SUBROUTINE

Syntaxes:

CALL openUnit(fileName,myUnit)

CALL openUnit(fileName,myUnit, openKind)

Arguments:

fileName is CHARACTER(len=*), INTENT(IN)

openKind is CHARACTER, INTENT(IN), OPTIONAL.

B.9.10.6 RecordLambdas

Description:

`RecordLambdas(fileid, Lambdas,openKind)` writes the contents of Lambdas onto the file whose UNIT is fileID. If openKind='S' then the data is written in scientific notation and if openKind='B' the data is written in binary.

Type:

SUBROUTINE

Arguments:

fileid is INTEGER, INTENT(IN)

Lambdas is TYPE(vector), DIMENSION(:), POINTER

openKind is CHARACTER, INTENT(IN)

B.9.10.7 RecordGammas

Description:

`RecordGammas(fileid, Gammas, openKind)` writes the contents of Gammas onto the file whose UNIT is fileID. If openKind='S' then the data is written in scientific notation and if openKind='B' the data is written in binary.

Type:

SUBROUTINE

Arguments:

fileid is INTEGER, INTENT(IN)
Gammas is TYPE(tensor), DIMENSION(:), POINTER
openKind is CHARACTER, INTENT(IN)

B.9.10.8 RecordLabel

Description:

`RecordLabel(fileid, LabelLorR)` writes the contents of Label onto the file whose UNIT is fileID. Because Label is TYPE(vectorInt), the data is always written in INTEGER format.

Type:

SUBROUTINE

Arguments:

fileid is INTEGER, INTENT(IN)

Label is TYPE(vectorInt), DIMENSION(:), POINTER

B.9.10.9 readGammaLambda

Description:

`readGammaLambda(lambdafileID, gammafileID,Gammas,Lambdas, openKind, chiNow)` reads the Γs and λs from the files with UNITs gammafileID and lambdafileID, respectively, into the objects Gammas and Lambdas. chiNow specifies the entanglement cutoff parameter of the Γs and λs in the files. If openKind='S' the data in the files is assumed to be in scientific notation and if openKind='B' the data in the files is assumed to be in binary.

Type:

SUBROUTINE

Arguments:

lambdafileid, gammafileid, and chiNow are INTEGER, INTENT(IN)
Lambdas is TYPE(vector), DIMENSION(:), POINTER
Gammas is TYPE(tensor), DIMENSION(:), POINTER
openKind is CHARACTER, INTENT(IN)

B.9.10.10 readGammaLambdaLabels

Description:

`readGammaLambdaLabels(lambdafileID, gammafileID,labelleftFileID , labelrightFileID,Gammas,Lambdas,LabelLeft, LabelRight, openKind, chiNow)` reads the Γs, λs, LabelLeft, and LabelRight from the files with UNITs gammafileID, lambdafileID, labelleftFileID, and labelrightFileID, respectively, into the objects Gammas, Lambdas, LabelLeft, and LabelRight. chiNow specifies the entanglement cutoff parameter of the Γs and λs in the files. If openKind='S' the data in the files is assumed to be in scientific notation and if openKind='B' the data in the files is assumed to be in binary (except for the Labels, which are always written as INTEGERs, see Sec. B.9.10.8).

Type:

SUBROUTINE

Arguments:

lambdafileid, gammafileid, labelleftFileID, labelrightFileID, and chiNow are INTEGER, INTENT(IN)

Lambdas is TYPE(vector), DIMENSION(:), POINTER

Gammas is TYPE(tensor), DIMENSION(:), POINTER

LabelLeft and labelRight are TYPE(vectorInt), DIMENSION(:), POINTER

openKind is CHARACTER, INTENT(IN)

B.9.10.11 RecordOp

Description:

`RecordOp(fileid, Op)` writes the operator Op onto the file whose UNIT is fileid. The procedure accepts TYPE(matrix); TYPE(matrixReal); REAL(KIND=rKind), DIMENSION(:,:); or COMPLEX(KIND=rKind), DIMENSION(:,:) Op.

Type:

SUBROUTINE

Arguments:

fileid is INTEGER, INTENT(IN)

Op is TYPE(matrix); TYPE(matrixReal); REAL(KIND=rKind), DIMENSION(:,:); or COMPLEX(KIND=rKind), DIMENSION(:,:)

B.9.10.12 RecordOpList

Description:

`RecordOpList(fileid, Op)` writes the list of operators Op onto the file whose UNIT is fileid. The procedure accepts TYPE(matrix), DIMENSION(:), POINTER or TYPE(matrixReal), DIMENSION(:), POINTER Op.

Type:

SUBROUTINE

Arguments:

fileid is INTEGER, INTENT(IN)

Op is TYPE(matrix), DIMENSION(:), POINTER or TYPE(matrixReal), DIMENSION(:), POINTER

B.9.10.13 RecordOneSiteOb

Description:

`RecordOneSiteOb(fileid, Obs)` writes the `systemSize`-length list of one-site expectation values Ob onto the file whose UNIT is fileid. `RecordOneSiteOb(fileid, Obs, timeN)` records the value timeN and then the `systemSize`-length list of one-site expectation values Ob onto the file whose UNIT is fileid. The procedure accepts REAL or COMPLEX Ob.

Type:

SUBROUTINE

Syntaxes:

CALL RecordOneSiteOb(fileid, Obs)

CALL RecordOneSiteOb(fileid, Obs, timeN)

Arguments:

fileid is INTEGER, INTENT(IN)

Ob is REAL(KIND=rKind), DIMENSION(`systemSize`) or COMPLEX(KIND=rKind), DIMENSION(`systemSize`)

timeN is REAL(KIND=rKind), OPTIONAL

B.9.10.14 RecordTwoSiteOb

Description:

`RecordTwoSiteOb(fileid, Obs)` writes the `systemSize`×`systemSize`-length array of two-site expectation values Ob onto the file whose UNIT is fileid. The procedure accepts REAL or COMPLEX Ob.

Type:

SUBROUTINE

Arguments:

fileid is INTEGER, INTENT(IN)

Ob is REAL(KIND=rKind), DIMENSION(`systemSize`,`systemSize`) or COMPLEX(KIND=rKind), DIMENSION(`systemSize`,`systemSize`)

B.9.11 propagation_module Procedure Listings

Contents:

B.9.11.1 TrotterStep

Description:

`TrotterStep(Udt, Gammas, Lambdas, totalTruncerr)` propagates the state defined by Gammas and Lambdas a single (real or imaginary) time step by the list of trotter propagators Udt. The Schmidt error defined in Eq. (B.84) is returned in totalTruncerr. This particular routine is an interface to the routines below which propagate for second or fifth order trotter expansions with open or periodic boundary conditions. The appropriate routine is chosen based on the global variables `BoundaryCond` and `trotterOrder`.

Type:

SUBROUTINE

Arguments:

Udt is TYPE(matrix), DIMENSION(:), POINTER

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

totalTruncerr is REAL(KIND=rKind), INTENT(OUT)

B.9.11.2 TrotterStep2ndOrder

Description:

`TrotterStep2ndOrder(Udt, Gammas, Lambdas, totalTruncerr)` propagates the state defined by Gammas and Lambdas a single (real or imaginary) time step by the list of trotter propagators Udt. The Schmidt error defined in Eq. (B.84) is returned in totalTruncerr. This particular routine uses the second order expansion with open boundary conditions (Sec. B.3.5.1).

Type:

SUBROUTINE

Arguments:

Udt is TYPE(matrix), DIMENSION(:), POINTER

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

totalTruncerr is REAL(KIND=rKind), INTENT(OUT)

B.9.11.3 TrotterStep2ndOrderPBC

Description:

`TrotterStep2ndOrderPBC(Udt, Gammas, Lambdas, totalTruncerr)` propagates the state defined by Gammas and Lambdas a single (real or imaginary) time step by the list of trotter propagators Udt. The Schmidt error defined in Eq. (B.84) is returned in totalTruncerr. This particular routine uses the second order expansion with periodic boundary conditions (Sec. B.3.5.3).

Type:

SUBROUTINE

Arguments:

Udt is TYPE(matrix), DIMENSION(:), POINTER

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

totalTruncerr is REAL(KIND=rKind), INTENT(OUT)

B.9.11.4 TrotterStep5thOrder

Description:

`TrotterStep5thOrder(Udt, Gammas, Lambdas, totalTruncerr)` propagates the state defined by Gammas and Lambdas a single (real or imaginary) time step by the list of trotter propagators Udt. The Schmidt error defined in Eq. (B.84) is returned in totalTruncerr. This particular routine uses the second order expansion with open boundary conditions (Sec. B.3.5.2).

Type:

SUBROUTINE

Arguments:

Udt is TYPE(matrix), DIMENSION(:), POINTER

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

totalTruncerr is REAL(KIND=rKind), INTENT(OUT)

B.9.11.5 TrotterStep5thOrderPBC

Description:

`TrotterStep5thOrderPBC(Udt, Gammas, Lambdas, totalTruncerr)` propagates the state defined by Gammas and Lambdas a single (real or imaginary) time step by the list of trotter propagators Udt. The Schmidt error defined in Eq. (B.84) is returned in totalTruncerr. This particular routine uses the fifth order expansion with periodic boundary conditions (Sec. B.3.5.3).

Type:

SUBROUTINE

Arguments:

Udt is TYPE(matrix), DIMENSION(:), POINTER

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

totalTruncerr is REAL(KIND=rKind), INTENT(OUT)

B.9.11.6 CanonicalFormAll

Description:

`CanonicalFormAll(Gammas,Lambdas)` puts all bipartite splittings into canonical form using the procedure `CanonicalForm` discussed in Sec. B.9.8.12. This amounts to re-orthogonalizing the Schmidt vectors, a necessary step after a non-unitary imaginary time step. See Sec. B.3.4.4 for more details.

Type:

SUBROUTINE

Arguments:

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

B.9.11.7 ImagTimeProp

Description:

`ImagTimeProp(H, GammasOuter, LambdasOuter, chiIn)` propagates the state represented by the Vidal decomposition GammasOuter and LambdasOuter in imaginary time under the Hamiltonian H. The routine performs two iterations of imaginary time propagation. The first uses the entanglement cutoff chiIn, and uses the convergence criterion $\left|\max_l \frac{\lambda_1^{[l]}(\tau+\Delta\delta\tau)-\lambda_1^{[l]}(\tau)}{\lambda_1^{[l]}(\tau)}\right| < \epsilon$, where $\lambda_1^{[l]}(\tau)$ represents $\lambda_1^{[l]}$ at imaginary time τ, Δ is `stepsForJudge` (Sec. B.8.1.10), $\delta\tau$ is `dtITP` (Sec. B.8.1.9), and ϵ is `convCriterion1` (Sec. B.8.1.12). The second iteration uses entanglement cutoff `chiMax` (Sec. B.8.1.8) and convergence criterion ϵ=`convCriterion2` (Sec. B.8.1.13). The OPTIONAL argument intDegFree specifies the presence of internal degrees of freedom. If `print_switch` is 1, the progress of the routine will be output to the screen every `stepsForJudge` imaginary time steps.

Type:

SUBROUTINE

Syntaxes:

CALL ImagTimeProp(H, GammasOuter, LambdasOuter, chiIn)

CALL ImagTimeProp(H, GammasOuter, LambdasOuter, chiIn, intDegFree)

Arguments:

H is TYPE(matrix), DIMENSION(:), POINTER

GammasOuter is TYPE(tensor), DIMENSION(:), POINTER

LambdasOuter is TYPE(vector), DIMENSION(:), POINTER

chiIn is INTEGER, INTENT(INOUT)

intDegFree is INTEGER, INTENT(IN), OPTIONAL

B.9.11.8 ImagTimePropSpin

Description:

`ImagTimePropSpin(H, GammasOuter, LambdasOuter, chiIn)` propagates the state represented by the Vidal decomposition GammasOuter and LambdasOuter in imaginary time under the (spin) Hamiltonian H. The routine performs two iterations of imaginary time propagation. The first uses the entanglement cutoff chiIn, and uses the convergence criterion $\left|\max_l \frac{\lambda_1^{[l]}(\tau+\Delta\delta\tau)-\lambda_1^{[l]}(\tau)}{\lambda_1^{[l]}(\tau)}\right| < \epsilon$, where $\lambda_1^{[l]}(\tau)$ represents $\lambda_1^{[l]}$ at imaginary time τ, Δ is `stepsForJudge` (Sec. B.8.1.10), $\delta\tau$ is `dtITP` (Sec. B.8.1.9), and ϵ is `convCriterion1` (Sec. B.8.1.12). The second iteration uses entanglement cutoff `chiMax` (Sec. B.8.1.8) and convergence criterion ϵ=`convCriterion2` (Sec. B.8.1.13). If `print_switch` is 1, the progress of the routine will be output to the screen every `stepsForJudge` imaginary time steps. The difference with `ImagTimeProp` is that this routine computes the average values of the spins instead of the average number.

Type:

SUBROUTINE

Arguments:

H is TYPE(matrix), DIMENSION(:), POINTER

GammasOuter is TYPE(tensor), DIMENSION(:), POINTER

LambdasOuter is TYPE(vector), DIMENSION(:), POINTER

chiIn is INTEGER, INTENT(INOUT)

B.9.11.9 TrotterStepNC

Description:

`TrotterStepNC(Udt, Gammas, Lambdas, LabelLeft, LabelRight, totalTruncerr, intDegFree)` propagates the state defined by Gammas and Lambdas a single (real or imaginary) time step by the list of trotter propagators Udt consistent with number conservation. The Schmidt error defined in Eq. (B.84) is returned in totalTruncerr. This particular routine is an interface to the routines below which propagate for second or fifth order trotter expansions with open or periodic boundary conditions. The appropriate routine is chosen based on the global variables `BoundaryCond` and `trotterOrder`. The OPTIONAL argument intDegFree specifies the presence of internal degrees of freedom.

Type:

SUBROUTINE

Arguments:

Udt is TYPE(matrix), DIMENSION(:), POINTER

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

LabelLeft and LabelRight are TYPE(vectorInt), DIMENSION(:), POINTER

totalTruncerr is REAL(KIND=rKind), INTENT(OUT)

intDegFree is INTEGER, INTENT(IN), OPTIONAL

B.9.11.10 TrotterStep2ndOrderNC

Description:

`TrotterStep2ndOrderNC(Udt, Gammas, Lambdas, LabelLeft, LabelRight, totalTruncerr, intDegFree)` propagates the state defined by Gammas and Lambdas a single (real or imaginary) time step by the list of trotter propagators Udt consistent with number conservation. The Schmidt error defined in Eq. (B.84) is returned in totalTruncerr. This particular routine uses the second order expansion with open boundary conditions (Sec. B.3.5.1). The OPTIONAL argument intDegFree specifies the presence of internal degrees of freedom.

Type:

SUBROUTINE

Arguments:

Udt is TYPE(matrix), DIMENSION(:), POINTER

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

LabelLeft and LabelRight are TYPE(vectorInt), DIMENSION(:), POINTER

totalTruncerr is REAL(KIND=rKind), INTENT(OUT)

intDegFree is INTEGER, INTENT(IN), OPTIONAL

B.9.11.11 TrotterStep2ndOrderPBCNC

Description:

`TrotterStep2ndOrderPBCNC(Udt, Gammas, Lambdas, LabelLeft, LabelRight, totalTruncerr, intDegFree)` propagates the state defined by Gammas and Lambdas a single (real or imaginary) time step by the list of trotter propagators Udt consistent with number conservation. The Schmidt error defined in Eq. (B.84) is returned in totalTruncerr. This particular routine uses the second order expansion with periodic boundary conditions (Sec. B.3.5.3). The OPTIONAL argument intDegFree specifies the presence of internal degrees of freedom.

Type:

SUBROUTINE

Arguments:

Udt is TYPE(matrix), DIMENSION(:), POINTER
Gammas is TYPE(tensor), DIMENSION(:), POINTER
Lambdas is TYPE(vector), DIMENSION(:), POINTER
LabelLeft and LabelRight are TYPE(vectorInt), DIMENSION(:), POINTER
totalTruncerr is REAL(KIND=rKind), INTENT(OUT)
intDegFree is INTEGER, INTENT(IN), OPTIONAL

B.9.11.12 TrotterStep5thOrderNC

Description:

`TrotterStep5thOrderNC(Udt, Gammas, Lambdas, LabelLeft, LabelRight,` `totalTruncerr, intDegFree)` propagates the state defined by Gammas and Lambdas a single (real or imaginary) time step by the list of trotter propagators Udt consistent with number conservation. The Schmidt error defined in Eq. (B.84) is returned in totalTruncerr. This particular routine uses the second order expansion with open boundary conditions (Sec. B.3.5.2). The OPTIONAL argument intDegFree specifies the presence of internal degrees of freedom.

Type:

SUBROUTINE

Arguments:

Udt is TYPE(matrix), DIMENSION(:), POINTER

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

LabelLeft and LabelRight are TYPE(vectorInt), DIMENSION(:), POINTER

totalTruncerr is REAL(KIND=rKind), INTENT(OUT)

intDegFree is INTEGER, INTENT(IN), OPTIONAL

B.9.11.13 TrotterStep5thOrderPBCNC

Description:

`TrotterStep5thOrderPBCNC(Udt, Gammas, Lambdas, LabelLeft, LabelRight, totalTruncerr, intDegFree)` propagates the state defined by Gammas and Lambdas a single (real or imaginary) time step by the list of trotter propagators Udt consistent with number conservation. The Schmidt error defined in Eq. (B.84) is returned in totalTruncerr. This particular routine uses the fifth order expansion with periodic boundary conditions (Sec. B.3.5.3). The OPTIONAL argument intDegFree specifies the presence of internal degrees of freedom.

Type:

SUBROUTINE

Arguments:

Udt is TYPE(matrix), DIMENSION(:), POINTER

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

LabelLeft and LabelRight are TYPE(vectorInt), DIMENSION(:), POINTER

totalTruncerr is REAL(KIND=rKind), INTENT(OUT)

intDegFree is INTEGER, INTENT(IN), OPTIONAL

B.9.11.14 CanonicalFormAllNC

Description:

`CanonicalFormAllNC(Gammas, Lambdas, LabelLeft, LabelRight,intDegFree)` puts all bipartite splittings into canonical form by acting with the two-site identity operator on all sites successively from left to right, and then successively from right to left. This amounts to re-orthogonalizing the Schmidt vectors, a necessary step after a non-unitary imaginary time step. The OPTIONAL argument intDegFree specifies the presence of internal degrees of freedom. See Sec. B.3.4.4 for more details.

Type:

SUBROUTINE

Arguments:

Gammas is TYPE(tensor), DIMENSION(:), POINTER

Lambdas is TYPE(vector), DIMENSION(:), POINTER

LabelLeft and LabelRight are TYPE(vectorInt), DIMENSION(:), POINTER

intDegFree is INTEGER, INTENT(IN), OPTIONAL

B.9.11.15 ImagTimePropNC

Description:

`ImagTimePropNC(H, GammasOuter, LambdasOuter, LabelLeftOuter, LabelRightOuter, chiIn)` propagates the state represented by the Vidal decomposition GammasOuter and LambdasOuter and lists number conserving vectors LabelLeftOuter and LabelRightOuter in imaginary time subject to number conservation under the Hamiltonian H. The routine performs two iterations of imaginary time propagation. The first uses the entanglement cutoff chiIn, and uses the convergence criterion $\left|\max_l \frac{\lambda_1^{[l]}(\tau+\Delta\delta\tau)-\lambda_1^{[l]}(\tau)}{\lambda_1^{[l]}(\tau)}\right| < \epsilon$, where $\lambda_1^{[l]}(\tau)$ represents $\lambda_1^{[l]}$ at imaginary time τ, Δ is `stepsForJudge` (Sec. B.8.1.10), $\delta\tau$ is `dtITP` (Sec. B.8.1.9), and ϵ is `convCriterion1` (Sec. B.8.1.12). The second iteration uses entanglement cutoff `chiMax` (Sec. B.8.1.8) and convergence criterion ϵ=`convCriterion2` (Sec. B.8.1.13). The OPTIONAL argument intDegFree specifies the presence of internal degrees of freedom. If `print_switch` is 1, the progress of the routine will be output to the screen every `stepsForJudge` imaginary time steps.

Type:

SUBROUTINE

Syntaxes:

CALL ImagTimePropNC(H, GammasOuter, LambdasOuter, LabelLeftOuter, LabelRightOuter, chiIn)

CALL ImagTimePropNC(H, GammasOuter, LambdasOuter, LabelLeftOuter, LabelRightOuter, chiIn, intDegFree)

Arguments:

H is TYPE(matrix), DIMENSION(:), POINTER

GammasOuter is TYPE(tensor), DIMENSION(:), POINTER

LambdasOuter is TYPE(vector), DIMENSION(:), POINTER

LabelLeftOuter and LabelRightOuter are TYPE(vectorInt), DIMENSION(:), POINTER

chiIn is INTEGER, INTENT(INOUT)

intDegFree is INTEGER, INTENT(IN), OPTIONAL

B.10 Acknowledgments

The effort which has become Open Source TEBD had quite a history before reaching its current state. It began in the group of Charles Clark and Barry Schneider at the Electron and Optical Physics Division of the National Institute of Standards and Technology. Jamie Williams wrote the first version in *Mathematica* in 2006 and ported some of it over to Fortran 90. Another member of the Clark group, Ippei Danshita, worked on the code, optimizing it for number conservation. Several members of Lincoln Carr's group, including Daniel Schirmer, continued code development at the National Institute of Standards and Technology and the Colorado School of Mines. We have done our best to acknowledge others' contributions by including their names in the author list of each part of the code they worked on, to the best of our knowledge. Currently, this work is supported by the National Science Foundation under the Physics at the Information Frontier program.

In preparation for this open source release, we have attempted to make the code as user-friendly as possible while still being high performance. The code and this document are both works in progress, so any comments are welcome. You can leave comments on the dedicated blog or email them to mwall@mines.edu. We are grateful to Stephen Clark, Ippei Danshita, Joseph Glick, Alexander Itin, Ryan Mishmash,

Dominik Muth, and Han Pu for comments and discussions regarding v1.0 of the Open Source package and TEBD in general.

B.11 References Cited

[1] D. Jaksch, C. Bruder, J. I. Cirac, C. W. Gardiner, and P. Zoller. Cold bosonic atoms in optical lattices. *Phys. Rev. Lett.*, 81(15):3108–3111, Oct 1998. doi: 10.1103/PhysRevLett.81.3108.

[2] Steven R. White. Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.*, 69(19):2863–2866, Nov 1992. doi: 10.1103/PhysRevLett.69.2863.

[3] Steven R. White. Density-matrix algorithms for quantum renormalization groups. *Phys. Rev. B*, 48(14):10345–10356, Oct 1993. doi: 10.1103/PhysRevB.48.10345.

[4] U. Schollwöck. The density-matrix renormalization group. *Reviews of Modern Physics*, 77(1):259, 2005. doi: 10.1103/RevModPhys.77.259. URL `http://link.aps.org/abstract/RMP/v77/p259`.

[5] Guifré Vidal. Efficient classical simulation of slightly entangled quantum computations. *Phys. Rev. Lett.*, 91(14):147902, Oct 2003. doi: 10.1103/PhysRevLett.91.147902.

[6] Guifré Vidal. Efficient simulation of one-dimensional quantum many-body systems. *Phys. Rev. Lett.*, 93(4):040502, Jul 2004. doi: 10.1103/PhysRevLett.93.040502.

[7] R. V. Mishmash. Quantum many-body dynamics of ultracold bosons in one-dimensional optical lattices: Theoretical aspects, simulation methods, and soliton formation and stability. Master's thesis, Colorado School of Mines, 2008. http://www.mines.edu/l̃carr/theses/mishmash_thesis_2008.pdf, http://inside.mines.edu/l̃carr/theses/mishmash_thesis_2008.pdf.

[8] Andrew John Daley. *Manipulation and Simulation of Cold Atoms in Optical Lattices*. PhD thesis, Leopold-Franzens-Universitat Innsbruck, 2005.

[9] Michael Philip Zwolak. *Dynamics and Simulation of open Quantum Systems*. PhD thesis, California Institute of Technology, 2008.

[10] G. Vidal, J. I. Latorre, E. Rico, and A. Kitaev. Entanglement in quantum critical phenomena. *Phys. Rev. Lett.*, 90(22):227902, Jun 2003. doi: 10.1103/PhysRevLett.90.227902.

[11] F. Verstraete and J. I. Cirac. Matrix product states represent ground states faithfully. *Physical Review B (Condensed Matter and Materials Physics)*, 73(9): 094423, 2006. doi: 10.1103/PhysRevB.73.094423. URL `http://link.aps.org/abstract/PRB/v73/e094423`.

[12] "Wikipedia entry for Cluster State". " http://en.wikipedia.org/wiki/cluster_state", 2009.

[13] G. Vidal. Classical simulation of infinite-size quantum lattice systems in one spatial dimension. *Physical Review Letters*, 98(7):070201, 2007. doi: 10.1103/PhysRevLett.98.070201. URL `http://link.aps.org/abstract/PRL/v98/e070201`.

[14] Y.-Y. Shi, L.-M. Duan, and G. Vidal. Classical simulation of quantum many-body systems with a tree tensor network. *Physical Review A (Atomic, Molecular, and Optical Physics)*, 74(2):022320, 2006. doi: 10.1103/PhysRevA.74.022320. URL `http://link.aps.org/abstract/PRA/v74/e022320`.

[15] Ippei Danshita and Pascal Naidon. Bose-hubbard ground state: Extended bogoliubov and variational methods compared with time-evolving block decimation. *Physical Review A (Atomic, Molecular, and Optical Physics)*, 79(4): 043601, 2009. doi: 10.1103/PhysRevA.79.043601. URL `http://link.aps.org/abstract/PRA/v79/e043601`.

[16] M. Suzuki. Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations. *Physics Letters A*, 146: 319–323, June 1990. doi: 10.1016/0375-9601(90)90962-N.

[17] Matthew P. A. Fisher, Peter B. Weichman, G. Grinstein, and Daniel S. Fisher. Boson localization and the superfluid-insulator transition. *Phys. Rev. B*, 40(1): 546–570, Jul 1989. doi: 10.1103/PhysRevB.40.546.

[18] Tin-Lun Ho. Spinor bose condensates in optical traps. *Phys. Rev. Lett.*, 81(4): 742–745, Jul 1998. doi: 10.1103/PhysRevLett.81.742.

[19] Giulio Racah. Theory of complex spectra. ii. *Phys. Rev.*, 62(9-10):438–462, Nov 1942. doi: 10.1103/PhysRev.62.438.

[20] A. Messiah. *Quantum Mechanics, Vol. 2.* Netherlands: North Holland, Amsterdam, 1962.

[21] I. P. McCulloch and M. Gulacsi. The non-abelian density matrix renormalization group algorithm. *Europhys. Lett.*, 57(6):852–858, January 2002.

[22] D. T. Pegg and S. M. Barnett. Phase properties of the quantized single-mode electromagnetic field. *Phys. Rev. A*, 39(4):1665–1675, Feb 1989. doi: 10.1103/PhysRevA.39.1665.

[23] G. K. Brennen. An observable measure of entanglement for pure states of multi-qubit systems. *Quant. Inf. Comp.*, 3:619–626, 2003.

[24] Howard Barnum, Emanuel Knill, Gerardo Ortiz, and Lorenza Viola. Generalizations of entanglement based on coherent states and convex sets. *Phys. Rev. A*, 68:032308, 2003.

[25] Howard Barnum, Emanuel Knill, Gerardo Ortiz, Rolando Somma, and Lorenza Viola. A subsystem-independent generalization of entanglement. *Phys. Rev. Lett.*, 92:107902, 2004.

APPENDIX C - DOCUMENTATION FOR ALPS V2.0 TEBD CODE

This appendix contains the documentation for the TEBD code included in the v2.0 release of the ALPS code, see Chapter 10. At the time of the writing of this thesis, this information was hosted on the ALPS wiki at http://alps.comp-phys.org. Sec. C.1 is documentation about the code in general, the background of the TEBD algorithm, and explanations of the parameters used as input to the code. The original website for this section is http://alps.comp-phys.org/mediawiki/index.php/Documentation:TEBD. Secs. C.2 and C.3 are tutorials on using the ALPS routines. The original websites are http://alps.comp-phys.org/mediawiki/index.php/ALPS_2_Tutorials:TEBD-01_bhquench and http://alps.comp-phys.org/mediawiki/index.php/ALPS_2_Tutorials:TEBD-02_kink, respectively.

C.1 Documentation: TEBD

C.1.1 Time-Evolving Block Decimation

The Time-Evolving Block Decimation (TEBD) algorithm is a method for simulating the time evolution of one-dimensional quantum lattice systems governed by a Hamiltonian with at most nearest neighbor interactions. It is closely related to the Density Matrix Renormalization Group (DMRG) method in that both methods operate on a class of states known as Matrix Product States (MPS). In addition to real time evolution, imaginary time evolution can also be used to find ground states. Essentially, TEBD consists of two parts: a canonical MPS representation of a many-body state, and a protocol for finding the MPS closest to a state which is acted upon by a two-site operator.

The particular implementation of TEBD used in ALPS simulates a series of global parameter quenches of the form $g(t) = g(t_i)+((t-t_i)/\tau)^p(g(t_f)-g(t_i))$. The timescale τ, power p, initial and final values $g(t_f)$ and $g(t_i)$, and Hamiltonian parameters g of each quench are all amenable to specification by the user. Additionally, because the TEBD method produces wavefunctions, a wide range of observables are available, including entropies, correlation functions, and overlaps between the state at different times.

C.1.2 References

G. Vidal
Efficient classical simulation of slightly entangled quantum computations
Phys. Rev. Lett. 91, 147902 (2003).

G. Vidal
Efficient simulation of one-dimensional quantum many-body systems
Phys. Rev. Lett 93, 040502 (2004).

A. J. Daley, C. Kollath, U. Schollwöck, and G. Vidal
Time-dependent density-matrix renormalization-group using adaptive effective Hilbert spaces
J. Stat. Mech. (2004) P04005.

C.1.3 TEBD-specific parameters

CHI_LIMIT

The maximum bond dimension of the MPS allowed during real time propagation. The default value is 50.

TRUNC_LIMIT

The maximum truncation error allowed for a specific two-site evolution. If the bond dimension corresponding to this truncation is greater than CHI_LIMIT, then CHI_LIMIT is chosen instead. The default value is 10^{-12}.

TAUS

The elements of this vector are the timescales τ of the global quenches.

GS

The elements of this vector are the Hamiltonian parameters g of the global quenches, given as character variables. Note that the elements of this vector may themselves be vectors, which corresponds to quenching several parameters at the same time. If this is so the corresponding elements of POWS, GIS, and GFS must also be vectors of the same length. Note that TAUS, NUMSTEPS, and STEPSFORSTORE will not be vectors, as the timescale, number of time steps, and number of steps between outputs are the same for each parameter being quenched.

POWS

The elements of this vector are the powers p of the global quenches.

GIS

The elements of this vector are the initial values of the Hamiltonian parameters g of the global quenches.

GFS

The elements of this vector are the final values of the Hamiltonian parameters g of the global quenches.

CONSERVED_QUANTUMNUMBERS

Quantum numbers conserved by the model of interest. For spin models 'Sz' can be conserved, and for particle models ’N’ can be conserved.

NUMSTEPS

The elements of this vector are the number of timesteps of the global quenches. This implicitly defines the time steps dt of the quenches.

STEPSFORSTORE

The elements of this vector are the number of timesteps between the calculation and output of observables.

INITIAL_STATE

The state used at $t = 0$, before real time propagation begins. Currently, only two values are supported: ’kink’, which produces a specific initial state to be discussed further in tutorial 2a, and ’ground’, which calculates the ground state of a specified initial hamiltonian via imaginary time propagation. The default value is ’ground’. See the tutorials for examples.

ITP_CHIS

The elements of this vector are the maximum bond dimensions used in iterations of imaginary time propagation to find the group state. It is only referenced if INITIAL_STATE is ’ground’.

ITP_DTS

The elements of this vector are the time steps used in iterations of imaginary time propagation to find the group state. It is only referenced if INITIAL_STATE is ’ground’.

ITP_CONVS

The elements of this vector are the convergence parameters used in iterations of imaginary time propagation to find the group state. An iteration of imaginary time propagation exits if the maximal difference between singular values at some time interval is less than the convergence parameter. It is only referenced if INITIAL_STATE is 'ground'.

SIMID

An optional integer input which differentiates a series of simulations and can simplify plotting commands.

NUM_THREADS

The number of OpenMP threads used.

VERBOSE

If set to 'true' then the code will output, time values, truncation errors, and other running messages. The default value is 'false'.

C.2 Tutorials: TEBD-01 bhquench

C.2.1 The Hardcore Boson Model

In this first tutorial we investigate the behavior of the hardcore boson model

$$H = -t\sum_{i=1}^{L-1}(b_i^{\dagger}b_{i+1} + b_i b_{i+1}^{\dagger}) + V\sum_{i=1}^{L-1} n_i n_{i+1} \tag{C.1}$$

as the parameter V is changed in time. It is well known that for large V/t the ground state of the hardcore boson model at half filling is a charge-density wave (CDW) insulator while for small V/t the ground state is a superfluid (SF). It is interesting to consider what happens to the system if we begin in one phase and then dynamically change, or "quench", one of the Hamiltonian parameters t or V such that we are in the other phase. As a simple first foray into the rich physics of quenches, we will consider quenching from one phase to the other and then back into the original phase. A particularly stringent criterion for adiabaticity of such a quench is how close the final state is to the initial state, i.e.

$$L(t;\gamma) \equiv |\langle\psi\,(t)\,|\psi\,(0)\rangle|^2 \tag{C.2}$$

which we call the Loschmidt Echo. Note that the t in this expression is the time and not the hopping parameter t. The parameter γ is meant to convey that this quantity in general depends on the manner in which the system is quenched.

The general structure of a quench in the ALPS TEBD routines is given by the parameterization

$$g(t) = g(t_i) + ((t - t_i)/\tau)^{\,p}(g(t_f) - g(t_i)) \tag{C.3}$$

where g is some Hamiltonian parameter. In the present case we will take g to be the interaction parameter V. We will begin our system in the CDW regime with $V/t = 10$, quench to the SF regime where $V/t = 0$, and then quench back to the CDW regime with $V/t = 10$. In the three parts of this tutorial we will investigate a)the effects of the timescale τ on the Loschmidt echo during a linear quench, b) the effects of "holding" the system in the SF phase for a time τ_{hold} before returning to the CDW phase, and c) the effects of changing the power p of the quench function.

C.2.2 Linear Quench

First, we will investigate the effects of the quench rate τ on the adiabaticity of a linear quench from the CDW to the SF phase and back.

C.2.2.1 Preparing and running the simulation using Python

To set up and run the simulation in Python we use the script tutorial1a.py. The first parts of this script imports the required modules and then prepares the input files as a list of Python dictionaries:

```
import pyalps
import matplotlib.pyplot as plt
import pyalps.plot

parms=[]
count=0
for A in [5.0, 10.0, 15.0, 25.0, 50.0]:
        count+=1
        parms.append({
                    'L'                           : 10,
                    'MODEL'                       : 'hardcore_boson',
                    'CONSERVED_QUANTUMNUMBERS'    : 'N',
                    'N' : 5,
                    't'                           : 1.0,
                    'V'                           : 10.0,
                    'ITP_CHIS' : [20, 30, 35],
                    'ITP_DTS' : [0.05, 0.05,0.025],
                    'ITP_CONVS' : [1E-8, 1E-8, 1E-9],
                    'INITIAL_STATE' : 'ground',
                    'CHI_LIMIT' : 40,
                    'TRUNC_LIMIT' : 1E-12,
                    'NUM_THREADS' : 1,
                    'TAUS' : [A,  A],
                    'POWS' : [1.0, 1.0],
                    'GS' : ['V',  'V'],
                    'GIS' : [10.0,  0.0],
                    'GFS' : [0.0,  10.0],
                    'NUMSTEPS' : [500,  500],
                    'STEPSFORSTORE' : [5, 3],
                    'SIMID' : count
                  })
```

Let's go through the TEBD-specific parameters in more detail (see the TEBD documentation, Sec. C.1, for a list of all such parameters). The parameter INITIAL_STATE is set to ground, which means that we begin from the ground state of our Hamiltonian with user-specified parameters. The parameters t and V specify that the initial

Hamiltonian parameters $t = 1$ and $V = 10$ are used to find the ground state. In order to find the ground state, TEBD performs evolution in imaginary time. We refer to this step as ITP, and so all parameters containing ITP deal with the ground state properties. The vectors ITP_CHIS, ITP_DTS, and ITP_CONVS are the entanglement cutoff parameters, time steps, and convergence criteria for successive applications of imaginary time propagation. These constitute the main convergence parameters for TEBD, and convergence should always be carefully checked in each parameter. For now, don't worry too much about their actual values, we'll see how errors are controlled in the next set of tutorials.

Now we turn to the real-time propagation parameters. We wish to perform a series of two quenches. First we want to quench the parameter V linearly in time from its initial value 10 to 0. Comparing with the general form of a quench $g(t) = g(t_i) + ((t - t_i)/\tau)^p (g(t_f) - g(t_i))$ we see that this corresponds to $g = V$, $g(t_i) = 10$, $g(t_f) = 0$, $p = 1$, and τ is the free parameter whose effects are to be investigated. Looking at the parameter list, we see that the first elements of the vectors GS, GIS, GFS, and POWS correspond to g, $g(t_i)$, $g(t_f)$, and p, respectively. The first element of the vector TAUS is looped over using the variable A, which means that we will perform a series of simulations with τ=5, 10, 15, 25, and 50. The second quench is essentially the reverse of the first, with $g = V$, $g(t_i) = 0$, $g(t_f) = 10$, $p = 1$, and τ the same as the first. Comparing with the parameters list, we see that this corresponds to the second elements of the vectors GS, GIS, etc. as above.

Time evolution is simulated by breaking the full propagator approximately into a series of operations which act only on two neighboring sites at a time. The error in using this approximate propagator is second order in the "infinitesimal" timestep dt. TEBD gives a protocol for updating the canonical form of our state after such a two-site operation has been applied. The error in this procedure is controlled by CHI_LIMIT, which is directly related to the amount of spatial entanglement, and TRUNC_LIMIT, which is akin to the TRUNCATION_ERROR in the DMRG routines. The parameter vector NUMSTEPS specifies how many timesteps are taken in performing each quench, which together with τ implicitly defines the timestep dt. The overall error is a nontrivial function of CHI_LIMIT, TRUNC_LIMIT, and NUMSTEPS which will be investigated in the next set of tutorials, so we won't worry about the choice of these much for now. Finally, STEPSFORSTORE determines how many time steps are taken before observables are computed and stored and SIMID is an integer differentiating the simulations with different τ.

We now move on to the actual computation. The lines:

```
baseName='tutorial_1a'
#write output files
nmlnameList=pyalps.writeTEBDfiles(parms, baseName)
#run the application
res=pyalps.runTEBD(nmlnameList)
```

prepare the input files for the TEBD routines and run the simulations for the range of τ specified in the parameters. We now load the Loschmidt Echo and interaction parameter U as functions of time via:

```
#Load the loschmidt echo and V
LEdata=pyalps.load.loadTimeEvolution(pyalps.getResultFiles(
    prefix='tutorial_1a'), measurements=['Loschmidt␣Echo', 'V'])
```

Finally, we plot the collected data using:

```
LE=pyalps.collectXY(LEdata, x='Time', y='Loschmidt␣Echo',foreach
    =['SIMID'])
for q in LE:
        q.props['label']=r'$\tau=$'+str(q.props['TAUS'][0])

plt.figure()
pyalps.plot.plot(LE)
plt.xlabel('Time␣$t$')
plt.ylabel('Loschmidt␣Echo␣$|<␣\psi(0)|\psi(t)␣>␣|^2$')
plt.title('Loschmidt␣Echo␣vs.␣Time')
plt.legend(loc='lower␣right')
```

```
Ufig=pyalps.collectXY(LEdata, x='Time', y='V',foreach=['SIMID'])
for q in Ufig:
        q.props['label']=r'$\tau=$'+str(q.props['TAUS'][0])

plt.figure()
pyalps.plot.plot(Ufig)
plt.xlabel('Time $t$')
plt.ylabel('V')
plt.title('Interaction parameter $V$ vs. Time')
plt.legend(loc='lower right')
plt.show()
```

C.2.2.2 Preparing and running the simulation using Vistrails

To run the simulation in Vistrails open the file tutorial1a.vt and look at the workflow labeled "tutorial1a". Click on "Execute" to prepare the input file, run the simulation and create the output figure.

C.2.2.3 Questions

- How does the behavior of the overlap change as the quench rate decreases?
- Roughly how slowly do you have to perform the quench in order for it to be adiabatic?
- Is it easier or harder for a larger system to be adiabatic? Why?
- Are these properties changed depending on whether the intermediate phase is gapped or not? One can test this by changing from the hardcore boson model to the (softcore) boson Hubbard model, and then quenching from the Mott-Insulating (MI) phase at large U/t and unit filling to the CDW phase with large V. As you quench from the Mott insulating to the CDW phase and back, how difficult is it to be adiabatic?

C.2.3 Linear Quench with hold

In this section we will investigate the effects of "holding" the system in the SF phase for a time τ_{hold} before quenching back to the CDW phase.

C.2.3.1 Preparing and running the simulation using Python

To set up and run the simulation in Python we use the script tutorial1b.py. The first parts of this script imports the required modules and then prepares the input files as a list of Python dictionaries:

```
import pyalps
import matplotlib.pyplot as plt
import pyalps.plot

#prepare the input parameters
parms=[]
count=0
for A in [5.0, 10.0, 15.0, 25.0, 50.0]:
        count+=1
        parms.append({
                    'L'                            : 10,
                    'MODEL'                        : 'hardcore boson',
                    'CONSERVED_QUANTUMNUMBERS'     : 'N',
                    'N' : 5,
                    't'                            : 1.0,
                    'V'                            : 10.0,
                    'ITP_CHIS' : [20, 30, 35],
                    'ITP_DTS' : [0.05, 0.05,0.025],
                    'ITP_CONVS' : [1E-8, 1E-8, 1E-9],
                    'INITIAL_STATE' : 'ground',
                    'CHI_LIMIT' : 80,
                    'TRUNC_LIMIT' : 1E-12,
                    'NUM_THREADS' : 1,
                    'TAUS' : [10.0, A, 10.0],
                    'POWS' : [1.0, 0.0,1.0],
                    'GS' : ['V', 'V', 'V'],
                    'GIS' : [10.0,0.0, 0.0],
                    'GFS' : [0.0, 0.0, 10.0],
                    'NUMSTEPS' : [500, int(A/0.05), 500],
                    'STEPSFORSTORE' : [5,5, 3],
                    'SIMID' : count
                })
```

Note that in this case we have three quenches as GS, GIS, etc. are all vectors of length three. The second quench keeps the Hamiltonian parameters fixed at $t = 1$, $V = 0$ for a time τ_{hold} before quenching back. We write the input files, run the simulations, get outputs, and plot as above:

```
baseName='tutorial_1b'
#write output files
nmlnameList=pyalps.writeTEBDfiles(parms, baseName)
#run the application
res=pyalps.runTEBD(nmlnameList)

#Load the loschmidt echo and U
LEdata=pyalps.load.loadTimeEvolution(pyalps.getResultFiles(
    prefix='tutorial_1b'), measurements=['Loschmidt␣Echo', 'V'])

LE=pyalps.collectXY(LEdata, x='Time', y='Loschmidt␣Echo',foreach
    =['SIMID'])
for q in LE:
        q.props['label']=r'$\tau_{\mathrm{hold}}=$'+str(q.props['
            TAUS'][1])
plt.figure()
pyalps.plot.plot(LE)
plt.xlabel('Time␣$t$')
plt.ylabel('Loschmidt␣Echo␣$|<␣\psi(0)|\psi(t)␣>␣|^2$')
plt.title('Loschmidt␣Echo␣vs.␣Time')
plt.legend(loc='lower␣right')

Ufig=pyalps.collectXY(LEdata, x='Time', y='V',foreach=['SIMID'])
for q in Ufig:
        q.props['label']=r'$\tau_{\mathrm{hold}}=$'+str(q.props['
            TAUS'][1])
plt.figure()
pyalps.plot.plot(Ufig)
plt.xlabel('Time␣$t$')
plt.ylabel('V')
plt.title('Interaction␣parameter␣$V$␣vs.␣Time')
plt.legend()
plt.show()
```

C.2.3.2 Preparing and running the simulation using Vistrails

To run the simulation in Vistrails open the file tutorial1b.vt and look at the workflow labeled ”tutorial1b”. Click on ”Execute” to prepare the input file, run the simulation and create the output figure

C.2.3.3 Questions

- How does the behavior of the overlap change as the hold time increases?

- Is this behavior monotonic in the hold time? Why or why not?

C.2.4 Nonlinear Quenches

In this section we will investigate the effects of varying the power of the quench away from being linear.

C.2.4.1 Preparing and running the simulation using Python

To set up and run the simulation in Python we use the script tutorial1c.py. The first parts of this script imports the required modules and then prepares the input files as a list of Python dictionaries:

```
import pyalps
 import matplotlib.pyplot as plt
 import pyalps.plot

 #prepare the input parameters
 parms=[]
 count=0
 for A in [1.0, 1.5, 2.0, 2.5, 3.0]:
         count+=1
         parms.append({
                   'L'                          : 10,
                   'MODEL'                      : 'hardcore␣boson',
                   'CONSERVED_QUANTUMNUMBERS'   : 'N',
                   'N' : 5,
                   't'                          : 1.0,
                   'V'                          : 10.0,
                   'ITP_CHIS' : [20, 30, 35],
                   'ITP_DTS' : [0.05, 0.05,0.025],
                   'ITP_CONVS' : [1E-8, 1E-8, 1E-9],
                   'INITIAL_STATE' : 'ground',
                   'CHI_LIMIT' : 40,
                   'TRUNC_LIMIT' : 1E-12,
                   'NUM_THREADS' : 1,
                   'TAUS' : [10.0,  10.0],
                   'POWS' : [1.0, A],
                   'GS' : ['V',  'V'],
                   'GIS' : [10.0,  0.0],
                   'GFS' : [0.0,  10.0],
                   'NUMSTEPS' : [1000,  1000],
                   'STEPSFORSTORE' : [10, 5],
                   'SIMID' : count
```

```
                    })
```

We then write the input files, run the simulations, get outputs, and plot as above:

```
baseName='tutorial_1c'
 #write output files
 nmlnameList=pyalps.writeTEBDfiles(parms, baseName)
 #run the application
 res=pyalps.runTEBD(nmlnameList)

 #Load the loschmidt echo and U
 LEdata=pyalps.load.loadTimeEvolution(pyalps.getResultFiles(
    prefix='tutorial_1c'), measurements=['V', 'Loschmidt Echo'])

 LE=pyalps.collectXY(LEdata, x='Time', y='Loschmidt Echo',foreach
    =['SIMID'])
 for q in LE:
         q.props['label']=r'$\tau=$'+str(q.props['POWS'][1])
 plt.figure()
 pyalps.plot.plot(LE)
 plt.xlabel('Time $t$')
 plt.ylabel('Loschmidt Echo $|< \psi(0)|\psi(t) > |^2$')
 plt.title('Loschmidt Echo vs. Time ')
 plt.legend(loc='lower left')

 Ufig=pyalps.collectXY(LEdata, x='Time', y='V',foreach=['SIMID'])
 for q in Ufig:
         q.props['label']=r'$\tau=$'+str(q.props['POWS'][1])
 plt.figure()
 pyalps.plot.plot(Ufig)
 plt.xlabel('Time $t$')
 plt.ylabel('U')
 plt.title('Interaction parameter $V$ vs. Time')
 plt.legend(loc='lower left')
 plt.show()
```

C.2.4.2 Preparing and running the simulation using Vistrails

To run the simulation in Vistrails open the file tutorial1c.vt and look at the workflow labeled "tutorial1c". Click on "Execute" to prepare the input file, run the simulation and create the output figure

C.2.4.3 Questions

- How does the behavior of the overlap change as the power changes?
- Again change from the hardcore boson model to the boson Hubbard model and investigate the dynamics of the MI-CDW transition, this time with a nonlinear quench. Is the behavior different from that of a linear quench?
- The present example uses an asymmetric quench which is linear one one side and nonlinear on the other. How is the behavior changed if you make both quenches nonlinear?

C.3 Tutorials: TEBD-02 kink

C.3.1 Evolution of a domain Wall

In this tutorial we will study the time evolution of a S=1/2 spin chain prepared in a nonequilibrium state. The particular state that we choose is that with all spins to the left of the chain center ”down” and all of those to the right of the center ”up,” $|\downarrow\downarrow \dots \downarrow\uparrow \dots \uparrow\uparrow\rangle$. This state can be chosen as the initial state by setting INITIAL_STATE to be ’kink’. Some exact results are known regarding the evolution of this state under the 1D XX model, which allows for a detailed study of the errors present in TEBD.

C.3.2 Exact Solution for the case of the XX model

The time evolution of the kink initial state under the XX model was solved exactly in Phys. Rev. E **59**, 4912 (1999) by a Jordan-Wigner transformation to free fermions. It was found that the expectation value of the magnetization at any site as a function of time can be represented as a sum of Bessel functions, and the magnetization in the limit of long times and large distances from the initial domain wall approaches a scaling form in the variable n/t, where n is the distance from the center and t the time. Explicitly, we have

$$M(n,t) = -\frac{1}{2}\sum_{i=1-n}^{n-1} j_i^2\left(t\right) \ , \tag{C.4}$$

$$\lim_{n\to\infty}\lim_{t\to\infty} M(n,t) \to \phi\left(n/t\right) = -\frac{1}{\pi}\arcsin\left(n/t\right) \ , \tag{C.5}$$

where $M(n,t)$ is the magnetization a distance n from the center and $j_i(t)$ is the Bessel function of order i. In the first part of this tutorial we demonstrate these two results.

C.3.2.1 Preparing and running the simulation using Python

To set up and run the simulation in Python we use the script tutorial2a.py. The first parts of this script imports the required modules and prepares the input files as a list of Python dictionaries:

```python
import pyalps
import matplotlib.pyplot as plt
import pyalps.plot
import numpy as np
import copy
import math
import scipy.special

#prepare the input parameters
parms = [{
            'L'                         : 50,
            'MODEL'                     : 'spin',
            'local_S'                   : 0.5,
            'CONSERVED_QUANTUMNUMBERS'  : 'Sz',
            'Jxy'                         : 1,
            'INITIAL_STATE' : 'kink',
            'CHI_LIMIT' : 40,
            'TRUNC_LIMIT' : 1E-12,
            'NUM_THREADS' : 1,
            'TAUS' : [20.0],
            'POWS' : [0.0],
            'GS' : ['H'],
            'GIS' : [0.0],
            'GFS' : [0.0],
            'NUMSTEPS' : [500],
            'STEPSFORSTORE' : [2]
         }]
```

The math and scipy.special modules are required to generate the special functions needed to compare with the exact solution. Note that we have chosen POWS to be zero, which corresponds to no quenching at all. Thus, the values of GS, GIS, and GFS are arbitrary, and TAUS and NUMSTEPS give us the total simulation time and the number of time steps, respectively. We write the input files, run the simulation, and get the output as usual:

```
baseName='tutorial_2a'
 nmlname=pyalps.writeTEBDfiles(parms, baseName)
 res=pyalps.runTEBD(nmlname)

 #Get the results of the simulation
 Data=pyalps.load.loadTimeEvolution(pyalps.getResultFiles(prefix=
     'tutorial_2a'), measurements=['Local Magnetization'])
```

We now must postprocess the raw output to compare with the exact solution. To do this we first define empty arrays to hold the postprocessed data

```
#define a dataset numericalSolution to contain the numerical
    result
 numericalResult=[]
 #define a dataset exactSolution to contain the exact solution
 exactResult=[]
 #define a dataset scalingForm to contain the scaling form
 scalingForm=[]
```

we then calculate the exact result from the time data, and use the computed values of the magnetization at each site to compare with the exact solution.

```
  #Compute the exact result M(n,t)=<S_n^z>=-(1/2)*sum_{i=1-n}^{n
      -1} j_i(t)^2, where
 # j_i(t) is the Bessel function of order i and compare to the
     numerically obtained result
 for q in Data:
         syssize=q[0].props['L']
         #Assign a label 'Distance' denoting the distance from
             the center n (only do the first two sites
         #to avoid cluttering the plot)
         for n in range(1,3):
                 #Create copies of the data for postprocessing
                 numericalCopy=copy.deepcopy(q)
```

```
                exactCopy=copy.deepcopy(q)

                numericalCopy[0].props['Distance']=n
                numericalCopy[0].props['SIMID']='Numerical␣at␣n=
                    '+str(n)
                exactCopy[0].props['Distance']=n
                exactCopy[0].props['SIMID']='Exact␣at␣n='+str(n)

                #compute the exact result of the manetization n
                    sites from the center
                loc=0.0
                for i in range(1-n,n):
                        loc-=0.5*scipy.special.jn(i,q[0].props['
                            Time'])*scipy.special.jn(i,q[0].props
                            ['Time'])
                exactCopy[0].y=[loc]
                #add to the exact dataset
                exactResult.extend(exactCopy)

                #get the numerical result of the magnetization n
                     sites from the center
                numericalCopy[0].y=[q[0].y[syssize/2+n-1]]
                #add to the numerical dataset
                numericalResult.extend(numericalCopy)
```

Next, we calculate the exact scaling function, and then compute magnetization as a function of the scaling variable n/t to compare with the exact solution

```
 #compute the scaling form
# \phi(n/t)=-(1/pi)*arcsin(n/t) that M(n,t) approaches as n->
    infinity and t->infinity
# and compare it with the numerically computed values of M(n/t)
for q in Data:
        syssize=q[0].props['L']
        #Assign a label 'Distance' denoting the distance from
            the center n (only do the first few sites
        #to avoid cluttering the plot)
        for n in range(0,5):
                #Create a copy of the data for postprocessing
                scalingCopy=copy.deepcopy(q)
                scalingCopy[0].props['Distance']=n

                #The first distance contains the exact scaling
                    form \phi(n/t)=-(1/pi)*arcsin(n/t)
                if n==0:
```

```
                         scalingCopy[0].props['Time']=1.0/
                            scalingCopy[0].props['Time']
                         scalingCopy[0].y=[-(1.0/3.1415926)*math.
                            asin(min(scalingCopy[0].props['Time'
                            ],1.0))]
                        scalingCopy[0].props['SIMID']='Exact'

                #The other distances contain the numerical data
                   as a function of the scaling variable M(n/t)
                else:
                         scalingCopy[0].props['Time']=n/
                            scalingCopy[0].props['Time']
                         scalingCopy[0].y=[scalingCopy[0].y[
                            syssize/2+n-1] ]
                        scalingCopy[0].props['SIMID']='Numerical␣
                           at␣n='+str(n)
                #add to the scaling dataset
                scalingForm.extend(scalingCopy)
```

Finally, we plot the exact and numerical results for comparison.

```
#Plot the numerical and exact magnetization for comparison
exactMag=pyalps.collectXY(exactResult, x='Time', y='Local␣
   Magnetization',foreach=['SIMID'])
for q in exactMag:
        q.props['label']=q.props['SIMID']
numericalMag=pyalps.collectXY(numericalResult, x='Time', y='
   Local␣Magnetization',foreach=['SIMID'])
for q in numericalMag:
        q.props['label']=q.props['SIMID']

plt.figure()
pyalps.plot.plot([exactMag, numericalMag])
plt.xlabel('Time␣$t$')
plt.ylabel('Magnetization')
plt.legend(loc='lower␣right')
plt.title('Magnetization␣vs.␣time')

#Plot the scaling form with the numerical data for comparison
Scal=pyalps.collectXY(scalingForm, x='Time', y='Local␣
   Magnetization', foreach=['SIMID'])
for q in Scal:
        q.props['label']=q.props['SIMID']

plt.figure()
pyalps.plot.plot(Scal)
plt.xlabel('Scaling␣variable␣$n/t$')
```

```
plt.ylabel('Magnetization$(n,t)$')
plt.legend()
plt.xlim(0,1.5)
plt.title('Magnetization␣scaling␣function;␣numerical␣and␣exact␣
    results')
plt.show()
```

We see that the magnetization agrees very well to visual accuracy, and approaches the exact scaling form in the relevant limit.

C.3.2.2 Preparing and running the simulation using Vistrails

To run the simulation in Vistrails open the file tutorial2a.vt and look at the workflow labeled "tutorial2a". Click on "Execute" to prepare the input file, run the simulation and create the output figure

C.3.3 Error analysis of TEBD 1:Time step error

We now use the exact solution to compute the error in a TEBD simulation as a function of time. We first investigate the effects of changing the "infinitesimal" time step dt.

C.3.3.1 Preparing and running the simulation using Python

To set up and run the simulation in Python we use the script tutorial2b.py. The first parts of this script imports the required modules and prepares the input files as a list of Python dictionaries:

```
import pyalps
import matplotlib.pyplot as plt
import pyalps.plot
import numpy as np
import math
import scipy.special

#prepare the input parameters
parms=[]
count=0
for nsteps in [100, 250, 500, 750, 1000]:
        count+=1
```

```
    parms.append({
            'L'                          : 50,
            'MODEL'                      : 'spin',
            'local_S'                    : 0.5,
            'CONSERVED_QUANTUMNUMBERS'   : 'Sz',
            'Jxy'                          : 1,
            'INITIAL_STATE' : 'kink',
            'CHI_LIMIT' : 20,
            'TRUNC_LIMIT' : 1E-12,
            'NUM_THREADS' : 1,
            'TAUS' : [20.0],
            'POWS' : [0.0],
            'GS' : ['H'],
            'GIS' : [0.0],
            'GFS' : [0.0],
            'NUMSTEPS' : [nsteps],
            'STEPSFORSTORE' : [int(math.floor(nsteps/100))
               ],
            'SIMID': count
        })
```

By changing the parameter NUMSTEPS we implicitly change the time step, since the total evolution time TAU is fixed. We now write the input files, run the simulations, and collect data:

```
baseName='tutorial_2b_'
nmlnameList=pyalps.writeTEBDfiles(parms, baseName)
res=pyalps.runTEBD(nmlnameList)

#Get magnetization data
Magdata=pyalps.load.loadTimeEvolution( pyalps.getResultFiles(
    prefix='tutorial_2b'), measurements=['Local Magnetization'])
```

We now calculate the exact result from the time data, and then calculate the difference between the numerical and the exact result for the magnetization

```
#Postprocessing-get the exact result for comparison
for q in Magdata:
        syssize=q[0].props['L']
        #Get the exact result of M(1,t)=-(1/2)*(j_0(t)^2), where
             j_0(t) is the 0^{th} order
        # bessel function and M(1,t) is the magnetization one
            site to the right of the chain center
```

```
        loc=-0.5*scipy.special.jn(0,q[0].props['Time'])*scipy.
            special.jn(0,q[0].props['Time'])
        #Get the difference between the computed and exact
            results
        q[0].y=[abs(q[0].y[syssize/2+1-1]-loc)]
```

Finally, we plot this magnetization error:

```
#Plot the Error in the magnetization one site to the right of
    the chain center
Mag=pyalps.collectXY(Magdata, x='Time', y='Local␣Magnetization',
     foreach=['SIMID'])
for q in Mag:
        dt=round(q.props['TAUS']/q.props['NUMSTEPS'],3)
        q.props['label']='dt='+str(dt)

plt.figure()
pyalps.plot.plot(Mag)
plt.xlabel('Time␣$t$')
plt.yscale('log')
plt.ylabel('Magnetization␣Error')
plt.title('Error␣in␣the␣magnetization␣vs.␣time')
plt.legend(loc='lower␣left')
plt.show()
```

We see that, for short times, the errors are roughly proportional to dt^2, reflecting the contribution to the error from the trotter breakup of our exponential. At long times, however, the simulations with the smallest dt have errors which become larger than those with larger dt, and eventually the errors blow up! We will have more to say about this behavior in the next section.

C.3.3.2 Preparing and running the simulation using Vistrails

To run the simulation in Vistrails open the file tutorial2b.vt and look at the workflow labeled "tutorial2b". Click on "Execute" to prepare the input file, run the simulation and create the output figure

C.3.4 Error analysis of TEBD 2:Entanglement cutoff error

We now investigate the effects of changing the entanglement cutoff parameter χ on the errors in the magnetization.

C.3.4.1 Preparing and running the simulation using Python

To set up and run the simulation in Python we use the script tutorial2c.py. The first parts of this script imports the required modules and prepares the input files as a list of Python dictionaries:

```
import pyalps
import matplotlib.pyplot as plt
import pyalps.plot
import math
import scipy.special

#prepare the input parameters
parms=[]
count=0
for chi in [10, 20, 30, 40]:
        count+=1
        parms.append({
                  'L'                         : 50,
                  'MODEL'                     : 'spin',
                  'local_S'                   : 0.5,
                  'CONSERVED_QUANTUMNUMBERS'  : 'Sz',
                  'Jxy'                         : 1,
                  'INITIAL_STATE' : 'kink',
                  'CHI_LIMIT' : chi,
                  'TRUNC_LIMIT' : 1E-12,
                  'NUM_THREADS' : 1,
                  'TAUS' : [20.0],
                  'POWS' : [0.0],
                  'GS' : ['H'],
                  'GIS' : [0.0],
                  'GFS' : [0.0],
                  'NUMSTEPS' : [500],
                  'STEPSFORSTORE' : [5],
                  'SIMID': count
                })
```

We now write the input files, run the simulations, collect data, and compute the errors as above

```
baseName='tutorial_2c_'
nmlnameList=pyalps.writeTEBDfiles(parms, baseName)
res=pyalps.runTEBD(nmlnameList)
```

```
#Get magnetization data
Magdata=pyalps.load.loadTimeEvolution( pyalps.getResultFiles(
   prefix='tutorial_2c'), measurements=['Local␣Magnetization'])

#Postprocessing-get the exact result for comparison
for q in Magdata:
        syssize=q[0].props['L']
        #Get the exact result of M(1,t)=-(1/2)*(j_0(t)^2), where
            j_0(t) is the 0^{th} order
        # bessel function and M(1,t) is the magnetization one
           site to the right of the chain center
        loc=-0.5*scipy.special.jn(0,q[0].props['Time'])*scipy.
           special.jn(0,q[0].props['Time'])
        #Get the difference between the computed and exact
           results
        q[0].y=[abs(q[0].y[syssize/2+1-1]-loc)]
```

Finally, we plot the magnetization error

```
#Plot the Error in the magnetization one site to the right of
   the chain center
Mag=pyalps.collectXY(Magdata, x='Time', y='Local␣Magnetization',
    foreach=['SIMID'])
for q in Mag:
       q.props['label']='$\chi$='+str(q.props['CHI_LIMIT'])
plt.figure()
pyalps.plot.plot(Mag)
plt.xlabel('Time␣$t$')
plt.yscale('log')
plt.ylabel('Magnetization␣Error')
plt.title('Error␣in␣the␣magnetization␣vs.␣time')
plt.legend(loc='lower␣left')
plt.show()
```

We see that, for short times, the errors are roughly proportional to dt^2, again reflecting the contribution to the error from the trotter breakup of our exponential. As time increases, however, a cascade of diverging errors ensues. First the simulation with $\chi = 10$ diverges around $t = 5$, then the simulation with $\chi = 20$ diverges around $t = 9$ and so on. This breakdown is due to the fact that the protocol for finding the matrix product state which best approximates the time-evolved state is approximate when the state becomes highly entangled. This approximation involves a renormalization of the wavefunction, and so the errors accumulate roughly exponentially in time.

This exponential growth of errors also accounts for the failure of the simulations with smaller dt. As dt becomes smaller we must apply the approximate propagation scheme more to reach the same fixed final time, and this means more accumulation of the exponentially growing truncation error. Thus, we must strike a delicate balance between the error incurred by increasing the time step and the error incurred by taking more time steps. All results should be carefully checked for convergence in both dt and χ.

C.3.4.2 Preparing and running the simulation using Vistrails

To run the simulation in Vistrails open the file tutorial2c.vt and look at the workflow labeled "tutorial2c". Click on "Execute" to prepare the input file, run the simulation and create the output figure

C.3.5 Solution in the case of the XXZ model

We saw from the exact solution that the magnetization profile had a well defined front which expanded ballistically with velocity $v = 1$. The XX model has many special properties and so it is natural to ask if this same magnetization behavior holds under more general conditions. In this part of the tutorial we investigate the effects of adding a $J_z S_i^z S_{i+1}^z$ term to the Hamiltonian, corresponding to the XXZ model. In the limit as this term dominates the spins become frozen in a parallel configuration, and so the initial state becomes an exact eigenstate of the Hamiltonian. The XX terms in the Hamiltonian try to flip the spins, and are responsible for the propagating magnetization wavefront we saw in the pure XX model. As a quantitative measure of the ability of the system to transport spin, we consider the integrated flow of magnetization through the center defined in Phys. Rev. E **71**, 036102 (2005) as

$\Delta M(t) = \sum_{n>L/2}^{L} (\langle S_n^z(t) \rangle + 1/2)$

C.3.5.1 Preparing and running the simulation using Python

To set up and run the simulation in Python we use the script tutorial2d.py. The first parts of this script imports the required modules and prepares the input files as a list of Python dictionaries:

```
import pyalps
import matplotlib.pyplot as plt
import pyalps.plot
import math
import scipy.special

#prepare the input parameters
parms=[]
count=0
for z in [0.0, 0.3, 0.9, 1.0, 1.1, 1.5]:
        count+=1
        parms.append({
                'L'                         : 50,
                'MODEL'                     : 'spin',
                'local_S'                   : 0.5,
                'CONSERVED_QUANTUMNUMBERS'  : 'Sz',
                'Jxy'                         : 1,
                'Jz'                         : z,
                'INITIAL_STATE' : 'kink',
                'CHI_LIMIT' : 40,
                'TRUNC_LIMIT' : 1E-12,
                'NUM_THREADS' : 1,
                'TAUS' : [20.0],
                'POWS' : [0.0],
                'GS' : ['H'],
                'GIS' : [0.0],
                'GFS' : [0.0],
                'NUMSTEPS' : [500],
                'STEPSFORSTORE' : [5],
                'SIMID': count
              })
```

Note that we are simulating a range of Jz-couplings. We then write the input files, run the simulation, and get the output as usual:

```
baseName='tutorial_2d'
nmlnameList=pyalps.writeTEBDfiles(parms, baseName)
res=pyalps.runTEBD(nmlnameList)
```

```
#Get magnetization data
Magdata=pyalps.load.loadTimeEvolution( pyalps.getResultFiles(
    prefix='tutorial_2d'), measurements=['Local␣Magnetization'])
```

From the computed magnetization data we calculate the integrated magnetization as defined above:

```
#Compute the integrated magnetization across the center
for q in Magdata:
        syssize=q[0].props['L']
        #Compute the integrated flow of magnetization through the
             center \Delta M=\sum_{n>L/2}^{L} (<S_n^z(t)>+1/2)
        #\Delta M= L/4
        loc=0.5*(syssize/2)
        #\Delta M+=<S_n^z(t)> from n=L/2 to L
        q[0].y=[0.5*(syssize/2)+sum(q[0].y[syssize/2:syssize])]
```

Finally, we plot the integrated magnetization for the range of Jz couplings simulated.

```
#Plot the integrated magnetization
Mag=pyalps.collectXY(Magdata, x='Time', y='Local␣Magnetization',
     foreach=['Jz'])

plt.figure()
pyalps.plot.plot(Mag)
plt.xlabel('Time␣$t$')
plt.ylabel('Integrated␣Magnetization␣$\Delta␣M(t)$')
plt.title('Integrated␣Magnetization␣vs.␣Time')
plt.legend(loc='upper␣left')
plt.show()
```

We see that for Jz<1 the integrated magnetization increases roughly linearly in time, and so the magnetization transport is ballistic as in the XX case. For Jz around 1, we see a change in the qualitative behavior to one in which the integrated magnetization eventually saturates.

C.3.5.2 Preparing and running the simulation using Vistrails

To run the simulation in Vistrails open the file tutorial2d.vt and look at the workflow labeled "tutorial2d". Click on "Execute" to prepare the input file, run the simulation and create the output figure

C.3.5.3 Questions

- The point Jz=1 where the behavior of the integrated magnetization undergoes a distinct qualitative change is the point at which the XXZ model transitions from a critical phase to the Antiferromagnetic phase. However, this phase transition is a priori a low-energy phenomenon, affecting the ground state. Can you deduce how this low energy change affects the dynamical properties of our high-energy initial state?

APPENDIX D - EDUCATIONAL MATERIALS: A GENTLE INTRODUCTION TO TIME EVOLVING BLOCK DECIMATION (TEBD)

D.1 Pre-test

Before we begin, I'm assuming that you have finished the "Introduction to the Bose-Hubbard Model and Fock State Basis" problem set. In particular, it is important that you have finished problem 6 and have a working code which will compute the ground state of an N particle, L site Bose-Hubbard system with local dimension d. The below exercises will test your familiarity with the Bose-Hubbard Hamiltonian and the Fock state basis. If you can't complete them with relative ease, you may want to review the material presented in the other problem set before proceeding.

D.1.1 Exercise 1

Consider a system with three sites occupied by bosons. The Fock space is spanned by the states $|n_1 n_2 n_3\rangle$. What is the overlap of the states $|110\rangle$ and $|010\rangle$ (That is, what is $\langle 110|010\rangle$)? What are $\langle 110|\hat{b}_1|010\rangle$ and $\langle 110|\hat{b}_1^\dagger|010\rangle$?

D.1.2 Exercise 2

Now consider the total number operator defined by $\hat{N} \equiv \sum_{i=1}^{3} \hat{n}_i$, where $\hat{n}_i = \hat{b}_i^\dagger \hat{b}_i$. What is the action of $\hat{N}$ on the states $|\psi_1\rangle = |231\rangle$, $|\psi_2\rangle = \frac{1}{\sqrt{5}}|022\rangle + \sqrt{\frac{4}{5}}|101\rangle$, and $|\psi_3\rangle = \sqrt{\frac{1}{3}}|112\rangle + \sqrt{\frac{2}{3}}|022\rangle$. Which of these states (if any) are total number eigenstates such that $\hat{N}|\psi\rangle = N|\psi\rangle$?

D.1.3 Exercise 3

In the Bose-Hubbard Hamiltonian, the tunneling term is

$$-t \sum_{\langle i,j\rangle} \left(\hat{b}_i^\dagger \hat{b}_j + \hat{b}_i \hat{b}_j^\dagger\right) . \tag{D.1}$$

What does the second term in parentheses represent physically? Why is it important mathematically?

D.2 The Limitations of Exact Diagonalization

In the previous problem set you wrote a code to explicitly construct the Bose-Hubbard Hamiltonian, diagonalize it, and thus find the ground state energy and eigenvector. The process of finding the ground state of a many-body hamiltonian in terms of a fixed Fock basis is referred to as *exact diagonalization* (ED). ED is very powerful in that it gives us maximal information about the energy spectrum and eigenstates of the Hamiltonian, and also has the benefit of being very simple to understand and to program. As we shall see, ED is also very limited in the sizes of the systems it can handle.

D.2.1 Exercise 4: The Limitations of Exact Diagonalization

Using your ED code, find the largest Bose-Hubbard system your computer can handle for non-number conserving code before running out of memory. Repeat for the number conserving case.

From this previous exercise, we gather two things. The first is that ED is not going to be of much use in studying systems of more than a few sites. The second is that judicious exploitation of symmetries and their associated conservation laws, such as the conservation of total number in the Bose Hubbard model, drastically improves the performance of many-body simulations.

Time-Evolving Block Decimation (TEBD) is a powerful method to simulate many-body systems such as the Bose-Hubbard model whose computation time and memory usage scale polynomially in the system size (as opposed to exponentially in the system size, as in ED). It is also, however, much more difficult to conceptually understand and to program. In the next few subsections I will provide a crash course in what TEBD is and how it works. Those interested in learning only how to *use* the open source TEBD package should skip to subsection D.6.

D.3 Preliminaries

D.3.1 Mathematical Preliminaries

As is often the case, before we get to the physics we have to learn some math. In this subsection I present two theorems, the singular value decomposition and the Schmidt decomposition.

Theorem: (Singular value decomposition) Let A be an $m \times n$ matrix. Then there exists an $m \times m$ unitary matrix Γ_A, an $n \times n$ unitary matrix Γ_B, and an $m \times n$ positive diagonal (as defined for rectangular matrices) matrix λ such that

$$A = \Gamma_A \lambda \Gamma_B^T \,, \tag{D.2}$$

or, stated element-wise,

$$A_{ij} = \sum_{k=1}^{\min(m,n)} \left[\Gamma_A\right]_{ik} \lambda_{kk} \left[\Gamma_B^T\right]_{kj} \,. \tag{D.3}$$

This is referred to as the *singular value decomposition* of A, and the diagonal elements of λ are referred to as the *singular values* of A. Note that the form of the decomposition implies that A has at least one and at most $\min(m,n)$ distinct singular values. The number of nonzero singular values of a matrix is its *rank*.

The most important property of the SVD for our purposes is that the matrix $A^{(l)}$ defined by the matrix elements

$$A_{ij}^{(l)} = \sum_{k=1}^{l} \left[\Gamma_A\right]_{ik} \lambda_{kk} \left[\Gamma_B^T\right]_{kj} \tag{D.4}$$

is the closest rank-l matrix to A, meaning that $A^{(l)}$ minimizes the *Frobenius norm*: the sum of the absolute squares of the element-wise difference between the rank-

l approximation and the full matrix, $\sum_{ij}\left|A_{ij}-A_{ij}^{(l)}\right|^2$. Thus, the SVD gives us a means to find the best lower-rank approximation to a matrix. The reduction in rank is important because matrices of lower rank require much less storage and allow for more efficient operations. Explicitly, a rank-l approximation to an $m \times n$ matrix is comprised of $ml + nl + l$ numbers whereas the full rank matrix is comprised of mn numbers. For matrices with ranks small compared to their size, $l \ll m, n$, the reduction in storage is enormous.

We saw in the previous paragraph that a reduced rank approximation to a matrix requires less storage than the full rank matrix. Here we also explore how reducing the rank of a matrix improves the *efficiency* of operations performed with it. Consider multiplying an $n \times n$ matrix A to a vector v. The resulting vector Av can be written in indicial notation as

$$[Av]_i = \sum_{k=1}^{n} A_{ik} v_k \,. \tag{D.5}$$

The formation of each element of Av requires n multiplications, and there are n elements of Av, so the formation of the full vector Av requires $\mathcal{O}\left(n^2\right)$ operations,[99] where $\mathcal{O}\left(a\right)$ is read “of order a.”[100] Now consider taking the singular value decomposition of A, and forming a rank-l approximation $A^{(l)}$:

$$A_{ij}^{(l)} = \sum_{k=1}^{l} \left[\Gamma_A\right]_{ik} \lambda_{kk} \left[\Gamma_B^T\right]_{kj} \,. \tag{D.6}$$

How many operations are required to form $A^{(l)}v$? Naively, we would write

$$\left[A^{(l)}v\right]_{ij} = \sum_{k=1}^{l} \sum_{j=1}^{n} \left[\Gamma_A\right]_{ik} \lambda_{kk} \left[\Gamma_B^T\right]_{kj} v_j \,, \tag{D.7}$$

---

[99] by operation we mean an elementary operation such as an addition, multiplication, subtraction, or division. In computer science language, where we deal with floating point representations of numbers, such an operation is called a floating-point operation, or FLOP.

[100] For more information on this notation, see the wikipedia page for Big-O notation.

which seems to involve $\mathcal{O}\left(n^2 l\right)$ operations, *increasing* the number of operations by a factor of l! We can be more clever, however, and first multiply v by Γ_B^T to get a vector q

$$q_i = \sum_{k=1}^{n} \left[\Gamma_B^T\right]_{ik} v_k \,. \tag{D.8}$$

Note that q has l elements and this multiplication involves $\mathcal{O}\left(nl\right)$ operations. We now multiply each element of q by an element of λ to get

$$q_i = \lambda_i q_i \tag{D.9}$$

which involves $\mathcal{O}\left(l\right)$ operations. Finally, we multiply q by Γ_A to get

$$\left[A^{(l)} v\right]_i = \sum_{j=1}^{l} \left[\Gamma_A\right]_{ij} q_j \,, \tag{D.10}$$

which again involves $\mathcal{O}\left(nl\right)$ operations. Thus, our total operation count is $nl + l + nl$, which is $\mathcal{O}\left(nl\right)$! For $l \ll n$, the reduction in the number of operations is again enormous.

D.3.1.1 Exercise 5: General matrix-vector multiply using the SVD

How many operations does it take to multiply an $m \times n$ matrix A to an n dimensional vector v both with and without the singular value decomposition?

D.3.1.2 Exercise 6: Simple examples of the SVD

Compute the singular value decompositions of the matrices

$$c_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \; c_2 = \begin{pmatrix} 0 & 1/\sqrt{2} \\ -1/\sqrt{2} & 0 \end{pmatrix}, \; c_3 = \begin{pmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{pmatrix} \tag{D.11}$$

numerically. The simplest way to do so is using *Matlab* or *Mathematica*, where the commands are `[U,S,V]=svd(A)` and

`{U,S,V}=SingularValueDecomposition[A]`,[101] respectively. Those who wish to use Fortran or C++ can use the lapack routine `dgesvd`. What are the ranks of these matrices? Could you have guessed their ranks simply by looking at them?

The singular value decomposition is a general mathematical statement about a way to rewrite any arbitrary matrix. The next theorem, the Schmidt decomposition, gives a more physical understanding of what the singular value decomposition *means* in the context of a tensor product space.

Theorem: (Schmidt Decomposition) Let $|\psi\rangle$ be a state vector in the $d_A d_B$ dimensional Hilbert space $\mathcal{H}_A \otimes \mathcal{H}_B$.[102] Then there exist vectors $\left\{|\phi_\alpha^A\rangle\right\}$ and $\left\{|\phi_\alpha^B\rangle\right\}$ and scalars χ_S and $\{\lambda_\alpha\}$ such that

$$|\psi\rangle = \sum_{\alpha=1}^{\chi_S} \lambda_\alpha |\phi_\alpha^A\rangle \otimes |\phi_\alpha^B\rangle \,, \tag{D.12}$$

$$1 \le \chi_S \le \min\left(d_A, d_B\right) \,, \tag{D.13}$$

$$\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_{\chi_S} > 0 \,, \tag{D.14}$$

$$\sum_\alpha \lambda_\alpha^2 = 1 \,. \tag{D.15}$$

This is referred to as the *Schmidt decomposition* of $|\psi\rangle$. χ_S is referred to as the *Schmidt rank*, and the $\{\lambda_\alpha\}$ are referred to as the *Schmidt coefficients*. We shall prove this theorem to elucidate the connection between the Schmidt rank and the singular value decomposition.

Proof: Let $\{|j_A\rangle\}$ and $\{|n_B\rangle\}$ be two orthonormal bases of $\mathcal{H}_A$ and $\mathcal{H}_B$ of dimension d_A and d_B, respectively. The most general decomposition of $|\psi\rangle$ in this basis is

$$|\psi\rangle = \sum_{j=1}^{d_A} \sum_{n=1}^{d_B} c_{jn} |j_A\rangle \otimes |n_B\rangle \,. \tag{D.16}$$

---

[101] Note in *Mathematica* that it returns U, S, and V such that $A = USV^T$

[102] To be concrete, you could consider $\mathcal{H}_A$ to be the Hilbert space of the first site and $\mathcal{H}_B$ to be the Hilbert space of the second site of a two site system. Or you could consider $\mathcal{H}_A$ to represent the internal states of some particle and $\mathcal{H}_B$ to represent the internal states of another particle.

Applying the singular value decomposition to the coefficient matrix C such that

$$c_{jn} = \sum_{i=1}^{\min(d_A,d_B)} \left[\Gamma_A\right]_{ji} \lambda_{ii} \left[\Gamma_B^T\right]_{in} , \tag{D.17}$$

we have

$$|\psi\rangle = \sum_{j=1}^{d_A} \sum_{i=1}^{\min(d_A,d_B)} \sum_{n=1}^{d_B} \left[\Gamma_A\right]_{ji} \lambda_{ii} \left[\Gamma_B^T\right]_{in} |j_A\rangle \otimes |n_B\rangle . \tag{D.18}$$

If we now define

$$|\phi_i^A\rangle \equiv \sum_{j=1}^{d_A} \left[\Gamma_A\right]_{ji} |j_A\rangle , \quad |\phi_i^B\rangle \equiv \sum_{n=1}^{d_B} \left[\Gamma_B^T\right]_{in} |n_B\rangle , \quad \lambda_i \equiv \lambda_{ii} , \tag{D.19}$$

we have

$$|\psi\rangle = \sum_{i=1}^{\chi_S} \lambda_i |\phi_i^A\rangle \otimes |\phi_i^B\rangle , \tag{D.20}$$

as was to be proven. We can identify χ_S generally as the number of nonzero singular values of the decomposition matrix, with $1 \leq \chi_S \leq \min(d_A, d_B)$. We also note that the Schmidt decomposition may be easily performed numerically using the singular value decomposition of the coefficient matrix C.

What has the Schmidt decomposition done for us? The Schmidt decomposition of $|\psi\rangle$ provides us with unique orthonormal bases for the two subsystems A and B such that $|\psi\rangle$ can be written as a superposition of tensor products with the least possible number of terms. In addition, the sizes of the Schmidt coefficients λ_i give the "importance" of the particular state $|\phi_i^A\rangle \otimes |\phi_i^B\rangle$ in the representation of $|\psi\rangle$, and the Schmidt rank tells us how difficult it is to write $|\psi\rangle$ as a tensor product, loosely speaking.

These statements can be made more precise and physically meaningful by introducing the quantum mechanical idea of *entanglement*, as is done in the next subsection.

D.3.2 Entanglement

We again consider a system with two parts, which we refer to as a *bipartite* system. A general ket $|\psi\rangle$ in this system can be written as

$$|\psi\rangle = \sum_{n,m} c_{nm} |n\rangle \otimes |m\rangle \tag{D.21}$$

where the $\{|n\rangle\}$ and $\{|m\rangle\}$ form complete orthonormal bases for the first and second parts, respectively. We see that this is not, in general, a tensor product $|\phi_1\rangle \otimes |\phi_2\rangle$ of kets in the subspaces. States in the composite space which are not tensor products of kets from the subspaces are said to be *entangled*. For example, if we examine a two-qubit[103] system, a general ket can be written

$$|\psi\rangle = c_{11}|+\rangle \otimes |+\rangle + c_{12}|+\rangle \otimes |-\rangle + c_{21}|-\rangle \otimes |+\rangle + c_{22}|-\rangle \otimes |-\rangle \tag{D.22}$$

whereas general normalized kets of the subsystems can be written

$$\begin{aligned} |\phi_1\rangle &= a_1|+\rangle + b_1|-\rangle && \text{(D.23)} \\ |\phi_2\rangle &= a_2|+\rangle + b_2|-\rangle && \text{(D.24)} \\ \Rightarrow |\phi_1\rangle \otimes |\phi_2\rangle &= a_1a_2|+\rangle \otimes |+\rangle + a_1b_2|+\rangle \otimes |-\rangle + b_1a_2|-\rangle \otimes |+\rangle + b_1b_2|-\rangle \otimes |-\rangle && \text{(D.25)} \end{aligned}$$

$|\psi\rangle = |\phi_1\rangle \otimes |\phi_2\rangle$ iff $c_{11}c_{22} = c_{12}c_{21}$, which is certainly not the general case.

We saw above that the Schmidt rank gave us a measure of how difficult it was to accurately represent our state as a tensor product. From the above considerations, we see that states with a higher Schmidt rank are *more entangled*. By truncating at a fixed Schmidt rank, we find the best approximation to our state in a Hilbert space that truncates the amount of entanglement allowed. This statement is very key, so I'm going to write it out in bold so that you remember it:

---

[103] a *qubit* is a general two-state system, for example the internal states of a spin-1/2 particle. Here $|+\rangle$ represents one of the two states and $|-\rangle$ the other.

> **By performing the Schmidt decomposition on the state of a bipartite system and truncating the Schmidt rank, we find the best approximation to the state in a Hilbert space with restricted entanglement.**

The efficient approximation of a state in a space with restricted entanglement is the key idea of TEBD. In the next subsection we will compare the above mathematical ideas with the more tangible idea of image compression.

D.3.2.1 Exercise 7: Studying entanglement via the Schmidt Decomposition

We can now give a physical interpretation to the purely mathematical Exercise 6. Write down the coefficient matrices of the states

$$|\psi_1\rangle = |+\rangle \otimes |+\rangle \tag{D.26}$$

$$|\psi_2\rangle = \frac{1}{\sqrt{2}}\left(|+\rangle \otimes |-\rangle - |-\rangle \otimes |+\rangle\right) \tag{D.27}$$

$$|\psi_3\rangle = \frac{1}{2}\left[|+\rangle \otimes |+\rangle + |+\rangle \otimes |-\rangle + |-\rangle \otimes |+\rangle + |-\rangle \otimes |-\rangle\right] \tag{D.28}$$

$$\tag{D.29}$$

in the form

$$C = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}, \tag{D.30}$$

where c_{ij} is given as in Eq. (D.22). You should recover the matrices of Exercise 6. Using their singular value decompositions, find the Schmidt ranks. Which of these states are entangled according to the Schmidt rank? For those states that are not entangled, can you find a way to write them as a tensor product? How might you use the singular value decomposition to accomplish this?

D.4 Conceptual Basis of TEBD

The essential fact that allows for the efficiency and success of TEBD is that the singular values of reduced density matrices,[104] when arranged in non-increasing fashion, have an approximately exponential decay. This means that a rank-χ approximation to the reduced density matrix, $\hat{\rho}^{(\chi)}$, formed from the singular value decomposition as in Eq. (D.4) with $\chi \ll \chi_S$ will provide an excellent approximation to the true reduced density matrix. We can understand this fact using an analogy to image compression.

Consider the JPEG image shown in Figure D.1(a). We can represent this picture as a 300×416 array of pixels, requiring 124800 words of storage. If we look at the singular values of this pixel array, we find the distribution shown in Figure D.1(b), namely that the singular values decay roughly exponentially. This implies that the best approximation to the image given by the SVD will be excellent even if we keep only a fraction of the total singular values. To see this in action, examine Figures Figure D.1(c)–Figure D.1(e), which show the best approximation to the image for various numbers of singular values. Keeping 100 singular values gives an excellent approximation to the original image with only a quarter of the data storage. If we consider that the original image was a JPEG which itself was compressed down from a ∼3000×3000 pixel (10 Megapixel) raw array, we have benefitted enormously by carefully sampling the parts of "image space" that are the most important via the SVD.

Why were we able to represent the image with so little data? The reason is that a physical images are a very special subset of all 2D pixel arrays; they have a great deal of structure and regularity. If we consider instead the most probable image of the same size as our original image–one consisting of random pixel values–we get the singular value scaling shown in Figure D.1(f). In this case the singular values are all roughly the same, and our SVD compression would yield a miserable approximation

---

[104] The *reduced density matrix* is a general way of dealing with subsystems of quantum mechanical systems. For our purposes, it takes the place of the coefficient tensor when the system has more than two parts (e.g. more than two lattice sites).

to the original image if we were to use only a quarter of the singular values. In the same manner typical physical states in Hilbert space, including the ground state, have limited entanglement as quantified by *some* entanglement measure. This limited entanglement means the singular value spectra of physical states decay exponentially more quickly than a general state, enabling them to be "entanglement compressed" by algorithms such as TEBD.

D.5 TEBD in Summary

Before going on to use TEBD, let's pause to reiterate what TEBD is and why it works where ED fails. TEBD is a method which gives the best approximation to the state of a many-body system by truncating the amount of entanglement allowed in the state. This is done by the singular value decomposition, which reduces the rank of the matrices carrying the state information. We have seen that reducing the rank of a matrix allows it to be stored more compactly, and operations to be done more efficiently. In the case of TEBD, the rank of the matrices we keep is *exponentially smaller* than the size of the matrices they approximate. The exponential decrease in storage and exponential increase in efficiency of operations are what allow us to use TEBD where ED fails. For all of the gory details on TEBD, you can consult the User's guide for the open source package, located on the downloads page.

D.6 Using OSTEBD

To begin, download the most recent version of the open source TEBD package from the downloads page (unless you have a more recent version from another place). After decompressing the `.tar.gz` file, go into the `Case_Studies` directory, and then into `Bose_Hubbard_Wrapper`. Open a terminal in this directory. Type `make ITP`. If you see `Execute_ITP` in this folder after a while, the case study has compiled. Otherwise, it didn't compile for some reason. You should be sure that you have gfortran as well as lapack installed and try again.

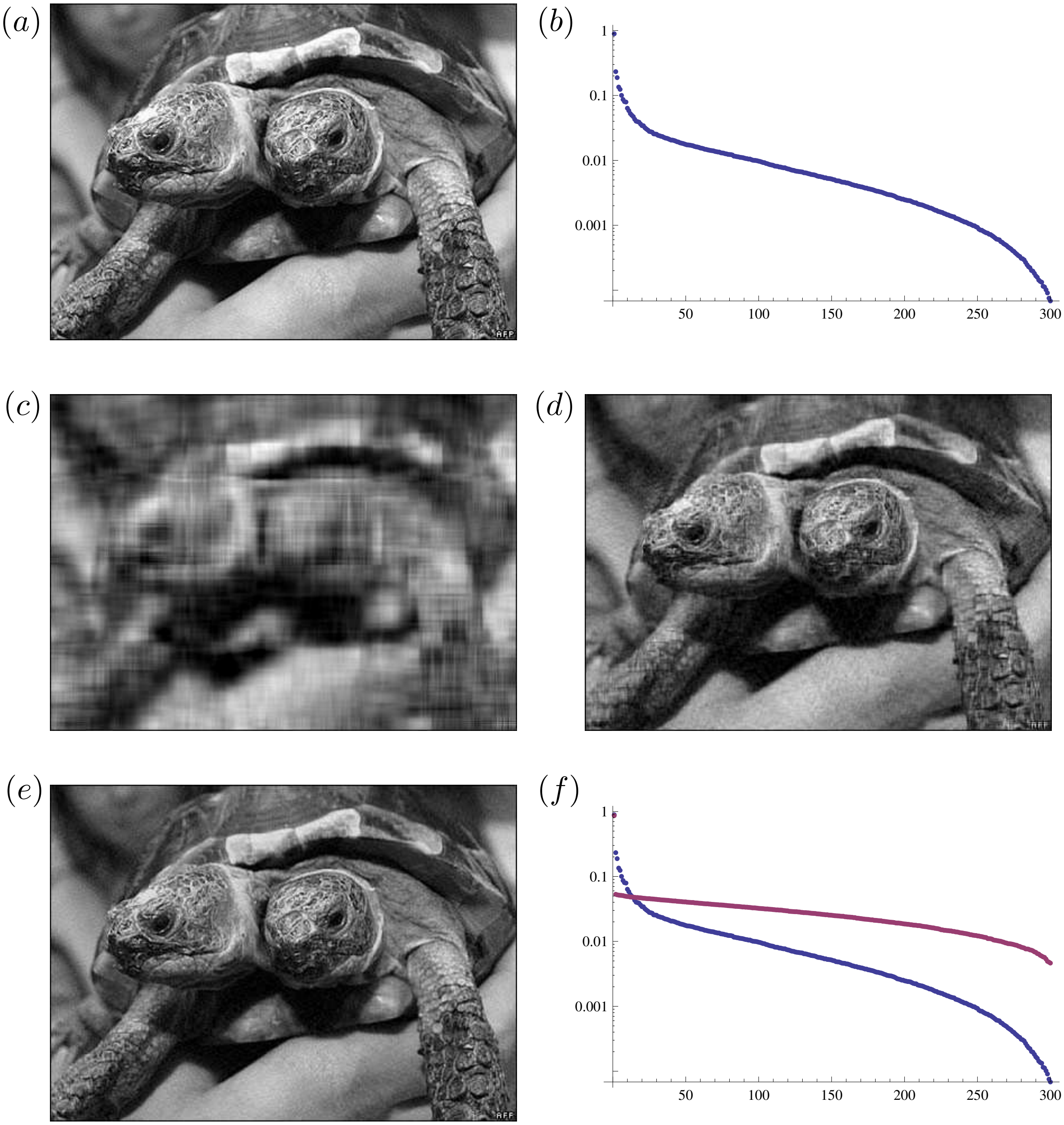


Figure D.1: SVD representation of an image. (a) Original Image. (b) Log plot of singular values. (c) 10 Singular values. (d) 50 Singular values. (e) 100 Singular values. (f) Log plot of singular values: General pixel array (pink), Physical pixel array (blue).

Once you have compiled successfully, open the file `BoseHubbard_ITP.nml` This is the input file, which sets the parameters for a specific TEBD job. You will see three lines, denoted by `SystemSettings`, `BHParams`, and `ITPParams`. The parameters in `SystemSettings` define the properties of the system as a whole. Specifically, they are

1. `systemSize`, the number of lattice sites.
2. `maxFilling`, the maximum number of bosons allowed per lattice site.
3. `totNum`, the total number of bosons for number conserving code. If `totNum`=0, the non-number conserving code is used.
4. `BoundaryCond`, the boundary conditions used. It can be either 'O' for open boundary conditions or 'P' for periodic boundary conditions.
5. `trotterOrder`, the order of the trotter expansion of the propagator. It can be either 2 or 5.

In very recent versions you may also have `numThr`, which specifies the number of threads used for OpenMP parallelization. Unless you know what you are doing, set this to be 1. The parameters in `BHParams` represent the parameters in the Bose-Hubbard model. Specifically they are

1. `jTunn`, the tunneling energy t.
2. `U0`, the on-site interaction energy U.
3. `V0`, the nearest-neighbor interaction energy V.
4. `mu0`, the chemical potential μ. Note that the chemical potential sets the total number when the non-number conserving code is used, but only gives an overall (unimportant) energy shift when the number conserving code is used.

The parameters contained in `ITPParams` determine the convergence and output properties of the TEBD run. Specifically, they are

1. `chiMin`, the minimum value of the entanglement cutoff parameter χ used to calculate the ground state.
2. `chiMax`, the maximum value of the entanglement cutoff parameter χ used to calculate the ground state.
3. `convCriterion1`, the convergence criterion for the first ITP.[105] attempt.
4. `convCriterion2`, the convergence criterion for the first ITP attempt.
5. `stepsForJudge` determines how many ITP steps occur before convergence is checked for.
6. `dtITP`, the "infinitesimal" timestep used for ITP.
7. `maxITPsteps`, the maximum number of ITP steps allowed.
8. `itpDir`, the directory where the output will be written. Note that this directory must exist when you run the program or it will exit!

After setting the parameters you want in the `BoseHubbard_ITP.nml` file, type `./Exectute_ITP` to run the TEBD program.

D.6.1 Exercise 8: Using TEBD:Non-Number Conserving Version

Set `SystemSize`=4, `totNum`=0, `BoundaryCond`='O', `trotterOrder`=2, `chiMin`=`chiMax`=10, and `maxFilling` to 3. Then set `jTunn`=1.12, `U0`=0.72, `V0`=0.0, and `mu0`=0.5. Run the code as described above, and note the total number and total energy. Using your ED code, calculate the ground state energy and total number of particles for the same parameters and without number conservation and compare them to the TEBD predictions.

---

[105] ITP is short for *imaginary time propagation* Imaginary time propagation replaces $t = -it$ in the Schrödinger equation, which turns it into a diffusion equation. The highest energy eigenmodes decay the most quickly during imaginary time evolution, and so in the limit of long times imaginary time propagation gives us the ground state.

D.6.2 Exercise 9: Using TEBD:Number Conserving Version

Set `SystemSize`=4, `totNum`=4, `BoundaryCond`='O', `trotterOrder`=2, `chiMin`=`chiMax`=10, and `maxFilling` to 3. Then set `jTunn`=1.12, `U0`=0.72, `V0`=0.0, and `mu0`=0.0. Run the code as described above, and note the total energy. Using your ED code, calculate the ground state energy with number conservation and compare it to the TEBD prediction. Also note the large increase in efficiency of the number conserving codes vs. non-number conserving codes.

I hope the last two exercises give you some confidence that the OSTEBD package works. However, TEBD is hardly useful for 4 sites where ED still works and is likely faster. In the next exercise we will study the analytically solvable *tight-binding* limit of the Bose-Hubbard model using TEBD for large numbers of sites where ED fails.

D.6.3 Exercise 10: The Tight-Binding Chain

Consider the Bose-Hubbard model in the limit $U \to 0$, $V \to 0$ and a fixed number of particles:

$$\hat{H} = -t \sum_i \left(\hat{b}_i^\dagger \hat{b}_{i+1} + \hat{b}_i \hat{b}_{i+1}^\dagger \right) . \tag{D.31}$$

This Hamiltonian, known as the *tight-binding* Hamiltonian, represents noninteracting bosons which are free to hop between sites on a lattice. It is the many-body generalization of the particle in an infinite well problem from single-particle quantum mechanics, and can be solved similarly. We can write the k^{th} eigenstate for a single particle as

$$|\psi_k\rangle = \sum_{i=1}^{L} \psi_{ik} \hat{b}_i^\dagger |0 \ldots 0\rangle \tag{D.32}$$

where the ψ_{ik} are complex scalars to be determined and $|0 \ldots 0\rangle$ is the Fock state with no particles in any site (the Fock vacuum). Inserting this solution into the

Schrödinger equation $\hat{H}|\psi\rangle = E|\psi\rangle$, we have the recursion relation

$$-t\left(\psi_{i-1,k} + \psi_{i+1,k}\right) = E_k \psi_{ik} \tag{D.33}$$

together with appropriate boundary conditions. For open boundary conditions, we have

$$\psi_{0k} = 0\,,\ \psi_{L+1,k} = 0 \tag{D.34}$$

and for periodic boundary conditions we have

$$\psi_{1k} = \psi_{L+1,k}\,. \tag{D.35}$$

Let us now define the ansatz

$$\psi_{jk} = \begin{cases} Ae^{ikj} + Be^{-ikj} & \text{OBC} \\ Ae^{ikj} & \text{PBC} \end{cases}\,. \tag{D.36}$$

Inserting this ansatz into the above recursion relation gives a set of algebraic equations for the ψ_{ik} and the energy E_k. Solve these equations and determine the single-particle eigenfunctions and energies for both open and periodic boundary conditions. Then, normalize these solutions such that $\langle\psi_k|\psi_k\rangle = 1$.

We note that the operator

$$\hat{s}_k^\dagger = \sum_{i=1}^{L} \psi_{ik} \hat{b}_i^\dagger \tag{D.37}$$

creates an eigenstate of the tight binding Hamiltonian with quantum number k. Since the particles do not interact, the solution for N particles with quantum numbers $k_1, k_2 \ldots k_N$ is simply

$$|\psi\rangle = \mathcal{N}\hat{s}_{k_N}^\dagger \ldots \hat{s}_{k_2}^\dagger \hat{s}_{k_1}^\dagger |0\ldots 0\rangle\,, \tag{D.38}$$

where $|0\dots0\rangle$ is again the Fock vacuum and

$$\mathcal{N} = \frac{1}{\sqrt{\prod_k n_k!}} \tag{D.39}$$

is a normalization factor accounting for the indistinguishability of bosons. The energy of this state is $E = E_{k_1} + E_{k_2} + \cdots + E_{k_N}$. This completes the solution of the N-particle tight-binding Hamiltonian. For bosons, the lowest energy state is the one where all particles are in the lowest energy single-particle eigenstate. Using the above results, find what the expectation value of the number operator in the ground state is at each site for both open and periodic boundary conditions.[106] As a slight generalization, also compute the *single-particle density matrix* defined by

$$\rho_{ij} = \langle \text{g.s.}|\hat{b}_i^{\dagger}\hat{b}_j|\text{g.s.}\rangle \,. \tag{D.40}$$

Using some system size which is much too large for ED (say, 25), study the tight-binding Hamiltonian using TEBD for both open and periodic boundary conditions. In particular, compare the results for the energy and the on-site number expectation values. To compare with your predictions for the on-site number expectation values, look in the directory `ITPDATA` for the file ending in `localmeasures.dat`. The full filename will be in the form `BH_L$N@ChiXjTunnP.PPUQQ.QQ*BClocalmeasures`, where `$` is the number of lattice sites, `@` the number of particles, `X` the chi value and so on. On the first line of this file are the on-site number expectation values, listed in order of increasing site index.

---

[106] There's an easy way and a hard way to do this. The easy way involves representing the number operator $\hat{n}_i = \hat{b}_i^{\dagger}\hat{b}_i$ in terms of the $\hat{s}_k$ using Eq. (D.37) and then taking the expectation value using Eq. (D.38). The hard (but fun) way is to expand the solution Eq. (D.38) in terms of $\hat{b}_i$ using Eq. (D.37) and then take the expectation value of $\hat{n}_i = \hat{b}_i^{\dagger}\hat{b}_i$.

D.6.4 Exercise 11: Fidelity Susceptibility and the Superfluid-Mott Insulator Quantum Phase Transition

The previous exercise allowed us to go beyond where ED could, but the results were still amenable to analytic computation. Here we go beyond the reach of analytic techniques to study a strongly interacting Bose-Hubbard system, and investigate its *quantum phase transition*. By a quantum phase transition we mean a transition between two different states of matter (characterized by different symmetries) which is driven by quantum fluctuations at zero temperature.[107] Specifically, in the Bose-Hubbard model the quantum phase transition occurs because of fluctuations in the on-site number as we change the ratio t/U of the hopping strength to the interaction.

To discover where the quantum phase transition is in the Bose-Hubbard model, let us return to the single particle density matrix (SPDM) from the last exercise, and focus on the case of OBC and $N = L$ (as many particles as lattice sites).[108] You should have found that the SPDM in the non-interacting OBC case had the form

$$\rho_{ij} = \frac{2N}{L+1} \sin\left(\frac{\pi i}{L+1}\right) \sin\left(\frac{\pi j}{L+1}\right) \tag{D.41}$$

which is plotted in Figure D.2 . What does this quantity tell us? Notice that the elements of the SPDM can be interpreted as the overlap of the two wavefunctions $\hat{b}_j|\text{g.s.}\rangle$ and $\hat{b}_i|\text{g.s.}\rangle$ Thus, we can interpret the elements of the single particle density matrix as the probability amplitude that if we remove a particle from the ground state at site i we will find it missing at site j, averaged over all the other $N-1$ particles (with the "diagonal" elements $i = j$ just being the number on site i, of course). For this to be nonzero with $i \neq j$ implies that the particles must be delocalized. In the present case the particles are highly delocalized, and this gives rise to mostly

---

[107] This is in contrast to classical phase transitions, which are driven by thermal fluctuations, and thus necessarily occur at nonzero temperatures.

[108] For good reason, as this quantum phase transition happens to occur only for integer filling $N/L = 1, 2, \dots$.

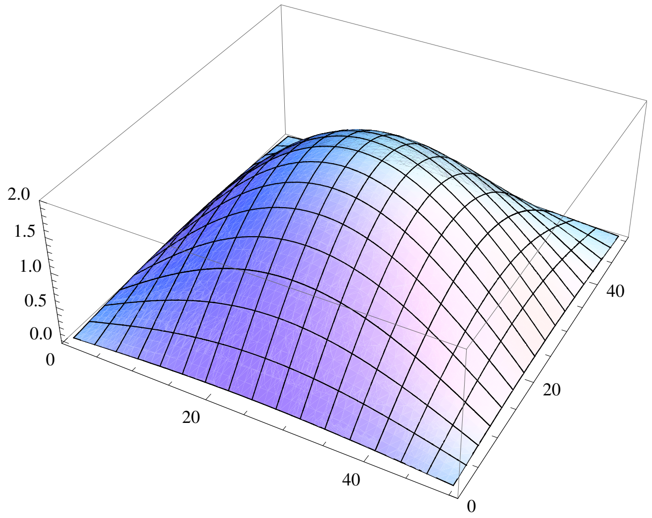


Figure D.2: Single particle density matrix, noninteracting system of $L = 50$ lattice sites and $N = 50$ particles.

nonzero SPDM matrix elements. Secondly, because this matrix is Hermitian it may be diagonalized and written in terms of its eigenvalues $\mathcal{N}_\nu$ and eigenvectors $\phi_\mu(i)$ as

$$\rho_{ij} = \sum_\mu \mathcal{N}_\mu \phi_\mu(i)^\star \phi_\mu(j) \ . \tag{D.42}$$

The eigenfunctions $\phi_\mu(i)$ behave in many respects like single-particle wavefunctions (although they do not, in general, diagonalize the single-particle part of the full many-body Hamiltonian) and we can interpret the eigenvalues $\mathcal{N}_\mu$ as the number of particles in the "single-particle state" ϕ_μ.[109] For our noninteracting case there is one eigenvalue $\mathcal{N}_1 = N$, with all others being zero. The corresponding eigenfunction ϕ_1 is an actual single-particle wavefunction in this case, and corresponds to the $k = 1$ mode (or the $k = 0$ mode for PBC). Thus we have that all of the particles occupy one particular single-particle mode, which is a hallmark of the phase of the Bose-Hubbard model known as the *superfluid* phase. We can express this result in terms of another quantity, called the *depletion*, as

---

[109] For more information on the physical content of the eigenvalues/vectors of the SPDM and their connection with Bose-Einstein condensation, I recommend A. J. Leggett's highly readable book "Quantum Liquids," which is available in the library.

$$D = 1 - \mathcal{N}_1/N \, . \tag{D.43}$$

The depletion in this case is 0, meaning that the all particles are condensed into one single-particle mode (the condensate is not at all depleted). When we add small interactions $\mathcal{N}_1$ will shrink somewhat, giving rise to nonzero depletion, and the eigenfunction ϕ_1 will no longer be a true single-particle wavefunction, but the overall behavior will be qualitatively the same (most of the elements of ρ_{ij} will be nonzero, there will be one SPDM eigenvalue that is $\mathcal{O}(N)$, etc.).

Let us now focus on the opposite limit of the Bose-Hubbard model, where we let the hopping $t \to 0$ and have U nonzero. The only surviving term for a fixed number of particles is

$$\hat{H} = \frac{U}{2} \sum_i \hat{n}_i \left(\hat{n}_i - 1\right) \, . \tag{D.44}$$

This Hamiltonian does not couple the different sites together, and so its ground state is a Fock state. In particular, for $N = L$, the ground state is the Fock state with one particle at each lattice site $|11\dots1\rangle$.[110] What does the SPDM look like in this case? A trivial calculation gives

$$\rho_{ij} = \delta_{ij} \, , \tag{D.45}$$

where δ_{ij} is the Kronecker delta. Loosely speaking, there is *no* probability of finding a removed particle missing from anywhere besides the exact spot it was removed from, and so we see that the particles are now highly localized! Also, we see that all of the eigenvalues of the single particle density matrix are 1, giving the depletion

$$D = 1 - 1/N \, . \tag{D.46}$$

---

[110] We arrive at this conclusion by noting that occupation of any site by two particles costs an energy U more than if those two particles occupied separate sites.

In the limit as $L \to \infty$ and $N \to \infty$ with N/L fixed (the thermodynamic limit) where a true quantum phase transition occurs, the depletion becomes 1, meaning that there are no particles in a condensed state (the condensate is fully depleted). The resulting phase is known as the *Mott Insulator*. If the transition from the superfluid to Mott insulator occurs at a finite value of t/U, then we expect the depletion to be zero below this value and nonzero above it. It is clear that the depletion is thus not a smooth (meaning infinitely continuously differentiable) function of t/U, and it is this singularity in the depletion (or one of its derivatives) that signals the quantum phase transition.

To "see" the quantum phase transition using TEBD, we could calculate the ground state for a series of lattices with increasing $L = N$ and a range of t/U, compute the depletion for each case, and try to find the point where it goes to zero as $L \to \infty$. However, this would require us to know ahead of time that the depletion is the quantity signaling the quantum phase transition. In TEBD we have the option of taking a more general approach, which I will outline here.

The ground states on either side of a quantum phase transition are very different from one another, and the non-analyticity at the transition point disallows us from smoothly "connecting" one type of ground state to another. Thus, we expect that the overlap of the ground state on one side of the critical point with the ground state on the other side should go to zero in the thermodynamic limit. This motivates the definition of the *fidelity*

$$f\left(\frac{t}{U}, \delta\frac{t}{U}\right) \equiv \left|\langle \text{g.s.}\left(\frac{t}{U}\right)|\text{g.s.}\left(\frac{t}{U}+\delta\frac{t}{U}\right)\rangle\right| \tag{D.47}$$

which is the overlap of the ground state at t/U with the ground state at $t/U + \delta t/U$. For small $\delta t/U$ this quantity should drop sharply only right near a quantum critical point, and does not require us to know anything about the system at hand. To remove the dependence on the step size $\delta t/U$ it is actually better to work with the *fidelity* *susceptibility*

$$\chi_f\left(\frac{t}{U}\right)=\frac{2}{L\left(\delta\frac{t}{U}\right)^2}\left(1-f\left(\frac{t}{U},\delta\frac{t}{U}\right)\right) \tag{D.48}$$

which can be shown to be independent of $\delta\frac{t}{U}$. This quantity diverges at the quantum critical point, and will be the basis of our TEBD study.

In order to be able to properly extrapolate finite size results to the infinite size limit, we will need to run TEBD for several numbers of sites L. In addition, because the prediction of where the quantum critical point is will change as L changes, we will need to simulate several t/U for each L. This amounts to running many simulations, and you will run out of time and/or patience if you try and do it sequentially on a desktop computer! Thus, in this exercise we will use the high performance resources available here at Mines to run TEBD simulations in parallel. The TEBD code we will use distributes a group of tasks (here computation of the ground state for a range of t/U) among a group of processors, and then compiles the results into a single file.

Log in either to Ra or Mio (Mio is preferred) and go to the carr group directory.[111] Go to the Case studies folder and then enter the `MPI_BH_Wrapper` directory. Type `make FS`, which compiles the main program `BoseHubbard_FS_MPI.f90`. This program calculates the ground state of the Bose-Hubbard model for $N = L$ and a user-specified range of t/U in parallel, and saves the states to disk as they are generated. Once all states have been generated, the routine reads them in and computes the fidelity susceptibility. We can use this routine to study the behavior of the fidelity susceptibility as L gets progressively larger. Also, because we save the states we generate, we can progressively refine our result for the fidelity susceptibility by gradually increasing the amount of entanglement allowed (controlled by χ) or by making the imaginary time step smaller.

---

[111] On Ra, this is /lustre/scratch/projects/carrgroup/. We will get such a directory started on Mio once it is fully operational.

The NAMELIST parameter file is `BH_FS.nml`. The parameters are:

1. `systemSize`, same meaning as before.
2. `maxFilling`, same meaning as before.
3. `BoundaryCond`, same meaning as before.
4. `Jmin`, the minimum value of t/U considered.
5. `jMax`, the maximum value of t/U considered.
6. `jres`, the number of t/U points from `Jmin` to `jMax`, inclusive.
7. `chiOld`, the old value of χ, used when reading in previously generated states.
8. `chiIn`, the value of χ for which you want to generate the fidelity susceptibility.
9. `convCriterion`, the convergence criterion for ITP.
10. `stepsForJudge`, same meaning as before.
11. `dtOld`, the old value of δt, used when reading in previously generated states.
12. `dtIn`, the value of δt for which you want to generate the fidelity susceptibility.
13. `maxITPsteps`, same meaning as before.
14. `FSDir`, the directory where the fidelity susceptibility data is stored.
15. `statesDir`, the directory where the state information is stored.

For the first run at a given `systemSize`, `chiOld` and `chiIn` should be the same. If more accuracy is desired after this first run, `chiOld` should be set to the value of χ you previously generated and `chiIn` should be set to the value you want to output at the end of the calculation. Identical reasoning applies to `dt*`. The output file in the directory `FSDir` ending in `FS.dat` has as its columns the values of t/U

and χ_f. These can be plotted using your favorite program (e.g. gnuplot, *Matlab*, or *Mathematica*). See Figure D.3 for example fidelity susceptibility curves to get you started on appropriate parameters.

To run a job in a high performance computing environment we must submit our job with a request for resources to the queue. This is done using the attached PBS file `tebdpbs`. The contents of the file are

```
#!/bin/csh
#PBS -l nodes=1:ppn=8
#PBS -l walltime=00:59:00
#PBS -N FS_MPI
#PBS -o FSstdout.$PBS_JOBID
#PBS -e FSstderr.$PBS_JOBID
#PBS -V
#PBS -m abe
#PBS -M YOURNAME@mines.edu
#------------------------
cd $PBS_O_WORKDIR
sort -u $PBS_NODEFILE  > mynodes.$PBS_JOBID

mpiexec Execute_FS > FSstdout.$PBS_JOBID
```

You should change `YOURNAME` to your mines user name. This has the supercomputer email you when a job begins, finishes, or is aborted. Out of the remainder, the lines you need to focus on are

```
#PBS -l nodes=1:ppn=8
#PBS -l walltime=00:59:00
```

The first specifies the number of nodes you are asking for, where each node is a computer with 8 processors. Choosing n nodes will speed up the calculation by a factor of $8n-1$ in the ideal case. The second line specifies how long the computer gives your job to finish before killing it. In the above it is set to 59 minutes. The longer you ask for, the longer you may have to wait before your job starts running (see the GECO webpage for more information about the different queues). To submit your job to the queue, type `msub tebdpbs`.

After you have generated χ_f for a series of L values, plot the locations of the maxima versus the lattice size and fit to a function of the form

$$L = f\left(\zeta_{\max L}\right) = C\left|\zeta_{\max L} - \zeta_{\max\infty}\right|^{-\eta} \tag{D.49}$$

with $C, \eta > 0$ as fit parameters, $\zeta = t/U$, $\zeta_{\max L}$ the value of ζ where χ_f is a maximum for a particular L, and $\zeta_{\max\infty}$ a fit parameter estimating the location of the quantum phase transition in the thermodynamic limit. Note that this calculation was only first done in 2007![112] Quantum phase transitions are an exciting and very current area of research, and powerful new methods such as TEBD are key to advancing understanding in the field.

---

[112] The relevant paper is *Phys. Rev. Lett.* **98** 110601, but don't look at it until you have a prediction for $\zeta_{\max\infty}$!

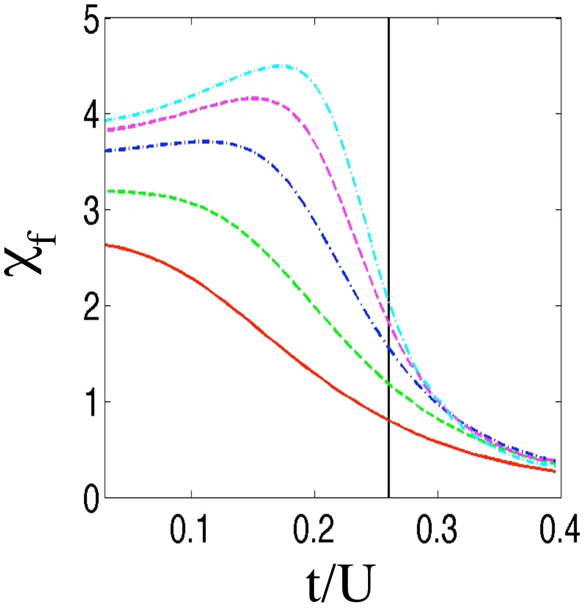


Figure D.3: Fidelity susceptibility for (bottom to top) $L = 3, 5, 10, 20, 40$.

APPENDIX E - EDUCATIONAL MATERIALS: INTRODUCTION TO MPS ALGORITHMS

The goal of this document is to give a series of successive steps leading to a working variational matrix product state (MPS) code for the Ising model. New in v3: sections marked with an asterisk ($*$) represent optimizations which should be skipped on first reading, but are important for efficient MPS code.

E.1 Matrix Product States and their Canonical Forms

The definition of a matrix product state (MPS) is

$$|\mathrm{MPS}\rangle=\sum_{i_1,\dots,j_L=1}^{d}\mathrm{Tr}\left(A^{i_1}\dots A^{i_L}\right)|i_1\dots i_L\rangle\,, \tag{E.1}$$

where L is the number of lattice sites on a 1D chain, i_j is a *physical index* denoting the state of the j^{th} lattice site, and d is the on-site dimension (which I will also call the local dimension). Because each A^{i_j} with i_j fixed is a matrix we use the terminology MPS. However, each A in fact has three elements: $A^{i}_{\alpha\beta}$. We will refer to an object with greater than two indices generically as a tensor, and the number of indices will be its rank. The dimensions of the spaces indexed by α and β are referred to as *bond dimensions*, with the bond dimension of an MPS being the maximum bond dimension of the matrices A. We will generically give a bond dimension the symbol χ. Simple arguments using the Schmidt decomposition show that the left and right bond dimensions in the case of open boundary conditions satisfy

$$\begin{aligned}\chi_\alpha&\le\min\left(d^{j-1},d^{L-j+1}\right)\,,\\ \chi_\beta&\le\min\left(d^{j},d^{L-j}\right)\end{aligned} \tag{E.2}$$

on the j^{th} site, as can be easily proved by Schmidt decomposition. Note in particular that this implies that the furthest left and furthest tensors $A^{[1]}$ and $A^{[L]}$[113] are $\max(\chi_\alpha, d, \chi_\beta) = (1, d, d)$ and $(d, d, 1)$, respectively. We will focus on the case of open boundaries for the remainder of this document. The case of periodic boundary conditions is considerably more complicated and should really be considered a separate algorithm altogether.

We plan to use MPS as a variational ansatz for finding the ground state of the Ising model. As such, we will need to define an MPS structure and initialize it with random numbers.

E.1.1 Step 1: MPS Structure

Define a `tensor` [114] structure which is an allocatable array with three indices. This structure will hold the tensors $A^i_{\alpha\beta}$. Next, define an `MPS` structure which is an array of `tensors`. For the first part of this problem set, real `tensors` and `MPSs` will suffice.

E.1.2 Step 2: MPS Initialization

Write a subroutine `AllocateMPS(psi,L,bondD,d)` which allocates an `MPS` `psi` on `L` sites with maximum bond dimension `bondD` and local dimension `d`. The left and right bond dimensions of each tensor A should obey the conditions Eq. (E.2). Also write a routine `DeallocateMPS` which deallocates the `MPS` structure. Finally, write a routine `CreateRandomMPS(psi,L,bondD,d)` which allocates an `MPS` as in `AllocateMPS`, but also assigns each element a random number$\in [-1, 1]$.

As discussed in Schollwöck,[115] the imposition and maintenance of canonical forms is absolutely essential to the efficiency of a variational MPS program. The canonical

---

[113] The notation $A^{[j]}$ means the tensor $A^{i_j}_{\alpha\beta}$ on site j. When all four indices are required, we will use $A^{[j]i_j}_{\alpha\beta}$

[114] I will use `typewriter face` whenever I refer to something which is to be coded.

[115] By which I mean Prof. Ulrich Schollwöck's major review *The density-matrix renormalization group in the age of matrix product states*, Annals of Physics **326**, 96 (2011).

form we will use most often is *Mixed canonical form* at site k, in which all tensors to the left of k satisfy the left orthogonality conditions

$$\sum_{\alpha,i} {A^{i}_{\alpha\beta}}^{\star} A^{i}_{\alpha\beta'} = \delta_{\beta,\beta'} \tag{E.3}$$

and all tensors to the right of k satisfy the right orthogonality conditions

$$\sum_{i,\beta} A^{i}_{\alpha\beta} {A^{i}_{\alpha'\beta}}^{\star} = \delta_{\alpha,\alpha} \,. \tag{E.4}$$

The site k itself satisfies no such orthogonality conditions, but we note that $\mathrm{Tr}_{i_1,\dots,i_L}\left(|\psi\rangle\langle\psi|\right) = \langle\psi|\psi\rangle = \sum_{\alpha,\beta,i_k} A^{[k]i_k}_{\alpha\beta} {A^{[k]i_k}_{\alpha\beta}}^{\star}$, and so this site carries all information about the norm of the state. We will refer to the site k in mixed canonical form as the *orthogonality center* of the MPS.

We can always choose the orthogonality conditions Eqs. (E.3) and (E.4) to hold because of the so-called *gauge freedom* inherent in the MPS representation. This freedom refers to the fact that if we consider any two MPS tensors $A^{[i]}$ and $A^{[i+1]}$, the same MPS results if we replace these by $\tilde{A}^{[j]i_j}$ and $\tilde{A}^{[j+1]i_{j+1}}$, where

$$\tilde{A}^{[j]i_j} \equiv A^{[j]i_j} X \,, \tag{E.5}$$

$$\tilde{A}^{[j+1]i_{j+1}} \equiv X^{-1} A^{[j+1]i_{j+1}} \,, \tag{E.6}$$

and X is any invertible matrix. In practice we implement the orthogonality conditions Eqs. (E.3) and (E.4) as in Figure E.1 and Figure E.2, respectively.[116]

You should take a minute to convince yourself of why this works, recalling that the matrices U and V of the singular value decomposition are unitary. Furthermore, you should pay close attention to groupings of indices and note that *two* tensors are changed at a time.

---

[116] The notation $[\alpha i]$ refers to the *Kronecker product* of the indices α and i. The product index $\xi = [\alpha i]$ runs from 1 to $d_\alpha d_i$, and an explicit representation is $\xi = (\alpha - 1)\, d_i + i$.

1. Reshape $A^{[j]i}_{\alpha\beta} \to A_{[\alpha i],\beta}$.
2. Perform the singular value decomposition (SVD) on this matrix to obtain $A_{[\alpha i]\beta} \to U_{[\alpha i]\gamma} S_{\gamma} V_{\gamma\beta}$.
3. Replace $A^{[j]i}_{\alpha\beta} = U_{[\alpha i]\beta}$.
4. Combine $(SV)_{\gamma\beta} = S_{\gamma} V_{\gamma\beta}$.
5. Contract SV into the A tensor to the right of the one just replaced $A^{[j+1]i}_{\alpha\beta} = (SV)_{\alpha\gamma} A^{i[j+1]}_{\gamma\beta}$.

Figure E.1: Algorithm to put site j into left canonical form.

1. Reshape $A^{[j]i}_{\alpha\beta} \to A_{\alpha[i\beta]}$.
2. Perform an SVD on this matrix to obtain $A_{\alpha[i\beta]} \to U_{\alpha\gamma} S_{\gamma} V_{\gamma[i\beta]}$.
3. Replace $A^{[j]i}_{\alpha\beta} = V_{\alpha[i\beta]}$.
4. Combine $(US)_{\alpha\gamma} = U_{\alpha\gamma} S_{\gamma}$.
5. Contract US into the A tensor to the left of the one just replaced $A^{[j-1]i}_{\alpha\beta} = A^{[j-1]i}_{\alpha\gamma} (US)_{\gamma\beta}$.

Figure E.2: Algorithm to put site j into right canonical form.

E.1.3 Step 3: Canonical Form

Write a routine `OrthogonalizeMPS(psi,k,kl,kr)` which puts the `MPS` `psi` into mixed canonical form with the orthogonality center at site `k`. The optional arguments `kl` and `kr` denote the furthest left and furthest right sites which are altered. The default values are 1 and L, respectively. Test this routine by explicitly checking the relations Eqs. (E.3) and (E.4) at each site. Do tensors in left canonical form obey any sort of right canonical form or vice versa? What about the orthogonality center-does it have any structure that you can discern?[117] Then, write a routine

---

[117] These are not trick questions...

`OrthonormalizeMPS(psi,k)` which shifts the orthogonality center of `psi` to site `k` and then normalizes the state by replacing $A^{[k]i}_{\alpha\beta} \to A^{[k]i}_{\alpha\beta}/\sqrt{\langle\psi|\psi\rangle}$, where $\langle\psi|\psi\rangle = \sum_{\alpha,\beta,i_k} A^{[k]i_k}_{\alpha\beta} A^{[k]i_k\star}_{\alpha\beta}$ as shown above. The svd may be performed using the LAPACK routine `dgesvd`.

OK, you now have an MPS in a form amenable to the variational procedure. Let's now move on to constructing the Hamiltonian in an appropriate way.

E.2 Matrix Product Operators

Just as an MPS can be viewed as a sum over tensor products of state-valued matrices:

$$|\mathrm{MPS}\rangle = \sum_{i_1\ldots i_L} \mathrm{Tr}\left(\mathcal{A}^{i_1} \otimes \cdots \otimes \mathcal{A}^{i_L}\right) , \tag{E.7}$$

$$\mathcal{A}^{i_j}_{\alpha\beta} \equiv A^{i_j}_{\alpha\beta}|i_j\rangle \tag{E.8}$$

we can define an object which is like an MPS but is operator-valued instead of state-valued

$$\hat{H}_{\mathrm{MPO}} = \sum_{i_1\ldots i_L} \mathrm{Tr}\left(\mathcal{W}^{i_1 i_1'} \otimes \cdots \otimes \mathcal{W}^{i_L i_L'}\right) , \tag{E.9}$$

$$\mathcal{W}^{i_j i_j'}_{\alpha\beta} \equiv W^{i_j i_j'}_{\alpha\beta}|i_j\rangle\langle i_j'| . \tag{E.10}$$

Such an object is called a *Matrix Product Operator* (MPO). As with an MPS, the i_j indices are physical, referring to actual sites in the lattice, and the indices implicit in the matrix product are called bond indices. The reason that MPOs are useful is that they map MPSs to MPSs. To see this, note that an MPO-MPS product is

$$\hat{H}|\psi\rangle = \mathrm{Tr}\left(\tilde{A}^{i_1}\ldots\tilde{A}^{i_L}\right)|i_1\ldots i_L\rangle , \tag{E.11}$$

$$\tilde{A}^{i_j}_{[\alpha k][\beta k']} \equiv \sum_{i_j'} A^{i_j'}_{\alpha\beta} W^{i_j i_j'}_{kk'} . \tag{E.12}$$

We see that, generically, the action of an MPO on an MPS increases the bond dimension of that MPS by the multiplicative factor χ_H, the bond dimension of the MPO. Because MPOs are operators, the relevant norm for them is the Frobenius norm $\langle \hat{H}|\hat{H}\rangle = \sqrt{\mathrm{Tr}\left(\hat{H}^\dagger \hat{H}\right)}$. However, this norm scales as d^L, and so we cannot hope to orthogonalize the matrices of an MPO in the same way that we did MPSs without running into serious numerical issues concerning precision. Luckily, because of the structure of physical operators such as many-body Hamiltonians it is possible to explicitly construct the MPOs in a canonical form in which all of the (matrix-valued!) matrices $W_{kk'}$ are lower triangular. For the case of a generic nearest-neighbor Hamiltonian

$$\hat{H} = \sum_{\langle i,j\rangle} \sum_{\alpha=1}^{p} J_\alpha \hat{O}^{(1)}_{i\alpha} \hat{O}^{(2)}_{j\alpha} + \sum_i \sum_{\beta=1}^{q} h_\beta \hat{O}^{(0)}_{i\beta}\,, \tag{E.13}$$

where α denotes the different nearest neighbor terms and β denotes the different on-site terms (each characterized by different operators $\hat{O}$), the MPO representation is

$$W^{[1]} = \left(\begin{array}{ccccc} \sum_{\beta=1}^{q} h_\beta \hat{O}^{(0)}_{1\beta} & J_1 \hat{O}^{(1)}_{11} & \dots & J_p \hat{O}^{(1)}_{1p} & I \end{array} \right), \tag{E.14}$$

$$W^{[j]} = \left(\begin{array}{ccccc} I & 0 & \dots & 0 & 0 \\ \hat{O}^{(2)}_{j1} & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \hat{O}^{(2)}_{jp} & 0 & \dots & 0 & 0 \\ \sum_{\beta=1}^{q} h_\beta \hat{O}^{(0)}_{j\beta} & J_1 \hat{O}^{(1)}_{j1} & \dots & J_p \hat{O}^{(1)}_{jp} & I \end{array} \right), j \neq 1\,,\ L\,, \tag{E.15}$$

$$W^{[L]} = \left(\begin{array}{c} I \\ \hat{O}^{(2)}_{L1} \\ \vdots \\ \hat{O}^{(2)}_{Lp} \\ \sum_{\beta=1}^{q} h_\beta \hat{O}^{(0)}_{L\beta} \end{array} \right). \tag{E.16}$$

To be very concrete, in the case of the Ising model

$$\hat{H} = -J \sum_{\langle i,j \rangle} \hat{\sigma}_i^z \hat{\sigma}_j^z - h \sum_i \hat{\sigma}_i^x \,, \tag{E.17}$$

and so $p = 1$, $q = 1$, $J_\alpha = -J$, $h_\beta = -h$, $\hat{O}_{i\alpha}^{(1)} = \hat{O}_{i\alpha}^{(2)} = \hat{\sigma}_i^z$, $\hat{O}_{i\beta}^{(0)} = \hat{\sigma}_i^x$. The MPO in this case is

$$\begin{aligned}
W^{[1]} &= \begin{pmatrix} -h\sigma^x & -J\hat{\sigma}^z & I \end{pmatrix} \,, \\
W^{[j]} &= \begin{pmatrix} I & 0 & 0 \\ \hat{\sigma}^z & 0 & 0 \\ -h\hat{\sigma}^x & -J\hat{\sigma}^z & I \end{pmatrix} , j \neq 1 \,,\ L \,, \\
W^{[L]} &= \begin{pmatrix} I \\ \hat{\sigma}^z \\ -h\hat{\sigma}^x \end{pmatrix} .
\end{aligned} \tag{E.18}$$

We note that the MPO has a bond dimension of 3.[118] The reader should convince themselves that the MPO form given reproduces the correct Hamiltonian. A simple check is to compute the two-site hamiltonian using only the boundary operators and then the three site Hamiltonian with an additional matrix between them etc. with the understanding that the matrices that are elements of the matrices W are multiplied according to the tensor product. This method also suggests a straightforward way to prove the consistency by induction for those who wish to be more thorough.

E.2.1 Step 4: MPO Structure

Introduce a `matrix` structure which is an allocatable array with two indices. Then, define an `MPOm` structure which is an allocatable array of matrices with two indices. This represents the on-site MPO tensors $W_{kk'}^{ii'}$, with i and i' being the indices of the on-site operator. Finally, define an `MPO` structure which is an allocatable array of `MPOm`s.

---

[118] The general case given above has a bond dimension of $p + 2$, *independent* of the number of on-site operators.

E.2.2 Step 5: MPO Representation of the Ising Hamiltonian

Write a subroutine `IsingMPO(H,L,Jz,hx)` which creates an `MPO H` on `L` sites containing the MPO representation of the Ising model with J=`Jz` and h=`hx` according to Eq. (E.18). Additionally, write a routine `DeallocateMPO(H)` which deallocates the `MPO H` which was allocated by the `IsingMPO` routine.

E.2.3 * Long Range Interactions with MPO.

Just as exponentially decaying correlations are naturally supported by an MPS with a constant bond dimension, interactions with exponentially decaying weight may be easily expressed as an MPO with a constant (independent of system size) bond dimension. As an example, a long range Ising model

$$\hat{H} = -J\sum_{i<j} e^{-\xi|i-j-1|}\hat{\sigma}_i^z\hat{\sigma}_j^z - h\sum_i \hat{\sigma}_i^x \tag{E.19}$$

can be expressed as an MPO as

$$\begin{aligned} W^{[1]} &= \begin{pmatrix} -h\sigma^x & -J\hat{\sigma}^z & I \end{pmatrix}, \\ W^{[j]} &= \begin{pmatrix} I & 0 & 0 \\ \hat{\sigma}^z & e^{-\xi} & 0 \\ -h\hat{\sigma}^x & -J\hat{\sigma}^z & I \end{pmatrix}, j \neq 1,\ L, \\ W^{[L]} &= \begin{pmatrix} I \\ \hat{\sigma}^z \\ -h\hat{\sigma}^x \end{pmatrix}. \end{aligned} \tag{E.20}$$

Power-law decaying functions can be approximated by using sums of exponentials in the following manner. We have some function $f\,(i-j-1)$ and we want to approximate it by the sum of k weighted exponentials $\sum_{i=1}^{k}\alpha_k\beta_k^{i-j-1} = \sum_{i=1}^{k}\alpha_k e^{(i-j-1)\log\beta_k}$. We do so in the least squares sense, that is we find the α_k and β_k such that

$$\left| f\,(x) - \sum_{i=1}^{k}\alpha_k\beta_k^x \right|^2 \tag{E.21}$$

is minimized in the desired (integer valued) domain of x. This can be done with standard numerical packages such as MINPACK. My own implementation in Python follows:

```
from math import sqrt
import numpy as np
from scipy.optimize import leastsq
def fittoexpSum(func,L,maxnterms,tol):
    """Fit a sum of exponentials \sum_n a_n b_n**{x-1} to the
       function func
     across the range [1:L].  maxnterms is the maximum number
        of exponentials allowed
     and tol is the tolerance used to obtain the actual number
         of terms."""
    x=np.linspace(1,L,L)
    #function evaluated at x-points
    y= func(x)
    fail=True
    for n in range(1,maxnterms+1):
        #residual function
        resid = lambda p, x, y : sumexp(p,x)-y
        #initial guess
        p0=np.zeros(2*n)
        if n==1:
            for i in range(2*n):
                p0[i]=0.1
        else:
            #use old values to refine guess
            p0[:2*(n-1)]=p
            p0[2*(n-1):]=0.1
        p,cov,infodict,mesg,ier=leastsq(resid, p0,args=(x,y),ftol
           =tol,gtol=tol, maxfev=100000,full_output=1,warning=
           True)
        if np.dot(infodict['fvec'],infodict['fvec'])<tol:
            print 'resid',np.dot(infodict['fvec'],infodict['fvec'
                ])
            fail=False
            break
    if fail :
        raise Exception("Unable␣to␣converge␣decayingFunction␣to␣
           the␣desired␣tolerance␣with␣the␣given␣number␣of␣terms!␣
           ␣Try␣increasing␣maxnterms␣or␣decreasing␣tol!")
    return p

def sumexp(p,x):
    """Return y=\sum_n a_n b_n^{r-1}."""
```

```
    val=np.double(0.0)
    nterms=len(p)/2
    for n in range(nterms):
        val=val+p[2*n]*(p[2*n+1]**(x-1))
    return val
```

The resulting MPO has the same overall format as the exponential Ising MPO, with the border terms and the diagonals nonzero, but now there are k such nonzero terms instead of just 1. The number of terms which must be kept depends on the particular form of f and the domain of x.

E.3 Construction of the Effective Hamiltonian

We now have the state and Hamiltonian in the form appropriate for variational MPS studies. We now wish to variationally optimize the MPS tensors A one at a time until a global optimum has been reached. We optimize the state in the usual way, by changing its parameters such that the total energy functional

$$E\left[|\psi\rangle\right]=\frac{\langle\psi|\hat{H}|\psi\rangle}{\langle\psi|\psi\rangle} \tag{E.22}$$

is minimized. As shown in Schollwöck, minimization with respect to a single tensor A chosen to be the orthogonality center leads to the linear eigenequation

$$\sum_{\alpha',i',\beta'}\hat{H}^{\mathrm{eff}}_{[\alpha i\beta][\alpha' i'\beta']}A^{i'}_{\alpha'\beta'}=\lambda A^{i}_{\alpha\beta}\,, \tag{E.23}$$

where the effective Hamiltonian $\hat{H}^{\mathrm{eff}}$ is defined by the tensor network diagram shown in Figure E.3. We call this object the effective Hamiltonian because it gives the energies when all tensors except those at site j are held fixed, and so it incorporates nonlocal effects of the many-body state when choosing the minimum energy configuration at a particular site.

The overall algorithm for variationally finding the ground state is outlined in Figure E.4.

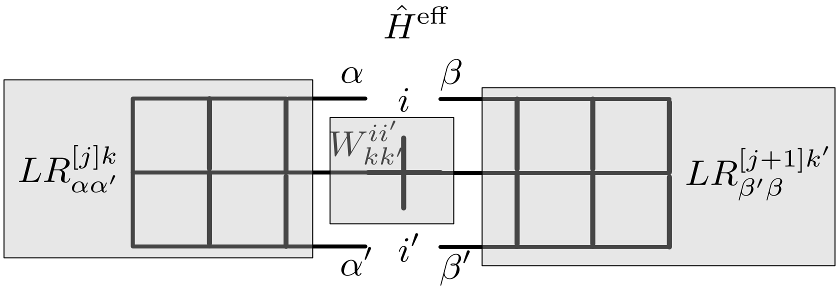


Figure E.3: Construction of the Effective Hamiltonian in tensor network diagram representation

From Figure E.3, we see that construction of the effective Hamiltonian at each site requires $\mathcal{O}(L)$ operations, as we have to contract over all MPS tensors not currently being optimized. Because a complete sweep runs over L sites, this implies that the overall algorithm scales as $\mathcal{O}(L^2)$, which is unacceptably slow. We now turn to how to generate the effective Hamiltonian using $\mathcal{O}(L)$ time. We do so by introducing an array of $L+1$ `tensors`, which we call LR. This object is defined for orthogonality center ℓ via the relations

$$LR^{[1]k}_{\alpha\alpha'} = \delta_{k,1}\delta_{\alpha,1}\delta_{\alpha',1} \tag{E.24}$$

$$LR^{[j]k}_{\alpha\alpha'} = \sum_{i,i',k',\gamma,\gamma'} A^{[j-1]i\star}_{\gamma\alpha} LR^{[j-1]k'}_{\gamma\gamma'} A^{[j-1]i'}_{\gamma'\alpha'} W^{[j-1]ii'}_{k'k}\,,\quad 2 \leq j \leq \ell \tag{E.25}$$

$$LR^{[L+1]k}_{\beta'\beta} = \delta_{k,1}\delta_{\beta',1}\delta_{\beta,1} \tag{E.26}$$

$$LR^{[j]k}_{\beta'\beta} = \sum_{i,i',k',\gamma,\gamma'} A^{[j]i'}_{\beta'\gamma'} LR^{[j+1]k'}_{\gamma'\gamma} A^{[j]i\star}_{\beta\gamma} W^{[j]ii'}_{kk'}\,,\quad \ell+1 \leq j \leq L\,. \tag{E.27}$$

Very special attention should be paid to the ordering of indices in each term. While the order of indices may seem arbitrary, the particular choice shown here will turn out to be beneficial for later optimization. We now note that the effective Hamiltonian at site j can be constructed from these overlaps as

$$\hat{H}^{\mathrm{eff}}_{[\alpha i\beta][\alpha' i'\beta']} = \sum_{k,k'} LR^{[j]k}_{\alpha\alpha'} W^{ii'}_{kk'} LR^{[j+1]k'}_{\beta'\beta}\,, \tag{E.28}$$

```
1. Create a random MPS with fixed bond dimension, normalize it, and shift the orthogonality center to some user-specified site $j$. Set $k = j$
2. Create the MPO representation of the Hamiltonian.
3. (Begin sweeping procedure: right sweep part 1) Create the effective Hamiltonian at site $k$, diagonalize it, and replace $A$ with the eigenvector corresponding to the lowest eigenvalue.
4. Move the orthogonality center one site to the right. $k = k + 1$
5. Repeat 3 and 4 until site $L$ is the orthogonality center.
6. (sweeping procedure: left sweep) Create the effective Hamiltonian at site $k$, diagonalize it, and replace $A$ with the eigenvector corresponding to the lowest eigenvalue.
7. Move the orthogonality center one site to the left. $k = k - 1$
8. Repeat 6 and 7 until site 1 is the orthogonality center.
9. (sweeping procedure: right sweep part 2) Repeat 3 and 4 until site $j$ is the orthogonality center.
10. Test for convergence (to be discussed) and either repeat from 3 or exit the loop.
```

Figure E.4: Pseudocode to find the ground state variationally.

as shown in Figure E.3. Once the present site has been variationally optimized and the orthogonality center shifted to the next site, we update the j^{th} element of the LR array using Eq. (E.25) if we are sweeping to the right (steps 3/4 or 9) or Eq. (E.27) if we are sweeping to the left (step 6/7). Note also that we must initialize the LR array for the beginning orthogonality center j.

E.3.1 Step 6: LR Array

Create a routine `InitializeLR(LR,H,psi,k)` which initializes the $L+1$-dimensional array of `tensor`s `LR` according to the above relations for orthogonality center `k`. Also, write a routine `UpdateLR(LR,H,psi,k,sense)` which updates the $k+1^{th}$ element of `LR` according to Eq. (E.25) if the integer `sense` is greater than 0 and which updates the k^{th} element of `LR` according to Eq. (E.27) if `sense` is less than 0.[119] These routines allow us to "recycle" old values of `LR` and keep the algorithm scaling as $\mathcal{O}(L)$.

E.3.2 Step 7: Effective Hamiltonian

Create a routine `EffectiveHamiltonian(j,Heff,H,LR)` which creates the `matrix` `Heff` at site `j` according to Eq. (E.28). Create another routine `MinimizeSite(j,psi,H, LR, energy)` which creates the effective Hamiltonian at site `j`, diagonalizes it,[120] outputs the lowest eigenvalue as `energy`, and replaces `psi` with the eigenvector corresponding to the lowest energy (be careful with indices).

E.3.3 Step 8: Putting It All together

Write a routine `FindGroundState(H,psi,k)` which takes as input an `MPO` Hamiltonian `H`, a random `MPS` `psi` with orthogonality center `k` and variationally optimizes it to the ground state of `H`. We do this in several parts. First, call `InitializeLR(LR,H,psi,k)`

---

[119] Note the asymmetry between which element we update based on the direction of the sweep. This arises from the fact that we use $LR^{[j]}$ and $LR^{[j+1]}$ to update site j.

[120] Because the effective Hamiltonian is real and symmetric(prove this to yourself) you can use the LAPACK routine DSYEV.

and set the logical flag `converged` to `.FALSE.`, `energySave=10000.0`, and `numsweeps=1`. We then have the following psuedocode recursion:

```
DO WHILE(.not.converged)
     sense=1
     DO j=k,L-1
          CALL MinimizeSite(j,psi,H,LR,energy)
          CALL OrthogonalizeMPS(psi,j+1,j,j+1)
          CALL UpdateLR(LR,H,psi,j,sense)
     END DO
     sense=-1
     DO j=L,2,(-1)
          CALL MinimizeSite(j,psi,H,LR,energy)
          CALL OrthogonalizeMPS(psi,j-1,j-1,j)
          CALL UpdateLR(LR,H,psi,j,sense)
     END DO
     sense=1
     DO j=1,kin-1
          CALL MinimizeSite(j,psi,H,LR,energy)
          CALL OrthogonalizeMPS(psi,j+1,j,j+1)
          CALL UpdateLR(LR,H,psi,j,sense)
     END DO
     IF(ABS(energySave-energy)<0.01) converged=.true.
     IF(numsweeps==10) EXIT
     IF(converged) EXIT
     numsweeps=numsweeps+1
     energySave=energy
END DO
```

Compare with Figure E.3. After convergence is reached, print the ground state energy and whether it converged to the screen. Write a main program which creates a random MPS, creates the MPO representation of the Ising Hamiltonian with user specified J_z and h_x, and then finds the ground state energy. Compare your results with the results of the attached code. The main program is `IsingGSMain.f90` which is compiled as `make IsingGSMain` and run as `./Execute_IsingGSMain`.

E.3.4 $*$ Sparse Eigensolvers-the Lanczos Algorithm

Looking again at Figure E.3, we see that the application of the effective Hamiltonian onto some MPS tensor A may be broken into three pieces:

$$F^{k'}_{\alpha i\beta}=\sum_{\beta'}A^{[j]i}_{\alpha\beta'}LR^{[j+1]k'}_{\beta'\beta}\;,\quad G^{k}_{\alpha i\beta}=\sum_{k'i'}W^{ii'}_{kk'}F^{k'}_{\alpha i'\beta}\,,\quad \left[\hat{H}_{\mathrm{eff}}A\right]^{i}_{\alpha\beta}=\sum_{\alpha'k}LR^{[j]k}_{\alpha\alpha'}G^{k}_{\alpha i\beta}\,. \tag{E.29}$$

The relative scalings of these contractions are $\chi^3 d\chi_O$, $\chi_O^2 d^2\chi^2$, and $\chi^3 d\chi_O$, where χ is the bond dimension of the MPS, d the local dimension, and χ_O the bond dimension of the MPO. On the other hand, the direct construction of the effective Hamiltonian, Eq. (E.28), scales as $\chi^4\chi_O^2 d^2$, which is *much* slower than a single Hamiltonian-tensor multiply, and the diagonalization of the Hamiltonian scales as $\chi^6 d^3$, which is incomparably slower than either of these operations when $\chi \gg 1$, as is typically the case. Luckily, there exist eigensolvers for one or a few of the extremal eigenvalues which require only Hamiltonian-tensor multiplies. The *Lanczos algorithm* is an example of such an alogiritthm. It may be stated as follows:[121]

```
input psi, a normalized vector (tensor)
input H, an effective Hamiltonian in the LR/W format
input epsilon, a tolerance for convergence

k=1
lanc=psi
v=H*psi
alpha(k)=psi*v
v=v-alpha(k)*psi
beta(k)=sqrt(v*v)
do until convergence
        temp=psi
        psi=v/beta(k)
        v=-beta(k)*temp
        v=v+H*psi
        k=k+1
        alpha(k)=psi*v
        v=v-alpha(k)*psi
        beta(k)=sqrt(v*v)
        solve alpha/beta system and test for convergence
end
```

---

[121] A great deal more can be said about the origins and interpretation of the Lanczos algorithm. See for example Trefethen and Bau's book "Numerical Linear Algebra" or Golub and Van Loan's book "Matrix Computations." Here, for the sake of compactness, we just outline the algorithm.

```
!generate eigenvector
psi=eigvec(1)*lanc
v=0
v=H*lanc
alpha(k)=lanc*v
v=v-alpha(k)*lanc
beta(k)=sqrt(v*v)
do i=1,k-1
        temp=lanc
        lanc=v/beta(k)
        v=-beta(k)*lanc
        psi=psi+eigvec(i+1)*lanc
        v=v+H*lanc
        k=k+1
        alpha(k)=lanc*v
        v=v-alpha(k)*lanc
        beta(k)=sqrt(v*v)
end
renormalize psi
```

Where $*$ for two tensors is the inner product of the Hilbert space $\langle A, B\rangle \equiv A * B = \sum_{\alpha i \beta} A^{i\star}_{\alpha\beta} B^{i}_{\alpha\beta}$ and for Hamiltonian-tensor multiplies is the three-step process Eq. (E.29). The section `solve alpha/beta system` at the k^{th} step means to solve for the lowest eigenvalue/eigenvector pair of the symmetric tridiagonal matrix with `alpha(1:k)` on the diagonals and `beta(1:k-1)` on the offdiagonals. `evec(i)` in the above represents the i^{th} component of this eigenvector. Specialized routines for tridiagonal problems exist in lapack. *Exact* convergence occurs when one of the `beta(k)` is zero, to within numerical tolerance. Convergence within a specified tolerance ε, in the sense that $\left|\hat{H}_{\text{eff}} A - \lambda A\right| < \varepsilon$, occurs when $|e_k \beta_{k-1}| < \varepsilon$, where e is the eigenvector of the alpha/beta matrix at the k^{th} iteration and β is `beta`.

The Lanczos algorithm often converges in a constant (independent of χ) number of iterations on the order of a few tens or hundreds. This leads to an algorithm which scales as χ^3 overall, *much* faster than the χ^6 scaling of the direct algorithm. The Lanczos algorithm should thus be considered an essential part of any efficient MPS program.

E.4 Observables

The simplest observables to compute are those at a single site: $\langle \hat{O}^j \rangle$. If that site is the orthogonality center then we have

$$\langle \hat{O}^j \rangle = \sum_{\alpha,\beta,i,i'} A^{[j]i\star}_{\alpha\beta} \hat{O}^j_{ii'} A^{[j]i'}_{\alpha\beta} \,. \tag{E.30}$$

To test these routines, compute $\sum_i \langle \hat{\sigma}^i_z \rangle$ and demonstrate that it is 0 (by symmetry) and compute $\sum_i \langle \hat{\sigma}^i_x \rangle$ and show it is $-\partial\langle \hat{H} \rangle/\partial h = -\left[E\left(h+\delta h\right) - E\left(h-\delta h\right)\right]/2\delta h + \mathcal{O}\left(\delta h^2\right)$. To compute the expectation of a general operator expressed as an MPO you can use the `LR` recursion

$$LR^{[L+1]k}_{\beta'\beta} = \delta_{k,1}\delta_{\beta',1}\delta_{\beta,1} \tag{E.31}$$

$$LR^{[j]k}_{\beta'\beta} = \sum_{i,i',k',\gamma,\gamma'} A^{[j]i'}_{\beta'\gamma'} LR^{[j+1]k'}_{\gamma'\gamma} A^{[j]i\star}_{\beta\gamma} W^{[j]ii'}_{kk'} \,, \quad j = 1 \dots L \,, \tag{E.32}$$

the expectation is $LR^{[1]1}_{11}$ (note that this tensor is $1 \times 1 \times 1$).

E.5 Excited States

To compute the n^{th} excited state, we again minimize the energy functional

$$E\left[|\psi\rangle\right] = \frac{\langle\psi|\hat{H}|\psi\rangle}{\langle\psi|\psi\rangle} \tag{E.33}$$

but this time subject to the constraints $\langle\psi|\varphi_\gamma\rangle = 0$, $\gamma = 1, \dots, n$, where $|\varphi_\gamma\rangle$ is the γ^{th} lowest energy state already obtained by the ground state procedure above or the current procedure. We note that each one of the constraints is a *linear form* in each of the tensors $A^{[\ell]\star}$. Thus, we can represent the constraints as[122]

$$\langle\psi|\varphi_\gamma\rangle = \sum_{\alpha i\beta} A^{[\ell]i\star}_{\alpha\beta} \mathbb{B}^{[\ell](\gamma)i}_{\alpha\beta} = 0 \,. \tag{E.34}$$

where

---

[122] Note that the bases of $\mathbb{B}$ are the same as the bases of A, as can be clearly seen from Figure E.5.

$$\mathbb{B}^{[\ell](\gamma)i}_{\alpha\beta} \equiv \frac{\partial}{\partial A^{[\ell]\star}} \langle \psi | \phi_\gamma \rangle = LR^{[\ell]\gamma}_{\alpha\alpha'} B^{[\ell](\gamma)i}_{\alpha'\beta'} LR^{[\ell+1]\gamma}_{\beta'\beta} \,. \tag{E.35}$$

The LR overlaps used in this expression are

$$LR^{[1]\gamma}_{\alpha\alpha'} = \delta_{k,1}\delta_{\alpha,1}\delta_{\alpha',1} \tag{E.36}$$

$$LR^{[j]\gamma}_{\alpha\alpha'} = \sum_{i,\beta,\beta'} A^{[j-1]i\star}_{\beta\alpha} LR^{[j-1]\gamma}_{\beta\beta'} B^{[j-1](\gamma)i}_{\beta'\alpha'} \,, \quad 2 \leq j \leq \ell \tag{E.37}$$

$$LR^{[L+1]\gamma}_{\beta'\beta} = \delta_{k,1}\delta_{\beta',1}\delta_{\beta,1} \tag{E.38}$$

$$LR^{[j]\gamma}_{\beta'\beta} = \sum_{i,\alpha,\alpha'} B^{[j](\gamma)i}_{\beta'\alpha'} LR^{[j+1]\gamma}_{\alpha'\alpha} A^{[j]i\star}_{\beta\alpha} \,, \quad \ell+1 \leq j \leq L \,, \tag{E.39}$$

and $B^{[\ell](\gamma)}$ is the on-site tensor at site ℓ for state $|\varphi_\gamma\rangle$. The tensor network diagram representing these objects is shown in Figure E.5. We use the $\mathbb{B}^{(\gamma)}$ to construct projectors into the subset of states orthogonal to the n lowest states

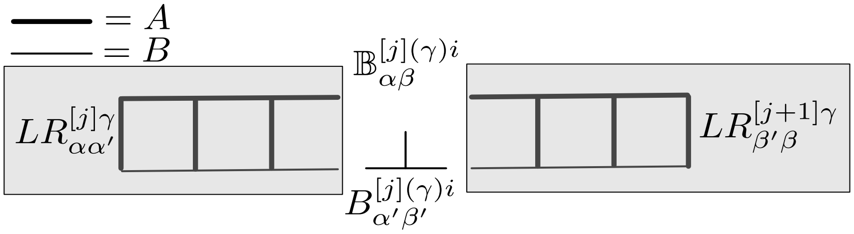


Figure E.5: Construction of linear forms for projection in tensor network diagram representation. The thick lines correspond to contractions over the bond dimensions of A and the thin lines are contractions over the bond dimensions of B.

$$\mathcal{P}^{[\ell]} = 1 - \sum_{\gamma\gamma'} \mathbb{B}^{[\ell](\gamma)} \left(\mathcal{N}^{-1}\right)_{\gamma\gamma'} \mathbb{B}^{[\ell](\gamma')\dagger} \tag{E.40}$$

where $\mathcal{N}^{-1}$ is the inverse of the Gram matrix

$$\mathcal{N}_{\gamma\gamma'} = \sum_{\alpha i \beta} \mathbb{B}^{[\ell](\gamma)i\star}_{\alpha\beta} \mathbb{B}^{[\ell](\gamma')i}_{\alpha\beta} \,. \tag{E.41}$$

The Gram matrix inverse is required for the projector to be idempotent: $\mathcal{P}^2 = \mathcal{P}$. While the n lowest states are all orthogonal the $\mathbb{B}$ are not guaranteed to be orthogonal, and may even be linearly dependent.[123] Thus, instead of the true inverse we use the Moore-Penrose pseudo-inverse of the (Hermitian) Gram matrix.

$$\left(\mathcal{N}^{-1}\right)_{\gamma\gamma'} = \sum_{\beta=1}^{N_S} V_{\gamma\beta} \frac{1}{\lambda_\beta} V^{\star}_{\gamma'\beta} \tag{E.42}$$

where V is the matrix with the eigenvectors of $\mathcal{N}$ as columns and N_S is the number of eigenvalues greater than a set tolerance (to avoid overflow).[124] By defining the new set of tensors

$$\mathbb{C}^{[\ell](\gamma)i}_{\alpha\beta} = \sum_{\gamma'} \frac{1}{\sqrt{\lambda_\gamma}} V_{\gamma'\gamma} \mathbb{B}^{[\ell](\gamma')i}_{\alpha\beta} \tag{E.43}$$

the projector becomes

$$\mathcal{P}^{[\ell]}_{[\alpha i\beta],[\alpha' i'\beta']} = \delta_{[\alpha i\beta],[\alpha' i'\beta']} - \sum_{\gamma} \mathbb{C}^{[\ell](\gamma)i}_{\alpha\beta} \mathbb{C}^{[\ell](\gamma)i'\star}_{\alpha'\beta'} \,. \tag{E.44}$$

We compute the projected effective Hamiltonian as $\mathcal{P}^{[\ell]\dagger}\hat{H}_{\text{eff}}\mathcal{P}^{[\ell]}$. The total algorithm proceeds much like the ground state search algorithm except that we also initialize the $\mathbb{B}$ overlaps LR using Eqs. (E.37) and (E.39), compute the projectors at each site to compute and diagonalize the projected effective Hamiltonian, and update the $\mathbb{B}$ LR after each optimization.[125] As a final note, clearly 0 is a valid eigenvalue of the projected Hamiltonian with eigenvector 0. Thus, if the next excited state energy is positive then the algorithm will fail to converge trying to force the state to be zero. This difficulty can be avoided by shifting the spectrum of the Hamiltonian to be negative definite.[126]

---

[123] One can see that if the number of states desired is larger than the local Hilbert space of a single MPS tensor then the vectors *must* be linearly dependent just by the dimension of the space.

[124] A good choice for the tolerance is $n\|\mathcal{N}\|_2\varepsilon = n\sqrt{\max\lambda}\varepsilon$ where ε is the machine epsilon, n is the linear dimension of the matrix, and $\sqrt{\max\lambda}$ is the square root of the largest eigenvalue. Note that because $\mathcal{N}$ is a Gram matrix it is guaranteed to be positive semidefinite and have at least one positive eigenvalue.

[125] Note that we don't need to shift the orthogonality center of the Bs, as all of them get contracted each time we form the $\mathbb{B}$s.

[126] Note that the largest eigenvalue of $\hat{H}$ can be found by solving for the "ground state" of $-\hat{H}$.

E.5.1 $*$ Sparse Solution of the Projected Eigenproblem

The above method can also easily be recast in a form amenable to the Lanczos algorithm. Every Hamiltonian-tensor multiply is replaced by projection, Hamiltonian-tensor multiply, and then projection again, where projection is

$$A^{i}_{\alpha\beta} = A^{i}_{\alpha\beta} - \sum_{\gamma}\left[\sum_{\alpha'\beta' i'} A^{i'}_{\alpha'\beta'}\mathbb{C}^{[\ell](\gamma)i'\star}_{\alpha'\beta'}\right]\mathbb{C}^{[\ell](\gamma)i}_{\alpha\beta}\,. \tag{E.45}$$

E.6 Time Evolution with TEBD

In this section we discuss the simplest method for time evolution of an MPS: Time-evolving block decimation (TEBD). TEBD involves two parts: 1.the approximation of the full propagator by a series of two-site propagators for short times and 2. representation of a two-site operator times an MPS as an MPS. There are many ways to accomplish the first task which fall generally under the heading of Suzuki-Trotter expansions. A simple expression which fits our purposes nicely is

$$\mathcal{U} = \exp\left(-i\hat{H}\delta t\right) = \prod_{j=L-1}^{1}\exp\left(-i\hat{H}^{(j)}\delta t/2\right)\prod_{j=1}^{L-1}\exp\left(-i\hat{H}^{(j)}\delta t/2\right) + \mathcal{O}\left(\delta t^{3}\right) \tag{E.46}$$

$$= \prod_{j=L-1}^{1}\mathcal{U}^{(j)}\prod_{j=1}^{L-1}\mathcal{U}^{(j)} + \mathcal{O}\left(\delta t^{3}\right) \tag{E.47}$$

where $\hat{H}^{(j)}$ is a *two-site* Hamiltonian acting on sites j and $j+1$ whose precise form will be determined shortly. We first note that this restriction implies that TEBD is restricted to nearest-neighbor Hamiltonians. As was discussed in the MPO section, the MPO of any Hamiltonian with only nearest-neighbor and local interactions has nonzero entries only on the "border" of the MPO matrices. We construct the Hamiltonians in Eq. (E.46) as

$$\hat{H}^{(1)}_{[ij][i'j']} = \sum_{k=1}^{\chi_O-1} \left(W^{[1]}\right)^{ii'}_{1k} \left(W^{[2]}\right)^{jj'}_{k1} + \frac{1}{2} \left(W^{[1]}\right)^{ii'}_{1\chi_O} \left(W^{[2]}\right)^{jj'}_{\chi_O 1} , \tag{E.48}$$

$$\begin{aligned} \hat{H}^{(\ell)}_{[ij][i'j']} =& \frac{1}{2} \left(W^{[\ell]}\right)^{ii'}_{11} \left(W^{[\ell+1]}\right)^{jj'}_{11} + \sum_{k=2}^{\chi_O-1} \left(W^{[\ell]}\right)^{ii'}_{1k} \left(W^{[\ell+1]}\right)^{jj'}_{k1} \\ &+ \frac{1}{2} \left(W^{[\ell]}\right)^{ii'}_{1\chi_O} \left(W^{[\ell+1]}\right)^{jj'}_{\chi_O 1} , \quad 2 \le \ell \le L-2 \end{aligned} \tag{E.49}$$

$$\hat{H}^{(L-1)}_{[ij][i'j']} = \frac{1}{2} \left(W^{[L-1]}\right)^{ii'}_{11} \left(W^{[L]}\right)^{jj'}_{11} + \sum_{k=2}^{\chi_O} \left(W^{[L-1]}\right)^{ii'}_{1k} \left(W^{[L]}\right)^{jj'}_{k1} . \tag{E.50}$$

Note that, with these definitions, $\hat{H} = \sum_{j=1}^{L-1} \hat{H}^{(j)}$. The factors of 1/2 in the local terms with $k=1$ or χ_O account for the fact that $\hat{H}^{(\ell)}$ and $\hat{H}^{(\ell+1)}$ both contribute a local term at site ℓ, and the boundary Hamiltonians account for the fact that only $\hat{H}^{(1)}$ and $\hat{H}^{(L-1)}$ contain local terms for the first and last site, respectively. With these Hermitian matrices in hand we can easily find the exponentials appearing in Eq. (E.46) by performing the eigenvalue decomposition $H = V\lambda V^{\dagger}$, where V is the matrix with the eigenvectors of H as columns and λ is a diagonal matrix with the eigenvalues on the diagonal. The matrix exponential $\exp\left(-iH\delta t\right)$ is $V\exp\left(-i\lambda\delta t\right)V^{\dagger}$ in the same notation.

Our task of applying the full propagator $\mathcal{U}$ to a site $|\psi\rangle$ expressed as an MPS has been reduced to computing the application of a two-site operator to an MPS. We do this in the following manner. We first define a new `fourtensor` structure which has four indices. We then fuse two tensors A and B into a fourtensor T as

$$T_{\alpha ij\beta} = \sum_{\gamma} A^{i}_{\alpha\gamma} B^{j}_{\gamma\beta} . \tag{E.51}$$

The operation of a two-site operator $\mathcal{U}$ on this object is

$$\tilde{T}_{\alpha ij\beta} = \sum_{i',j'} \mathcal{U}_{[ij][i'j']} T_{\alpha i'j'\beta} . \tag{E.52}$$

Finally, a fourtensor may be split into two tensors by first reshaping it into a $d_\alpha d_i \times d_j d_\beta$ matrix:

$$T_{[\alpha i][j\beta]} = T_{\alpha i j \beta} \tag{E.53}$$

performing an SVD

$$T_{[\alpha i][j\beta]} = U_{[\alpha i]\gamma} S_\gamma V_{\gamma[j\beta]} \tag{E.54}$$

and then unpacking the SVD matrices

$$A^i_{\alpha\gamma} = U_{[\alpha i]\gamma}\,, \quad B^j_{\gamma\beta} = S_\gamma V_{\gamma[j\beta]}\,. \tag{E.55}$$

As written, if either A or B was the orthogonality center then B is the new orthogonality center. One can make A the orthogonality center by contracting S into A instead of B:

$$A^i_{\alpha\gamma} = U_{[\alpha i]\gamma} S_\gamma\,, \quad B^j_{\gamma\beta} = V_{\gamma[j\beta]}\,. \tag{E.56}$$

The growth of the number of singular values is typically exponential in the number of time steps, and this causes the bond dimensions of the tensors A and B to also grow exponentially. We truncate the bond dimension optimally by truncating the singular values. In simulations it is often better to impose a condition on the decay of the singular values rather than to impose a condition on the bond dimension itself. That is, we want to discard at most ϵ of the norm of the state at each two site operation. This defines the new bond dimension χ_{new} implicitly as

$$1 - \sum_{\alpha=1}^{\chi_{\text{new}}} \frac{S_\alpha^2}{\sum_\beta S_\beta^2} < \epsilon\,. \tag{E.57}$$

Note that the sum in the denominator runs over *all* values of S, and so the sum of the squares is normalized to 1. Once we have χ_{new}, we define

$S = S(1:\chi_{\text{new}})/\sqrt{\langle S(1:\chi_{\text{new}})|S(1:\chi_{\text{new}})\rangle}$ and the γ summations in Eqs. (E.55) and (E.56) run from 1 to χ_{new}.

The procedure for TEBD with a time-independent Hamiltonian is thus as follows:

1. Shift the orthogonality center to the first site. $t = 0$. If the MPS structure is real, transfer to a complex MPS representation (be careful in your complex code that you have implemented all conjugates properly-this is an easy mistake to make!).
2. Construct the two-site Hamiltonians $\hat{H}^{(j)}$ as in Eqs. (E.48)-(E.50) and exponentiate them to find the two-site propagators $\hat{\mathcal{U}}^{(j)} = \exp\left(-i\hat{H}^{(j)}\delta t\right)$.
3. Starting from the first site, $j = 1$, fuse sites j and $j + 1$ into a fourtensor T.
4. act with $\hat{\mathcal{U}}^{(j)}$ on T, and split into new updated $A^{[j]}$ and $A^{[j+1]}$ using Eq. (E.55).
5. Repeat 3 and 4 up to and including $j = L - 1$.
6. Starting from $j = L - 1$ fuse sites j and $j + 1$ into a fourtensor T.
7. act with $\hat{\mathcal{U}}^{(j)}$ on T, and split into new updated $A^{[j]}$ and $A^{[j+1]}$ using Eq. (E.56).
8. Repeat 5 and 6 down to and including $j = 1$.
9. $t = t + \delta t$
10. Compute any desired observables. If t is greater than or equal to the maximum time desired, exit. Otherwise, return to step 3.

This completes the discussion of TEBD in the MPS language. It turns out to be a much simpler algorithm than the ground state search.