

The Use of Matrix Product State Representations in Simulating Many Body Quantum Systems

by

Matthew Low

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF SCIENCE

in

The Faculty of Science

(Physics)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

June 2010

© Matthew Low 2010

Abstract

The classical simulation of many body quantum systems has long been of interest to both condensed matter and quantum information theorists. The use of matrix product state representations of the wavefunction allows suitable approximations to be made that permit quantum simulations to be performed in polynomial time rather than exponential in the size of the system. The numerical methods are based on a minimization algorithm that can be used to find the ground state and first excited state energies and their respective matrix product states. Using the matrix product states, other quantities such as magnetization and correlation can also be computed. In this thesis, the theory and numerical techniques necessary for these simulations are studied and applied to the one dimensional Ising model in a transverse field. The results of the model are compared to the analytical solution and analyzed in depth.

Table of Contents

Abstract	ii
Table of Contents	iii
List of Figures	v
Acknowledgements	vii
Dedication	viii
1 Introduction to Many Body Quantum Systems	1
2 Motivation	3
3 Theory	4
3.1 Matrix Product States	4
3.2 Tensor Network Diagrams	7
4 Numerical Methods	10
4.1 Variational Method	10
4.1.1 Minimization	10
4.1.2 Gauge Freedoms	13
4.1.3 First Excited State	14
4.2 Imaginary Time Evolution Block Decimation Method	14
5 Study of Ising Model in a Transverse Field	17
5.1 Theory	17
5.2 Results	18
5.2.1 Energy	18
5.2.2 Magnetization	19
5.2.3 Correlation	24
5.3 Numerical Issues	27
5.3.1 Wrap Around	27
5.3.2 Variable Bond Length	28
5.3.3 Subspace Projection	28
6 Conclusion	30
Bibliography	32

Table of Contents

Appendices

A Mathematics	33
A.1 Schmidt Decomposition	33
A.2 Variational Principle	34
B MATLAB Code	35

List of Figures

3.1	A $\beta \times \alpha \times i$ tensor drawn as part of a tensor network.	7
3.2	The tensor product of A and A' contracting the indices of length β and β' , corresponding to Eq. (3.16)	8
3.3	The tensor product of A and A' contracting the indices of length β and β' , and i and i' , corresponding to Eq. (3.17)	8
3.4	Eq. (3.14) as drawn as a tensor network. The length of the indices have not been drawn for simplicity. The tensors $A^{[1]}$ and $A^{[N]}$ only have two indices as they are on the ends of the spin chain.	9
3.5	Eq. (3.15) as drawn as a tensor network. The length of the indices have not been drawn for simplicity. Each tensor $A^{[i]}$ has the same number and location of bonds illustrating the translational symmetry.	9
3.6	The representation of an inner product. A^* is the complex conjugate of A . This contraction is performed on each tensor of the wavefunction.	9
3.7	The representation of an expectation value. A^* is the complex conjugate of A and h is an operator. This contraction is performed on each tensor of the wavefunction.	9
4.1	The tensor network of $N_{eff} A^{[M]}\rangle$, varied with respect to $\langle A^{[M]} $ which is the tensor missing from the network. The uncontracted indices are α , β , and i	11
4.2	The tensor network of $H_{eff} A^{[M]}\rangle$, varied with respect to $\langle A^{[M]} $ which is the tensor missing from the network. The uncontracted indices are α , β , and i	11
4.3	The tensor network of N_{eff} , varied with respect to $ A^{[M]}\rangle$ where the missing tensors are the tensor associated with site M . The uncontracted indices are α , α' , β , β' , i , and i' . The variational method for OBCs ensures that $N_{eff} = \delta_{\alpha\alpha'}\delta_{\beta\beta'}\delta_{ii'}$	12
4.4	The tensor network of H_{eff} , varied with respect to $ A^{[M]}\rangle$ where the missing tensors are the tensor associated with site M . The uncontracted indices are α , α' , β , β' , i , and i'	12
5.1	The simulated ground state energy and exact ground state energy for $N = 60$ and $D = 8$	20
5.2	The simulated first excited state energy and exact first excited state energy for $N = 60$ and $D = 8$	20
5.3	The simulated energy gap, $E_{fs} - E_{gs}$, and exact energy gap for $N = 60$ and $D = 8$	21
5.4	The absolute error in the ground state energy and the first excited state energy for $N = 60$ and $D = 8$	21
5.5	The simulated energy gap, $E_{fs} - E_{gs}$, and exact energy gap for $N = 80$ and $D = 8$	22
5.6	The absolute error in the ground state energy and the first excited state energy for $N = 80$ and $D = 8$	22
5.7	The magnetization in the ground state for $N = 100$ and $D = 10$	23

List of Figures

5.8	The correlations, according to Eq. (5.12), as a function of h for $r = \frac{N}{2}$ in the ground state for $N = 80$ and $D = 8$	24
5.9	The x correlation, according to Eq. (5.13), as a function of h for $r = 5$, $r = 10$, and $r = 50$ in the ground state for $N = 100$ and $D = 10$	25
5.10	The x correlation, according to Eq. (5.13), as a function of r for $h = 1.05$, $h = 1.30$, and $h = 1.64$ in the ground state for $N = 50$ and $D = 8$	26
5.11	The correlation length, ξ , as a function of h in the ground state for $N = 60$ and $D = 8$	26
5.12	The correlation length, ξ , as a function of h in the ground state for $N = 80$ and $D = 8$	27

Acknowledgements

I would like to thank my supervisors Dr. Tzu-Chieh Wei and Dr. Robert Raussendorf for their direction and guidance, for conveying a knowledge and interest in the topic to me, and for their time spent on the project.

Dedication

I would like to dedicate this thesis to my parents and family who have provided me with unconditional love and support throughout my years as an undergraduate.

Chapter 1

Introduction to Many Body Quantum Systems

Since the start of the twentieth century the formulation of quantum mechanics has had an enormous impact on not only modern science, but also society in general. From the advances in modern electronics to the philosophy of science, quantum mechanics has made a unmistakable mark. Despite its success, however, there are still many aspects of quantum mechanics that not extremely well understood. As far as its philosophical implications go, even Einstein was uncomfortable with the idea of an underlying probabilistic theory. From a physics standpoint, one of the interesting problems arising from quantum mechanics is understanding many body quantum systems. These systems offer insights into condensed matter theory, nuclear physics, and quantum mechanics itself, but they are inherently difficult to solve. In approaching the study of such systems, it is first necessary to define precisely what is meant by a many body quantum system.

A many body quantum system is a physical system consisting of a number of constituents that follow the rules laid out by quantum mechanics. Let us consider a number of atoms placed along a line. Clearly, these atoms obey the laws of quantum mechanics forming a one dimensional quantum system. Since the system is comprised of more than one atom, its interactions become more complex and warrant the title *many body* system. This arrangement of atoms on a line is known as the *open boundary conditions* case, since the atoms on the end of the line experience different interactions, as they only have atoms on one side. Another one dimensional configuration is atoms placed on a ring, which unlike the line exhibits translational symmetry. This case is known as having *periodic boundary conditions*. The particular members of the system can be any quantum mechanical object, the most convenient abstraction of which is simply a *spin*.

The evolution of all quantum systems is described by the Schrödinger equation. The time-independent Schrödinger equation is

$$\mathcal{H}|\psi_n\rangle = E_n|\psi_n\rangle, \tag{1.1}$$

where \mathcal{H} is the Hamiltonian of the system, $|\psi_n\rangle$ are the eigenstates of the system, and E_n are the energies of the associated eigenstates. Even in the one dimensional case, this equation is often analytically unsolvable [1]. For reasons discussed in Chapter 2 we would like to solve the system, in particular for the wavefunction and energy of the ground state and the first excited state. In the absence of an analytical solution, it is necessary to turn to simulation, which is the method presented in this thesis.

Unfortunately, simply simulating a quantum system is not the end of the story. The Hilbert space of the system is the tensor product of the various constituent Hilbert spaces. As a result the number of coefficients required to parameterize the wavefunction of the system grows exponentially in the size of the system. This exponential growth renders the problem of quantum simulation intractable. In fact, quantum simulation is one of the primary interests that Richard Feynman cited when theorizing about quantum computing [2, 3]. While a number of other mo-

tivations are described in Chapter 2, this fact alone makes quantum simulation a very intriguing problem. There are several other interesting connections to physical systems and entanglement that arise from the topic of quantum simulation [4].

One of the first techniques used to simulate quantum systems was the Density Matrix Renormalization Group method (DMRG) as developed by White in 1992 [5–8]. The technique has been very useful in understanding properties of a number of condensed matter systems. While the technique has been the most successful in studying one dimensional systems with nearest neighbor interactions in the zero temperature limit, it has also been applied to other situations with good results [9]. One of the key features of the DMRG method is its use of Matrix Product States (MPS) to characterize the wavefunction [8, 10]. Based on the DMRG technique, another method has been developed to simulate many body quantum systems [11].

This technique, called the Variational MPS method by Verstraete et al., uses the MPS representation of the wavefunction to make a suitable approximation such that the minimal amount of relevant information is lost. An approximation is necessary because, as already discussed, the problem of quantum simulation is intractable in classical computation. As it turns out, however, physical ground states cannot occupy the full extent of their Hilbert space [11]. The approximation corresponds to assigning a finite entanglement content to spins in the ground state. It is precisely, these discoveries that have spurred the advancement and use of the numerical techniques that will be discussed.

Chapter 2

Motivation

The motivation for simulating quantum systems falls roughly into two categories. The first is the insight that simulating quantum systems provides into condensed matter problems [9], while the second is motivated by problems that arise in quantum information [4].

In terms of condensed matter physics, the simulation of many body quantum systems is a very relevant problem. There are many physical systems that arise in condensed matter that are modeled by a Hamiltonian using spin-spin interactions. It is exceedingly uncommon that any of these models can be solved analytically [12], which is why an accurate simulation becomes crucial. The motivation driving the models themselves is that analysis and understanding of very simple systems can still lead to non-trivial results.

For instance, consider Einstein's model of a solid. At the atomic level, Einstein described each atom as a quantum harmonic oscillator which all oscillate at the same frequency. From this model he derived the heat capacity of the solid as a function of temperature, at high temperatures. These results have been experimentally verified to be accurate [13]. Clearly it is not unreasonable to expect the presence of interesting phenomena even for simple models. In this thesis we will discuss one dimension models because they exhibit complex enough behaviour to warrant interesting analysis. In particular the one dimensional Ising model in a tranverse field, using a spin chain with periodic boundary conditions, will be discussed. The methods can be easily extended to examine two dimensional models, such as a surface, and three dimensional models, such as a lattice.

The second primary motivation for the study of these methods comes from quantum information theory. The nominal objective of using the MPS representation of the wavefunction and utilizing the numerical techniques developed is to acheive an efficient and accurate classical simulation of many body quantum systems. If a particular many body quantum system is found to be very accurately simulated classically, then that system will likely not provide any computational speed up if used as a the basis of a quantum computer [4]. The idea is that systems that can be classically simulated do not exhibit enough quantum features to provide resources for genuine quantum computation. Futhermore, one can observe which components of the wavefunction are being rounded or truncated in the approximation and identify those as necessary resources for genuine quantum computation. The MPS representation of wave function, in particular, approximates the wave function by effectively placing an upperbound on the Schmidt number of the wavefunction decomposition. The Schmidt number can be interpreted as a measure of the entanglement present in the system. Vidal used this information to identify entanglement as a necessary, but not sufficient, quantity for genuine quantum computation. Moreover, Vidal derived a lower bound on the amount of entanglement that must be present for a speed up in quantum computation [4].

It follows also that a detailed study of numerical techniques to classically simulate quantum systems will lead to a stronger understanding of the more subtle details in quantum mechanics.

Chapter 3

Theory

3.1 Matrix Product States

We will begin the discussion of Matrix Product State (MPS) representations with a derivation of the form that illustrates its validity and demonstrates where the appropriate approximation is made to achieve polynomial time simulation. We will follow the steps outlined by Vidal [4]. Let us first consider a quantum state $|\psi\rangle$ which consists of two spins $|i_1\rangle$ and $|i_2\rangle$ ¹. The most generic wavefunction can be represented as

$$|\psi\rangle = \sum_{i_1=0}^d \sum_{i_2=0}^d c_{i_1 i_2} |i_1\rangle |i_2\rangle, \quad (3.1)$$

where d is the dimension of $|i_i\rangle$. In principle each $|i_i\rangle$ may have a different dimension, but for simplicity we will assume they are all dimension d . If the states $|i_i\rangle$ are not entangled, then the coefficient $c_{i_1 i_2}$ can be written as the product of two complex scalars c_{i_1} and c_{i_2} . For instance, consider a system of two particles each of spin $\frac{1}{2}$. In general the system is written as

$$|\psi\rangle = c_{00}|00\rangle + c_{01}|01\rangle + c_{10}|10\rangle + c_{11}|11\rangle. \quad (3.2)$$

If the coefficients were such that $c_{00} = \alpha\gamma$, $c_{01} = \alpha\delta$, $c_{10} = \beta\gamma$, and $c_{11} = \beta\delta$, then the state could be expressed as a product state, since

$$|\psi\rangle = \alpha\gamma|00\rangle + \alpha\delta|01\rangle + \beta\gamma|10\rangle + \beta\delta|11\rangle \quad (3.3)$$

$$= (\alpha|0\rangle + \beta|1\rangle) \otimes (\gamma|0\rangle + \delta|1\rangle). \quad (3.4)$$

For this assignment of coefficients the state can be expressed as the product of two independent spins. One interpretation of this independence is that if one spin is measured, the outcome of a subsequent measurement on the second spin will not be affected by the outcome of the first. Quantum states, however, are not always product states. For instance, consider the state

$$|\psi\rangle = \alpha|00\rangle + 0|01\rangle + 0|10\rangle + \beta|11\rangle \quad (3.5)$$

$$= \alpha|00\rangle + \beta|11\rangle. \quad (3.6)$$

If $\beta = \pm\alpha = \pm\frac{1}{\sqrt{2}}$, Eq. (3.6) represents the Bell state, which is one example of a maximally entangled state [3]. There is no way to factor this equation into a tensor product of spins and

¹In this thesis, unless otherwise specified, we will use the computational basis $\{|0\rangle, |1\rangle\}$. This basis can be realized as the direct mapping from a spin $\frac{1}{2}$ particle which has the basis in the z orientation $\{|\uparrow\rangle, |\downarrow\rangle\}$. One can then simply relabel $|\uparrow\rangle \Rightarrow |1\rangle$ and $|\downarrow\rangle \Rightarrow |0\rangle$. This is standard notation in quantum information literature for *qubits* as it provides a direct analog to classical bits [3].

therefore it is not a product state. Also, conversely from the product state situation, if one spin is measured, the outcome of a subsequent measurement on the second spin is determined (assuming the second measurement is made right away) by the outcome of the first measurement.

What is worth noticing between these two types of states is the number of coefficients needed to completely describe the state. In the general case the coefficients are $\{c_{i_1 i_2 \dots i_{N-1} i_N}\}$ of which there are d^N of them. The physical interpretation of this is that as more spins are added to the system each new spin could, in principle, be entangled with any of the previous spins. This is an exponential growth in the size of the system and hence intractable for classical simulation. In the special case of the product state, the coefficients are $\{c_{i_1}, c_{i_2}, \dots, c_{i_{N-1}}, c_{i_N}\}$ of which there are dN of them. Intuitively as spins are added to the system, since they are not entangled, one simply needs to add their coefficients to the set of coefficients already used to describe the rest of the system.

The result of this observation is the realization that one way to achieve a tractable quantum simulation would be to deal solely with product states. While general systems cannot be decomposed into product states, perhaps if the coefficients specifying the wavefunction were permitted to decompose as matrices rather than scalars it might be possible to decompose every state into some type of product state. This is precisely the reason for the matrix product state (MPS) representation of the wavefunction.

The MPS representation can be formed efficiently using the Schmidt decomposition [3, 14]. This decomposition allows Eq. (3.1) to be expressed as

$$\sum_{i_1=0}^d \sum_{i_2=0}^d c_{i_1 i_2} |i_1\rangle |i_2\rangle = \sum_{k=1}^{\chi} \lambda_k |\phi_k^{[1]}\rangle \otimes |\phi_k^{[2]}\rangle, \quad (3.7)$$

where $|\phi_k^{[i]}\rangle$ are the spins in some basis that is determined by the Schmidt decomposition, λ_k are the Schmidt values of the decomposition, χ is the *Schmidt number*, and k is some index running over the Schmidt values. The proof of this decomposition is found in Appendix A.1. The Schmidt number, χ , can be thought of as the number of states within $|\phi_k^{[1]}\rangle$ with which $|\phi_k^{[2]}\rangle$ is entangled. Often, the Schmidt number is used as a measurement of the entanglement present in a quantum system [4].

We will now consider a system that, rather than just two states, consists of N states $|i_1\rangle, |i_2\rangle, \dots, |i_N\rangle$, such that

$$|\psi\rangle = \sum_{i_1=0}^d \sum_{i_2=0}^d \dots \sum_{i_N=0}^d c_{i_1 i_2 \dots i_N} |i_1\rangle |i_2\rangle \dots |i_N\rangle. \quad (3.8)$$

Then applying the Schmidt decomposition

$$|\psi\rangle = \sum_{\alpha_1=1}^{\chi} \lambda_{\alpha_1}^{[1]} |\phi_{\alpha_1}^{[1]}\rangle |\phi_{\alpha_1}^{[2 \dots N]}\rangle, \quad (3.9)$$

where the summation index is now labeled α_1 in place of k . The superscripts refer to which of the N states, $|\phi_{\alpha_1}\rangle$, we are referring. In this scheme $|\phi_{\alpha_1}^{[2 \dots N]}\rangle$ represents a system consisting of spins 2 to N in some possibly entangled state. The next step is to re-express $|\phi_{\alpha_1}^{[1]}\rangle$ in the computational basis $|i_1\rangle$

3.1. Matrix Product States

$$|\psi\rangle = \sum_{\alpha_1=1}^{\chi} \sum_{i_1=0}^d \Gamma_{\alpha_1}^{[1]i_1} \lambda_{\alpha_1}^{[1]} |i_1\rangle |\phi_{\alpha_1}^{[2\dots N]}\rangle, \quad (3.10)$$

where $\Gamma_{\alpha_1}^{[1]i_1}$ are the coefficients used in the change of basis. The Schmidt decomposition and change of basis can then be applied to $|\phi_{\alpha_1}^{[2\dots N]}\rangle$, yielding

$$|\psi\rangle = \sum_{\alpha_1=1}^{\chi} \sum_{i_1=0}^d \sum_{\alpha_2=1}^{\chi} \sum_{i_2=0}^d \Gamma_{\alpha_1}^{[1]i_1} \lambda_{\alpha_1}^{[1]} \Gamma_{\alpha_1\alpha_2}^{[2]i_1} \lambda_{\alpha_2}^{[2]} |i_1 i_2\rangle |\phi_{\alpha_2}^{[3\dots N]}\rangle. \quad (3.11)$$

The Schmidt decomposition can then be recursively applied to $|\phi_{\alpha_1}^{[3\dots N]}\rangle$ to separate the rest of the spins. The result is

$$|\psi\rangle = \sum_{i_1, i_2, \dots, i_N} c_{i_1 i_2 \dots i_N} |i_1 i_2 \dots i_N\rangle, \quad (3.12)$$

$$c_{i_1 i_2 \dots i_N} = \sum_{\alpha_1, \alpha_2, \dots, \alpha_{N-1}} \Gamma_{\alpha_1}^{[1]i_1} \lambda_{\alpha_1}^{[1]} \Gamma_{\alpha_1\alpha_2}^{[2]i_2} \lambda_{\alpha_2}^{[2]} \Gamma_{\alpha_2\alpha_3}^{[3]i_3} \dots \lambda_{\alpha_{N-1}}^{[N-1]} \Gamma_{\alpha_{N-1}}^{[N]i_N}, \quad (3.13)$$

which is the MPS representation. Since we have shown that an arbitrary state can be decomposed into a product state of tensors where each tensor is associated with a given basis function, $|i_i\rangle$, we can redefine the tensors however we wish. A notation similar to that of Verstraete, Murg, and Cirac will be adopted [11]

$$|\psi\rangle = \sum_{\alpha_1, \alpha_2, \dots, \alpha_{N-1}, i_1, i_2, \dots, i_{N-1}, i_N} A_{\alpha_1}^{[1]i_1} A_{\alpha_1\alpha_2}^{[2]i_2} \dots A_{\alpha_{N-2}\alpha_{N-1}}^{[N-1]i_{N-1}} A_{\alpha_{N-1}}^{[N]i_N} |i_1 i_2 \dots i_{N-1} i_N\rangle. \quad (3.14)$$

The derivation thus far has been for open boundary conditions as spins 1 and N are identified as the ends of the chain. The extension to periodic boundary conditions simply involves taking a trace over the product of tensors [11]

$$|\psi\rangle = \sum_{\alpha_1, \alpha_2, \dots, \alpha_{N-1}, i_1, i_2, \dots, i_{N-1}, i_N} \text{Tr} \left(A_{\alpha_1}^{[1]i_1} A_{\alpha_1\alpha_2}^{[2]i_2} \dots A_{\alpha_{N-2}\alpha_{N-1}}^{[N-1]i_{N-1}} A_{\alpha_{N-1}}^{[N]i_N} \right) |i_1 i_2 \dots i_{N-1} i_N\rangle. \quad (3.15)$$

The derivation of Eqs. (3.14) and (3.15) has involved no approximations, thus Eqs. (3.14) and (3.15) represent any wavefunction exactly. As mentioned in Chapter 1, however, this parameterization of the wavefunction still scales exponentially with the size of the system. In order to reduce the number of parameters we will need to make an approximation in the representation of these equations.

As discussed in Chapter 1, it has been argued that ground state of the wavefunction cannot utilize the full dimensionality of its Hilbert space. Since the states that will be simulated using the MPS representation are the ground state and first excited state, it is prudent to take advantage of the limited extent of the wavefunction in Hilbert space. The physical interpretation of this is that it is not entirely plausible to expect every single spin in the system to be highly entangled with every other spin in the system. In lieu of this interpretation a reasonable approximation

to make would be to limit the entanglement that can exist between parts of the system. As introduced in Eq. (3.7), χ is used as a measure of the entanglement. Thus the approximation made will be to truncate the Schmidt decomposition to some value D , known as the *bond length*, where $D \leq \chi$. D is known as the bond length because it, in some sense, quantitatively describes the bond between various parts of the system. If D is fixed apriori then as the system grows, the entanglement growth is bounded by D , whereas previously it was bounded by χ which would also grow as the system grew.

With this approximation the number of parameters scales polynomially with the size of the system and the simulation becomes tractable. It is important to note that as $D \rightarrow \chi$ the MPS representation once again becomes exact. In terms of the simulation this is noteworthy because one way to verify the accuracy of the simulation is to compute results for increasing values of D and observing the values remain the same as D increases, an indication the simulation is reliable, or change, an indication that the simulation maybe not be capturing the system interactions accurately.

The actual time complexity for open boundary conditions is now Nd^2D^3 . The d^2D^3 scaling comes from the dimension of the tensors that must be represented or used in intermediate calculations. This is assuming that the calculations are done optimally such that a tensor with four or more indices is never represented². For periodic boundary conditions the time complexity increases to Nd^2D^5 since there are now bonds connecting spins 1 and N which must always be included.

3.2 Tensor Network Diagrams

Due to the large number of indices to keep track of in the MPS representation, it is often cumbersome to show calculations using MPS's. Fortunately there is an alternative pictorial representation called the tensor network. The tensor network formulation represents each tensor itself as a node, also known as sites (since tensors each reside at a specified site). Each index of the tensor is represented by a outgoing edge. The length of the dimension of each index is written on the edge. For instance, Fig. 3.1 shows a $\alpha \times \beta \times i$ tensor, A^3 .

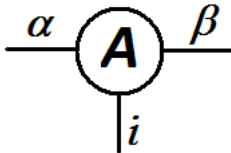


Figure 3.1: A $\beta \times \alpha \times i$ tensor drawn as part of a tensor network.

In order to represent the multiplication of tensors, which is the primary operation used in the calculation of MPS's, edges of tensors are connected. Since tensors can be of arbitrarily high dimensionality, unlike matrix multiplication, the product may sum over an arbitrary number of

²This means that the tensor depicted in Fig. 3.4 is never calculated explicitly, but rather computed as shown in Figs. 3.6 and 3.7.

³Note that a $\beta \times \alpha \times i$ tensor or a tensor with an permutation of the indices would be drawn in the same way. The order of the indices are, in some sense, unimportant because the order is simply used to keep track of which indices to sum over when multiplied by another tensor. This will be kept track of visually in the tensor network by matching edges.

indices. This is represented in the tensor network by the number of edges that are connected. For instance, consider the tensor A from Fig. 3.1 and a second tensor A' , which is $\alpha' \times \beta' \times i'$. One way to multiply these tensors would be to sum over β/β' index

$$[AA']_{iklm} = \sum_{j=1}^{\beta} A_{ijk} A'_{ljm}, \quad (3.16)$$

where β must be equal to β' , and i, k, l , and m are of lengths α, i, α' , and i' respectively. The corresponding tensor network diagram for Eq. (3.16) is shown in Fig. 3.2. In terms of tensor networks, the indices that are summed over are said to be *contracted*.

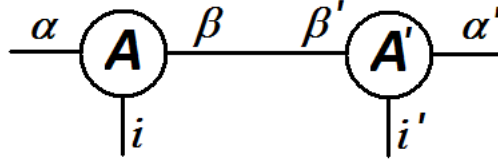


Figure 3.2: The tensor product of A and A' contracting the indices of length β and β' , corresponding to Eq. (3.16)

Another way to multiply the tensors A and A' would be to sum over both the β/β' index and the i/i' index

$$[AA']_{lm} = \sum_{j=1}^{\beta} \sum_{k=1}^i A_{ljk} A'_{mjk}, \quad (3.17)$$

where β must equal β' and i must equal i' . Fig. 3.3 is the corresponding tensor network.

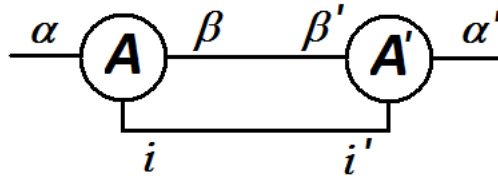


Figure 3.3: The tensor product of A and A' contracting the indices of length β and β' , and i and i' , corresponding to Eq. (3.17)

Following this scheme, the wavefunction in MPS form for open boundary conditions, given by Eq. (3.14), can be drawn as a tensor network as shown in Fig. 3.4. The tensor network for periodic boundary conditions, given by Eq. (3.15), is shown in Fig. 3.5.

Inner products can also be simply represented with tensor networks, as can expectation values, as shown in Figs. 3.6 and 3.7 respectively. The inner product is formed by taking the complex conjugate of A and contracting it with A for each tensor in the wavefunction. The expectation value is a straightforward extension contracting each tensor with opposite edges of an operator, h . To represent an expectation value, the operator h should be the same between each tensor and its complex conjugate for each site in the tensor network.

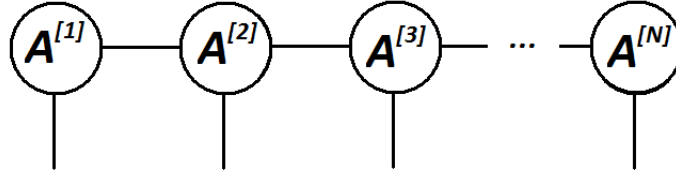


Figure 3.4: Eq. (3.14) as drawn as a tensor network. The length of the indices have not been drawn for simplicity. The tensors $A^{[1]}$ and $A^{[N]}$ only have two indices as they are on the ends of the spin chain.

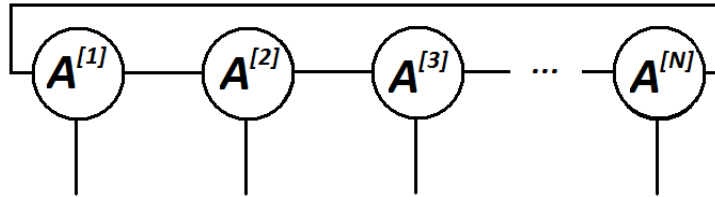


Figure 3.5: Eq. (3.15) as drawn as a tensor network. The length of the indices have not been drawn for simplicity. Each tensor $A^{[i]}$ has the same number and location of bonds illustrating the translational symmetry.

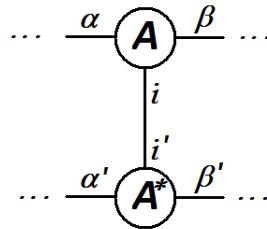


Figure 3.6: The representation of an inner product. A^* is the complex conjugate of A . This contraction is performed on each tensor of the wavefunction.

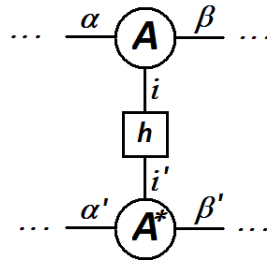


Figure 3.7: The representation of an expectation value. A^* is the complex conjugate of A and h is an operator. This contraction is performed on each tensor of the wavefunction.

Chapter 4

Numerical Methods

4.1 Variational Method

The variational MPS method is one of the most recent developments from a series of numerical methods of simulation quantum systems [11]. The technique itself is fairly general in terms of its range of applicability. The basis upon which the method builds is a minimization sweep which is relevant for finding both the ground state and first excited state⁴. In order to increase the efficiency of the solution, further numerical techniques can be applied to take advantage of gauge freedoms present in the MPS representation. Together these principles detail an efficient, accurate simulation method.

4.1.1 Minimization

In the use of the VMPS method, the primary pieces of information obtained are the ground state energy, first excited state energy, and their respective wavefunctions. Using the wavefunction MPS, one can compute a number of properties of interest in studying the quantum system as will be discussed in Chapter 5.

The variational principle, which is proved in Appendix A.2, states

$$E_{gs} \leq \langle \psi | H | \psi \rangle, \quad (4.1)$$

for any wavefunction $|\psi\rangle$ [15]. If $|\psi\rangle$ is permitted to vary arbitrarily then performing a minimization over the wavefunction will result in a convergence to the ground state. The minimization formulation of Eq. (4.1) is

$$E_{gs} = \min_{|\psi\rangle} \frac{\langle \psi | H | \psi \rangle}{\langle \psi | \psi \rangle}, \quad (4.2)$$

where $\langle \psi | \psi \rangle$ ensures normalization. The application of Lagrange multipliers to Eq. (4.2) yields

$$\tilde{E}_{gs} = \min_{|\psi\rangle} (\langle \psi | H | \psi \rangle - \lambda \langle \psi | \psi \rangle), \quad (4.3)$$

where λ is some scalar⁵. Since in the MPS form one cannot vary the entire wavefunction, Eq. (4.3) is applied locally to each spin, M . The minimization becomes

⁴In fact, higher energy states can be found, however, each successive energy level requires more computation time and since the computation of each energy level depends on the previous levels the error will propagate through each energy level.

⁵Formally the use of Lagrange multipliers under the normalization constraint yields $E_{gs} = \min_{|\psi\rangle} \langle \psi | H | \psi \rangle - \lambda(\langle \psi | \psi \rangle - 1)$, however since the variation will be taken with respect to $\langle \psi |$ rather than λ , the result will be the same.

$$\tilde{E}_{gs} = \min_{|A^{[M]}\rangle} \langle A^{[M]} | H_{eff} | A^{[M]}\rangle - \lambda \langle A^{[M]} | N_{eff} | A^{[M]}\rangle, \quad (4.4)$$

where H_{eff} and N_{eff} are defined as

$$|\psi^{[M]}\rangle = |A^{[1]} \otimes A^{[2]} \otimes \dots \otimes A^{[M-1]} \otimes A^{[M+1]} \otimes \dots \otimes A^{[N-1]} \otimes A^{[N]}\rangle \quad (4.5)$$

$$H_{eff} = \langle \psi^{[M]} | H | \psi^{[M]}\rangle \quad (4.6)$$

$$N_{eff} = \langle \psi^{[M]} | \psi^{[M]}\rangle. \quad (4.7)$$

In the global ground state, the variation of the energy with respect to a given site will be zero:

$$\frac{\delta \tilde{E}_{gs}}{\delta \langle A^{[M]} |} = \langle H_{eff} | A^{[M]}\rangle - \lambda \langle N_{eff} | A^{[M]}\rangle = 0, \quad (4.8)$$

$$\Rightarrow \langle H_{eff} | A^{[M]}\rangle = \lambda \langle N_{eff} | A^{[M]}\rangle. \quad (4.9)$$

Eq. (4.9) is the generalized eigenvalue equation which can be solved numerically with any number of available software packages. Additionally from Eq. (4.9)

$$\begin{aligned} \langle H_{eff} | A^{[M]}\rangle &= \lambda \langle N_{eff} | A^{[M]}\rangle, \\ \langle A^{[M]} | H_{eff} | A^{[M]}\rangle &= \lambda \langle A^{[M]} | N_{eff} | A^{[M]}\rangle, \\ \langle \psi | H | \psi \rangle &= \lambda \langle \psi | \psi \rangle, \\ \langle \psi | H | \psi \rangle &= \lambda, \end{aligned}$$

thus the eigenvalues λ are the energy eigenvalues of the Hamiltonian.

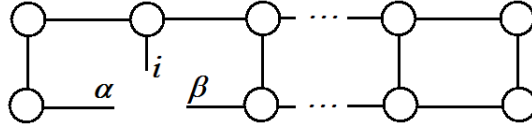


Figure 4.1: The tensor network of $N_{eff} |A^{[M]}\rangle$, varied with respect to $\langle A^{[M]} |$ which is the tensor missing from the network. The uncontracted indices are α , β , and i .

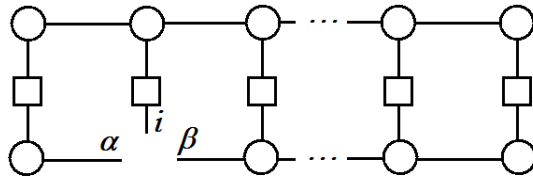


Figure 4.2: The tensor network of $H_{eff} |A^{[M]}\rangle$, varied with respect to $\langle A^{[M]} |$ which is the tensor missing from the network. The uncontracted indices are α , β , and i .

Figs. 4.1 and 4.2 show the tensor network representation of $N_{eff} |A^{[M]}\rangle$ and $H_{eff} |A^{[M]}\rangle$ respectively. In either case, the closed tensor network is represented by $\langle A^{[M]} | N_{eff} | A^{[M]}\rangle$ and

$\langle A^{[M]} | H_{eff} | A^{[M]} \rangle$, thus the variation with respect to the ket of a site, is the full tensor network with the given site removed. Moreover, N_{eff} and H_{eff} themselves are formed by removed the bra of the M th site as shown in Figs. 4.3 and 4.4.

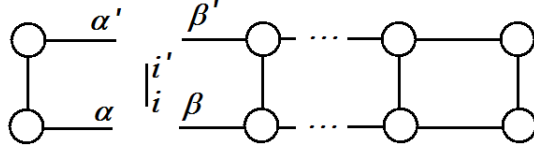


Figure 4.3: The tensor network of N_{eff} , varied with respect to $|A^{[M]}\rangle$ where the missing tensors are the tensor associated with site M . The uncontracted indices are $\alpha, \alpha', \beta, \beta', i,$ and i' . The variational method for OBCs ensures that $N_{eff} = \delta_{\alpha\alpha'}\delta_{\beta\beta'}\delta_{ii'}$.

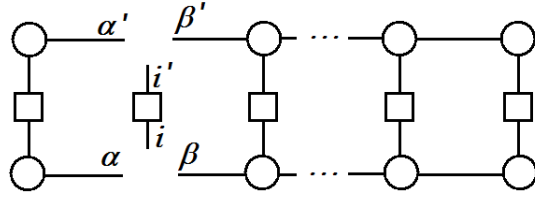


Figure 4.4: The tensor network of H_{eff} , varied with respect to $|A^{[M]}\rangle$ where the missing tensors are the tensor associated with site M . The uncontracted indices are $\alpha, \alpha', \beta, \beta', i,$ and i' .

In order to find the ground state wavefunction and energy a sweeping technique is used in the local minimization to achieve a global minimization. Throughout the calculation it is necessary to store the H_{eff} tensor for all spins. This H_{eff} was then updated with each successive local minimization. The sweeping technique was employed by first performing the minimization on the first spin, $A^{[1]}$. This requires solving Eq. (4.9), where λ , an approximation to the ground state energy, is recorded. The updated value of $A^{[1]}$ is then used in the calculation of H_{eff} and N_{eff} and Eq. (4.9) is solved for $M = 2$. The minimization proceeds to site $N - 1$, then proceeds back from site N to site 2. After the complete sweep, the standard deviation of the energies calculated at each site is compared to a chosen tolerance. If the deviation is not below the tolerance, the wavefunction is determined to have not converged sufficiently and the sweep is executed again. Due to Eq. (4.1) and the minimization used, in the absence of numerical errors, the ground state energy calculated will be a monotonically decreasing function of the number of sweeps.

While computing local minimizations does not guarantee a global minimum, the fact that the ground state decreases monotonically and the technique requires a standard deviation of energy values to be below a given tolerance lead to accurate results in practice. Also, as previously mentioned, one can compute values for increasing values of D to verify convergence to the actual ground state. In particular, the ground state energy of the Heisenberg model has been reliably computed [11]. In Chapter 5 the Ising model is simulated and compared with the analytical solution.

4.1.2 Gauge Freedoms

While the generalized eigenvalue problem, Eq. (4.9), is solvable, there is a straightforward manipulation of the MPS's in the open boundary conditions case which leads to a more efficient solution. If N_{eff} is sufficiently close to the identity tensor, then Eq. (4.9) reduces to a simple eigenvalue equation which can be solved significantly quicker than the generalized version.

The principle insight is the *gauge freedom* that is present in the multiplication of matrices (and tensors) [11]. Consider the matrix A , defined such that

$$A = BC. \quad (4.10)$$

One can then insert the identity matrix, \mathbb{I} , between B and C without changing the value of A ,

$$A = BC = B\mathbb{I}C. \quad (4.11)$$

Futhermore, one can decompose \mathbb{I} into a product of an invertible matrix X and its inverse and redefine B and C ,

$$A = BC = B\mathbb{I}C = BXX^{-1}C = \tilde{B}\tilde{C}, \quad (4.12)$$

where $\tilde{B} = BX$ and $\tilde{C} = X^{-1}C$. In this way, the matrices B and C can be redefined as \tilde{B} and \tilde{C} , respectively, provided they satisfy the constraint in Eq. (4.12).

This gauge freedom can be used to define the tensors, $\{A^{[M]}\}$, used for each spin in the system. The procedure utilizes the singular value decomposition [3, 14], which states that for any tensor A

$$A = USB, \quad (4.13)$$

where U and B are both unitary matrices and S is a diagonal matrix whose entries are the singular values of A . The decomposition is then applied to each site starting from one end of the network. Each pair of tensors is decomposed and redefined as

$$A^{[M-1]}A^{[M]} = A^{[M-1]}(U^{[M]}S^{[M]}B^{[M]}) = \tilde{A}^{[M-1]}\tilde{A}^{[M]}, \quad (4.14)$$

where $\tilde{A}^{[M-1]} = A^{[M-1]}U^{[M]}S^{[M]}$ and $\tilde{A}^{[M]} = B^{[M]}$. Applying this decomposition from site N , right to left, to site 1 leaves a unitary tensor, $B^{[M]}$, at each site. In this way, each update of the H_{eff} and N_{eff} tensors are done with unitary tensors. Similarly applying the decomposition allowing

$$A^{[M]}A^{[M+1]} = (U^{[M]}S^{[M]}B^{[M]})A^{[M+1]} = \tilde{A}^{[M]}\tilde{A}^{[M+1]}, \quad (4.15)$$

where $\tilde{A}^{[M]} = U^{[M]}$ and $\tilde{A}^{[M+1]} = S^{[M]}B^{[M]}A^{[M+1]}$, the update from left to right can be done with only unitary tensors. As a result N_{eff} can be constructed entirely out of unitary tensors. This particular construction of N_{eff} in the case of open boundary conditions lets the N_{eff} become

$$N_{eff} = \delta_{\alpha\alpha'}\delta_{\beta\beta'}\delta_{ii'}, \quad (4.16)$$

where the indices refer to the same indices used in Fig. 4.3. Thus, as shown in Eq. (4.16), N_{eff} can be made to be the identity tensor and Eq. (4.9) reduces to a simple eigenvalue problem. Furthermore, since N_{eff} is the identity it is no longer necessary to store the values during computation.

Unfortunately the situation is not as simple for the case of periodic boundary conditions. Since the PBC spin chain forms a ring, the N_{eff} is not split into two disjoint tensor networks, but rather is still connected as a single tensor network. This removes some freedom from the representation and the sites cannot be made all to be unitary. N_{eff} , therefore, cannot be made to be the identity tensor. This procedure, however, is still applied to N_{eff} using PBCs because it improves the conditioning of the N_{eff} tensor.

4.1.3 First Excited State

To calculate the first excited state energy and wavefunction, it is first necessary to have obtained the ground state wavefunction in MPS form. Once the ground state wavefunction has been found, it can be used to take advantage of a corollary to the variational principle, Eq. (4.1) [15]

$$E_{fe} \leq \langle \psi | H | \psi \rangle, \quad (4.17)$$

given that $\langle \psi | \psi_{gs} \rangle = 0$. According to Eq. (4.17) the same minimization technique used in finding the ground state, subjected to an additional constraint, can be applied to find the first excited state. The particular method used to ensure the orthogonality of the first excited state is to project Eq. (4.9) into a subspace orthogonal to the ground state, solve the eigenvalue problem, then project back into the full Hilbert space. One way of computing the projection matrix would be

$$\mathcal{P} = \mathbb{I} - |\psi_{gs}\rangle\langle\psi_{gs}|. \quad (4.18)$$

The calculation of Eq. (4.9) is then preceded by

$$\tilde{N}_{eff} = \mathcal{P}^T N_{eff} \mathcal{P}, \quad (4.19)$$

$$\tilde{H}_{eff} = \mathcal{P}^T H_{eff} \mathcal{P}. \quad (4.20)$$

Then the found MPS is projected back,

$$|\psi_{fe}\rangle = \mathcal{P}|\tilde{\psi}_{fe}\rangle, \quad (4.21)$$

yielding both the first excited state energy and the first excited state wavefunction in MPS form. This particular choice of projector works in theory, however, in practice the problem remains more stable if a function from a software library is used to calculate a projector that will ensure orthogonality. In this case, the function `orth` from the default MATLAB library was used.

4.2 Imaginary Time Evolution Block Decimation Method

An alternative method to finding the ground state energy and MPS by solving Eq. (4.9) is using the Imaginary Time Evolution Block Decimation method (TEBD) [16]. The imaginary TEBD method is a straightforward modification of the traditional TEBD method. The TEBD method starts with the time-dependent Schrödinger equation

$$\mathcal{H}\psi = i\frac{\partial\psi}{\partial t}, \quad (4.22)$$

where \mathcal{H} is the Hamiltonian and ψ is the wavefunction. If the Hamiltonian is time-independent then the solution that describes the time evolution of the system is

$$\psi = Ae^{-i\mathcal{H}t}, \quad (4.23)$$

where A is the normalization constant. To evolve the system by a time step δt one could apply the operator

$$\hat{T} = e^{-i\mathcal{H}\delta t}, \quad (4.24)$$

to the system⁶. Unfortunately for the types of spin chain models studied with this simulation, this application of Eq. (4.24) will not work because the terms in the Hamiltonian typically involve nearest neighbor spin interactions. A system comprised of these interactions will typically not have a Hamiltonian with terms that commute. One can then decompose the Hamiltonian into two or more pieces that do commute and alternately apply these operators. For instance, if the Hamiltonian is

$$\mathcal{H} = \sum_{i=1}^{N-1} \sigma_i^x \sigma_{i+1}^x, \quad (4.25)$$

it can be split into

$$\mathcal{H}_{odd} = \sum_{i=1, i \text{ odd}}^{N-1} \sigma_i^x \sigma_{i+1}^x, \quad (4.26)$$

$$\mathcal{H}_{even} = \sum_{i=1, i \text{ even}}^{N-1} \sigma_i^x \sigma_{i+1}^x, \quad (4.27)$$

where \mathcal{H}_{odd} and \mathcal{H}_{even} do commute. Since they commute, the Trotter decomposition can be applied [11]. The Trotter decomposition states

$$e^{A+B} = \lim_{n \rightarrow \infty} \left(e^{\frac{A}{n}} e^{\frac{B}{n}} \right)^n, \quad (4.28)$$

or equivalently

$$e^{(A+B)\delta t} = \lim_{\delta t \rightarrow 0} \left(e^{A\delta t} e^{B\delta t} + O(\delta t^2) \right), \quad (4.29)$$

where $O(\delta t^2)$ indicates the order in δt of the error in the approximation. Thus for small time steps δt the time evolution of the system can be approximated by alternating the application of the operators

$$\hat{T}_{odd} = e^{-i\mathcal{H}_{odd}\delta t}, \quad (4.30)$$

$$\hat{T}_{even} = e^{-i\mathcal{H}_{even}\delta t}. \quad (4.31)$$

Once \hat{T}_{odd} and \hat{T}_{even} have been determined, the variational method is applied again, this time minimizing

⁶Assuming the wavefunction is already normalized Eq. (4.24) will preserve normalization.

$$\|\hat{T}|\psi(k)\rangle - |\psi(k+1)\rangle\|^2, \quad (4.32)$$

where $|\psi(k)\rangle$ is the initial wavefunction, $|\psi(k+1)\rangle$ is the wavefunction advanced by a timestep to be found, \hat{T} is the relevant even or odd operator, and $\|\cdot\|$ is the Euclidean norm. The result of the minimization is the same as maximizing the overlap between $|\psi(k+1)\rangle$ and $\hat{T}|\psi(k)\rangle$. The minimization then continues in exactly the same fashion as the VMPS method. This method simulates the time evolution of the system.

The modification to the TEBD method that will allow the ground state wavefunction to be found is letting $i \rightarrow 1$ in Eq. (4.23) and subsequently in Eqs. (4.30) and (4.31). The resulting equation is

$$\psi = Ae^{-\gamma t}, \quad (4.33)$$

where the higher energy states will decay away quicker, leaving the ground state energy and wavefunction. In practice, when implemented, this method performed as well as the VMPS method in terms of accuracy. In terms of convergence time, the implementation used was significantly slower than the VMPS method⁷. As a result, the method is a very useful sanity check in comparing against results from the VMPS method, however, due to the convergence speed, the VMPS method was used as the method of choice in this thesis. The imaginary TEBD is also a useful method if only a rough estimate of the energy or wavefunction is needed. In this case, the method can be run for a fixed number of time steps to find an estimate, rather than needlessly perform the entire algorithm.

⁷In [16] Vidal has implemented the TEBD method differently which is likely to be as efficient as the VMPS method.

Chapter 5

Study of Ising Model in a Transverse Field

5.1 Theory

The Ising model in a transverse field belongs to a family of one dimensional models known as XY spin chains [17, 18]. The Ising model is interesting to study because despite its apparent simplicity it still exhibits interesting behaviour. The model itself can be used as a simplified description of ferromagnetism at the level of individual spins. The general XY spin chains are described by the Hamiltonian

$$\mathcal{H}_{XY} = - \sum_{j=1}^N \left(\frac{1+r}{2} \sigma_j^x \sigma_{j+1}^x + \frac{1-r}{2} \sigma_j^y \sigma_{j+1}^y + h \sigma_j^z \right), \quad (5.1)$$

where r describes the xy anisotropy of the system and h is the transverse external field [18]. In the case $r = 1$, Eq. (5.1) reduces to

$$\mathcal{H} = - \sum_{j=1}^N (\sigma_j^x \sigma_{j+1}^x + h \sigma_j^z), \quad (5.2)$$

which is precisely the Ising model in a transverse field. The $\sigma_j^x \sigma_{j+1}^x$ term describes the spin-spin interaction between nearest neighbours in the system while the $h \sigma_j^z$ describes the transverse external field strength.

When $h = 0$, intuitively, the ground state maximizes the $\sigma_j^x \sigma_{j+1}^x$ terms resulting in a spin chain with all spins aligned in the x direction. The spins can align in the $+x$ direction, $-x$ direction, or a linear superposition of the two

$$|\psi_{gs}^{h=0}\rangle = \alpha |++\dots+\rangle + \beta |--\dots-\rangle, \quad (5.3)$$

for $|\alpha|^2 + |\beta|^2 = 1$. The other limiting case is $|h| \gg 1$ where the $h \sigma_j^z$ term dominates. The ground state for this value of h maximizes the spin the $+z$ direction (assuming $h > 0$)

$$|\psi_{gs}^{h \gg 1}\rangle = |\uparrow\uparrow \dots \uparrow\rangle. \quad (5.4)$$

The region of interest is between the two limiting cases where $h \neq 0$ but is not too large. There exists an analytical solution to the general XY model, see [17, 18], which is

$$E_a = - \sum_{m=0}^{N-1} \sqrt{\left(h - \cos \frac{2\pi(m+1/2)}{N} \right)^2 + r^2 \sin^2 \frac{2\pi(m+1/2)}{N}}, \quad (5.5)$$

$$E_b = (h - 1) - \sum_{m=1}^{N-1} \sqrt{\left(h - \cos \frac{2\pi m}{N}\right)^2 + r^2 \sin^2 \frac{2\pi m}{N}}. \quad (5.6)$$

where the ground state $E_{gs} = \min(E_a, E_b)$ and the first excited state $E_{fs} = \max(E_a, E_b)$. For the Ising model in a transverse field this corresponds to $E_{gs} = E_a$ and $E_{fs} = E_b$. Furthermore, in this case one can reduce Eqs. (5.5) and (5.6) to

$$E_{gs} = - \sum_{m=0}^{N-1} \sqrt{\left(h - \cos \frac{2\pi(m+1/2)}{N}\right)^2 + \sin^2 \frac{2\pi(m+1/2)}{N}}, \quad (5.7)$$

$$E_{fs} = (h - 1) - \sum_{m=1}^{N-1} \sqrt{\left(h - \cos \frac{2\pi m}{N}\right)^2 + \sin^2 \frac{2\pi m}{N}}. \quad (5.8)$$

5.2 Results

The simulation used was based on a number of functions written by Verstraete et al. [11]. The original code was written for open boundary conditions and simply generated the ground state and first excited state energies and MPS's for a given Hamiltonian. Significant modifications were required to extend the code to utilize periodic boundary conditions and compute magnetizations and correlations and store the computations for plotting. The code is included in Appendix B.

Several datasets were generated for given values of N , the number of spins, and D , the bond length. For each dataset the field strength, h , was varied from 0 to 2 using different sampling rates.

N	D	h sampling rate
50	8	≈ 0.04
60	8	≈ 0.1
70	8	≈ 0.2
80	8	≈ 0.02
100	10	≈ 0.2

Table 5.1: The values of N , D , and h sampling rate for which datasets were generated.

The values for the datasets used are shown in Table 5.1. Due to the large amount of time required to generate each dataset only a limited number were produced. In the simulation, a tolerance was set to specify when to halt the minimization iteration. In all cases, the tolerance was set to 10^{-5} .

5.2.1 Energy

In order to evaluate the accuracy and validity of the simulation the simulated ground state energy and first excited state energy were compared to the exact values as shown in Eqs. (5.7) and (5.8).

Fig. 5.1 compares the simulated ground state energy with the exact solution. The results agree closely enough that any error at all is not evident from the plot. Fig. 5.2 similarly plots the first excited state. Again the simulation and exact solution are close enough that they appear

superimposed on the plot. In order to quantify the error Fig. 5.4 plots absolute difference between the exact solution and simulated value. As the tolerance was set to 10^{-5} , the agreement is very good since the error remains at or below 10^{-5} for all values of h . There are two particularly interesting features in Fig. 5.4. The first is that the error in the ground state is noticeably less than the error in the first excited state. This behaviour is expected because the ground state MPS is used in the calculation of the first excited state energy and MPS. Therefore any error present in the ground state MPS propagates to the first excited state. The second interesting feature is that for both the energy levels the error tends to be higher at $h \approx 1$. This is best explained with reference to Fig. 5.3 which shows the gap in the energy levels. For $h < 1$, the energy gap is 0, however, for $h > 1$ the energy gap increases linearly. Thus $h_c = 1$ can be identified as a critical point of the system. Since $h_c = 1$ is a critical point, it is reasonable to expect the error in the energy to be larger for values of h close to h_c , which is what is observed.

Another useful way to interpret the energy level results is comparing with the limiting values of Eqs. (5.7) and (5.8). For $h = 0$ the equations reduce to

$$E_{gs}^{h=0} = - \sum_{m=0}^{N-1} \sqrt{\left(-\cos \frac{2\pi(m+1/2)}{N}\right)^2 + \sin^2 \frac{2\pi(m+1/2)}{N}} = - \sum_{m=0}^{N-1} 1 = -N, \quad (5.9)$$

$$E_{fs}^{h=0} = -1 - \sum_{m=1}^{N-1} \sqrt{\left(-\cos \frac{2\pi m}{N}\right)^2 + \sin^2 \frac{2\pi m}{N}} = -1 - \sum_{m=1}^{N-1} 1 = -N. \quad (5.10)$$

Using Eqs. (5.9) and (5.10) as approximations for small h , the energy gap is expected to be 0. This is precisely confirmed in Fig. 5.3. For large h , the sum terms in Eqs. (5.7) and (5.8) are approximately equal since h will dominate. Note that as N always becomes large the extra term in Eq. (5.7) becomes negligible. Furthermore, for $h \approx 1$, a $m = 0$ term in the sum of Eq. (5.8) would have no contribution to the energy. In light of this approximation, the gap is expected to be $h - 1$ which is precisely the behaviour for $h > 1$.

Figs. 5.5 and 5.6 show the energy gap and error in energy levels, respectively, for a larger number of spins, $N = 80$, and more finely spaced field values. Again, the results agree with the expected values. The first state excited energies in Fig. 5.5 tend to oscillate more than in the $N = 60$ case. This is likely due to the various numerical approximations made in the method. The fact that the error does not significantly go above 10^{-5} , however, implies that the simulation is stable.

5.2.2 Magnetization

The magnetization of a system in the x direction is given by

$$M_x = \frac{1}{N} \sum_i \langle \sigma_i^x \rangle. \quad (5.11)$$

The equations for the magnetization in the y and z directions are defined similarly. The magnetization of the system in the z direction describes how well the spins align with the external field, while the magnetization in the x and y directions describes the spontaneous alignment among the spins. Some materials retain their magnetization even when removed from the external field and placed in any other external field. In this case the material is known as a *ferromagnetic*

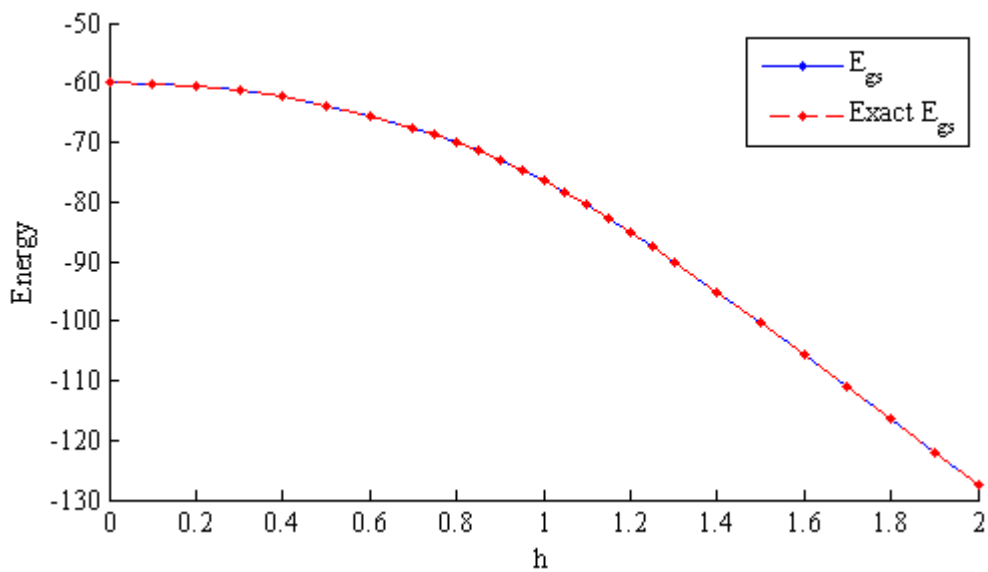


Figure 5.1: The simulated ground state energy and exact ground state energy for $N = 60$ and $D = 8$.

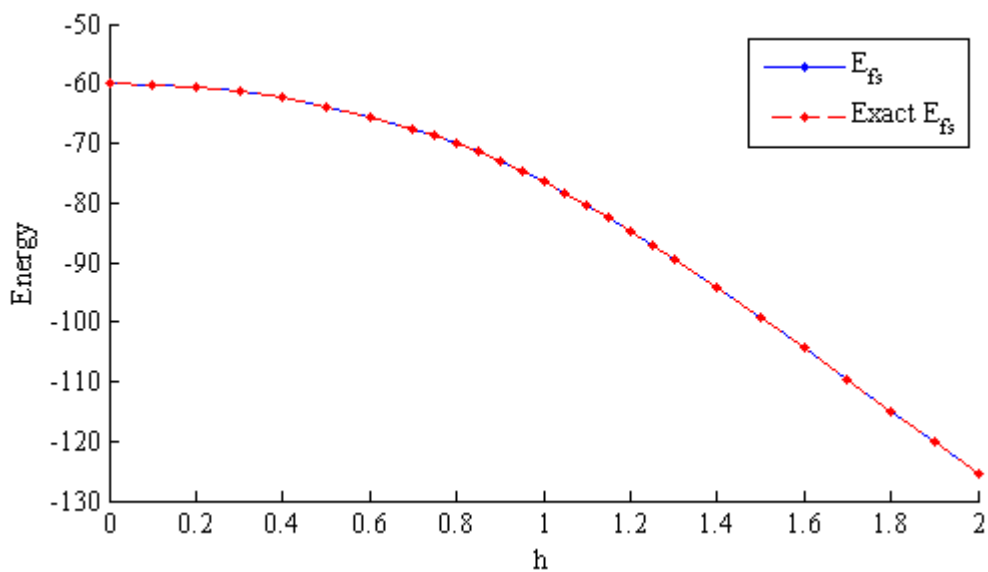


Figure 5.2: The simulated first excited state energy and exact first excited state energy for $N = 60$ and $D = 8$.

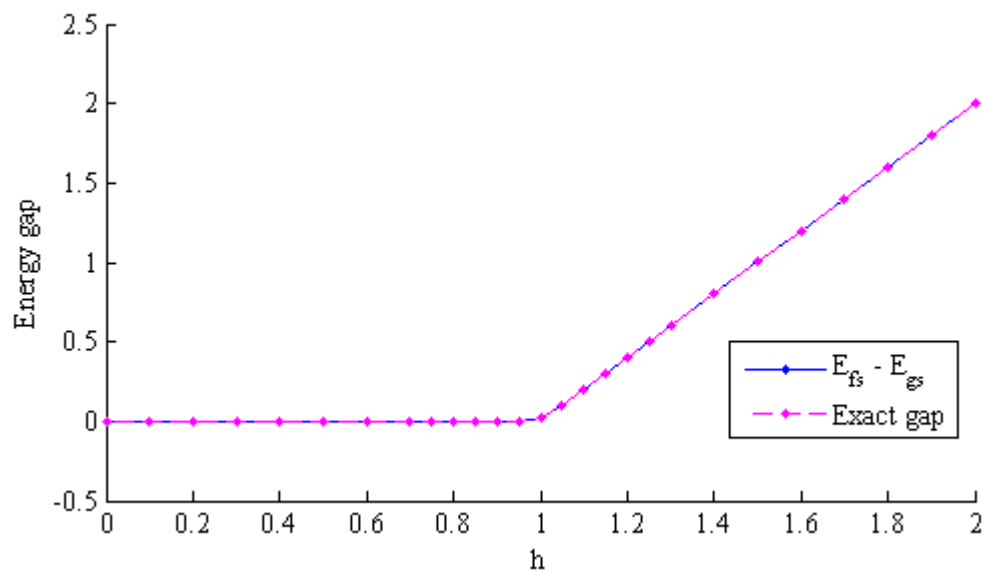


Figure 5.3: The simulated energy gap, $E_{fs} - E_{gs}$, and exact energy gap for $N = 60$ and $D = 8$.

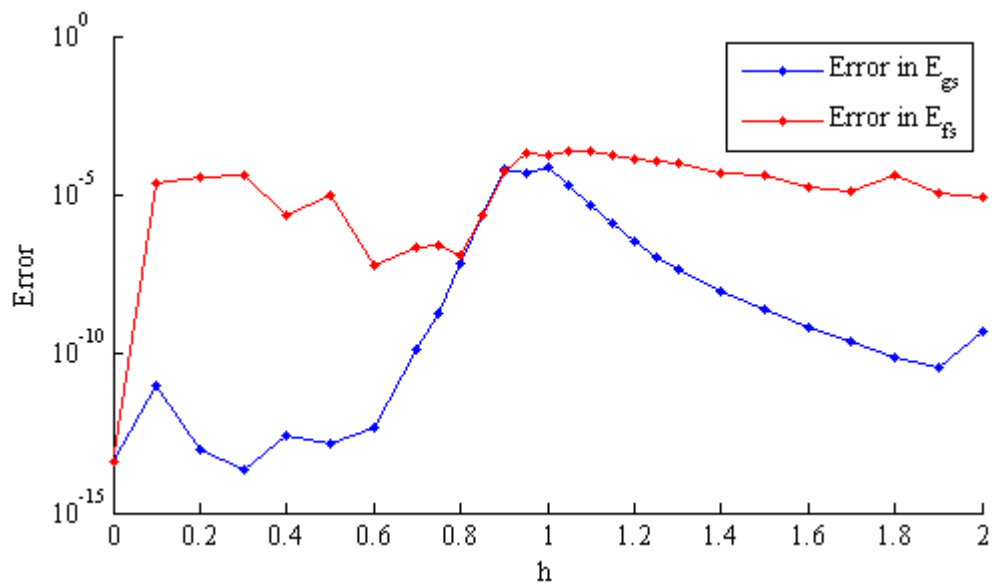


Figure 5.4: The absolute error in the ground state energy and the first excited state energy for $N = 60$ and $D = 8$.

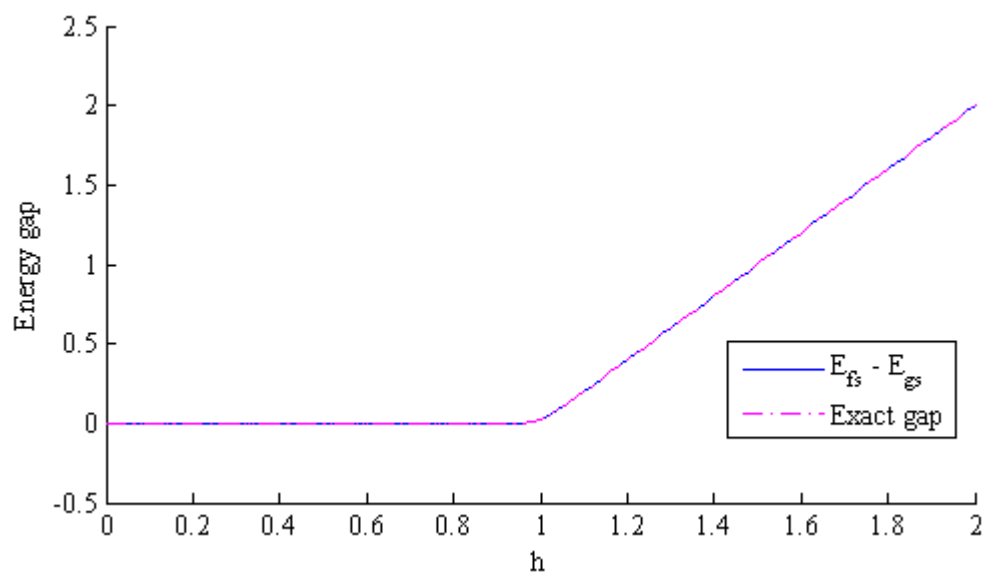


Figure 5.5: The simulated energy gap, $E_{fs} - E_{gs}$, and exact energy gap for $N = 80$ and $D = 8$.

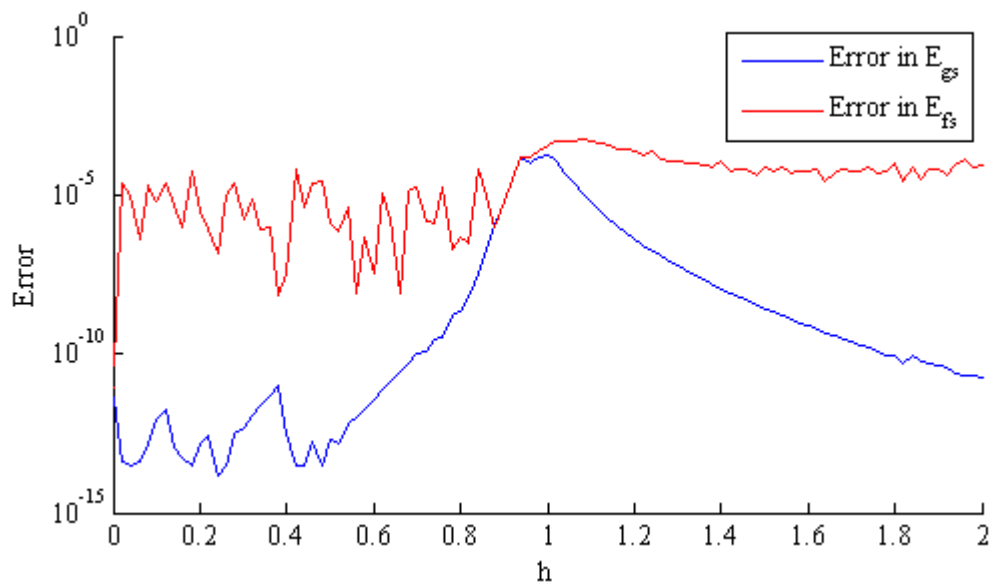


Figure 5.6: The absolute error in the ground state energy and the first excited state energy for $N = 80$ and $D = 8$.

material and the magnetization reveals how the material will magnetically interact with other materials.

With the one dimensional Ising model studied here we simply wish to describe the magnetization in an external field, h , and its behaviour close to the critical point, h_c . Fig. 5.7 plots M_x , M_y , and M_z as a function of field strength, h . The magnetization in the y direction remains around 0 which is expected given that there is no y direction spin interaction in Eq. (5.2). The magnetization in the z direction starts at 0 for $h = 0$. This is also expected since when $h = 0$, the z direction spin interaction vanishes. As h increases, the z direction spin interaction begins to dominate which is evident in the monotonic increase of M_z in Fig. 5.7.

The magnetization in the x direction offers a slightly richer analysis. In Eq. (5.2) at $h = 0$ the x direction spin interaction is the only term, thus M_x reaches its maximum value there. As h increases, M_x decreases. Unlike M_z , however, the behaviour of M_x abruptly changes near the critical point. For $h < h_c$, the x magnetization is slowly decreasing, but at $h \approx h_c$ the value quickly approaches 0 and remains there for $h > h_c$. In this case, M_x is identified as an order parameter of the system.

In the two dimensional Ising model, the magnetization is known, using mean-field theory, to vary $\propto (T_c - T)^{1/8}$ for $T < T_c$ where T is temperature [19]. The magnetization then goes to 0 for $T > T_c$. For finite size systems it is reasonable to expect the curve to not go directly to 0, but rather continuously decrease to 0 rapidly as shown in Fig. 5.7. Similarly for field strength h , in the two dimensional model the magnetization is $\propto (h_c - h)^{1/8}$ [19]. It is interesting to ask if it also varies as $\propto (h_c - h)^a$ for some exponent a also in the one dimensional model.

For $0 < h < h_c$ the magnetization curve was fit to $\propto (h_c - h)^a$ and it was found that $a = 0.122 \pm 0.006$ and $h_c = 0.9999 \pm 0.0002$. This result suggests that in this system $M_x \propto (h_c - h)^{1/8}$ with a critical point at $h_c \approx 1$.

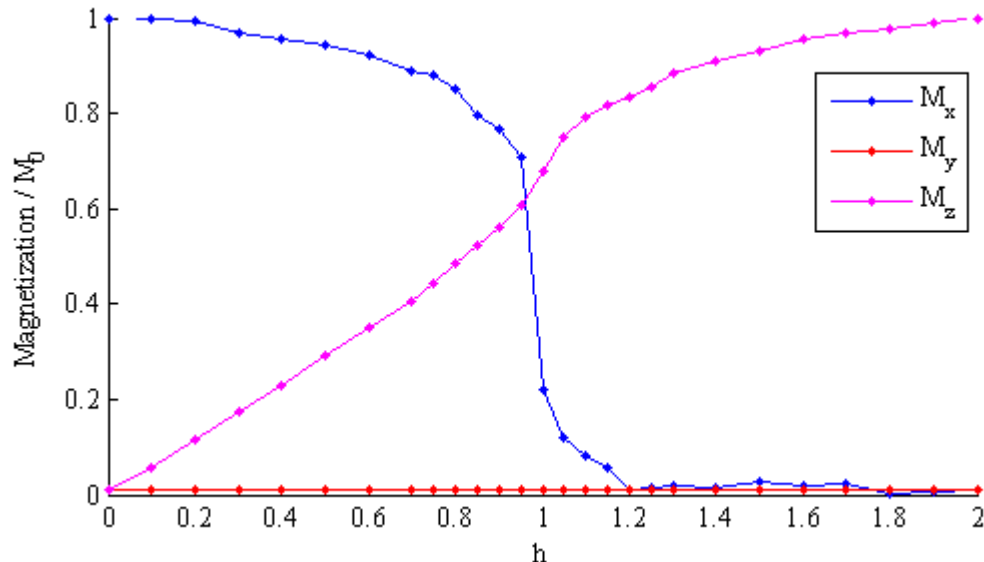


Figure 5.7: The magnetization in the ground state for $N = 100$ and $D = 10$.

5.2.3 Correlation

The correlation in a system describes how correlated spins in a system are in a given direction as a function of distance along the spin chain. The equation used to describe this is

$$c_{xx}(r) = \frac{1}{N} \sum_i \langle \sigma_i^x \sigma_{i+r}^x \rangle, \quad (5.12)$$

where $r \in [1, N]$. The equations for $c_{yy}(r)$ and $c_{zz}(r)$ are similarly defined. Fig. 5.8 shows the correlations as a function of h for fixed $r = \frac{N}{2}$. The results of Fig. 5.8 agree intuitively with the expected results. In the y direction, there is not expected to be any correlation, for the same reasons that the magnetization in the y direction is expected to be 0. In the z direction for $h = 0$, the correlation is expected to be 0 since there is no z direction spin interaction in Eq. (5.2). For $h > 0$, c_{zz} grows as expected. Similarly, c_{xx} starts at a maximum for $h = 0$, then decreases to 0 as its contribution to Eq. (5.2) varies with h .

The plots in Fig. 5.8 also resemble the magnetization plots in Fig. 5.7. This behaviour is expected for large system. Note that as $N \rightarrow \infty$, $\langle \sigma_i^x \sigma_{i+r}^x \rangle \rightarrow \langle \sigma_i^x \rangle \langle \sigma_{i+r}^x \rangle$ since the spins should be uncorrelated in an infinite size system. Furthermore as $N \rightarrow \infty$, $\langle \sigma_i^x \rangle \langle \sigma_{i+r}^x \rangle \rightarrow \langle \sigma_i^x \rangle^2 = \langle M_x \rangle^2$.

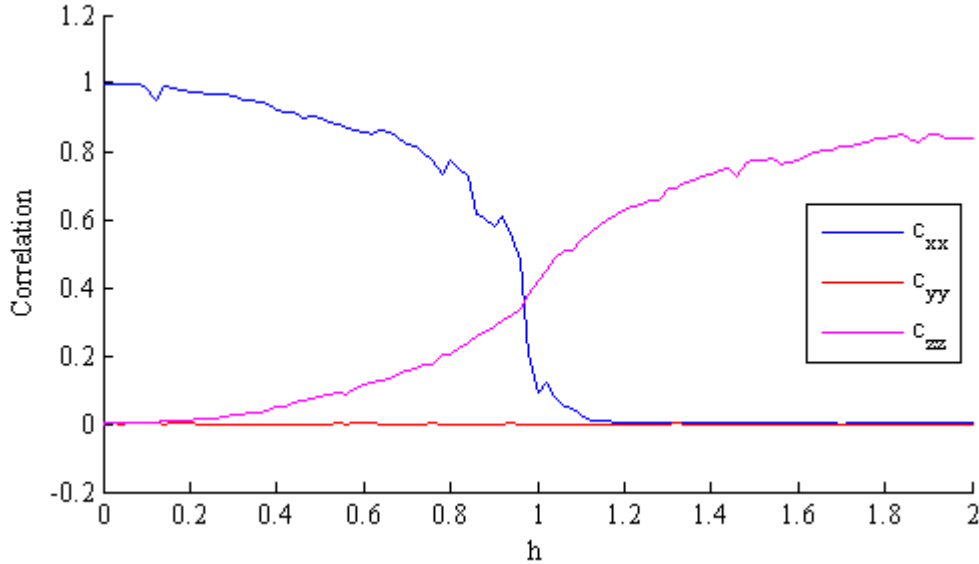


Figure 5.8: The correlations, according to Eq. (5.12), as a function of h for $r = \frac{N}{2}$ in the ground state for $N = 80$ and $D = 8$.

An alternate definition of the correlation is given by

$$\tilde{c}_{xx}(r) = \frac{1}{N} \sum_i (\langle \sigma_i^x \sigma_{i+r}^x \rangle - \langle \sigma_i^x \rangle \langle \sigma_{i+r}^x \rangle), \quad (5.13)$$

where $r \in [1, N]$ and the equations for $\tilde{c}_{yy}(r)$ and $\tilde{c}_{zz}(r)$ are similarly defined. This definition of the correlation is expected to decrease exponentially as a function of r [19].

Fig. 5.9 shows the x correlation for various values of r . The correlation peaks at the critical point, h_c , and decreases as h increases. The plots for larger values of r also have less correlation

5.2. Results

since as the distance in the spin chain increases, the correlation should decrease between spins. Fig. 5.10 clearly shows the exponential decay of the x correlation. Moreover, the plot shows that for larger values of h the correlation in the x direction decreases. The values of \tilde{c}_{yy} and \tilde{c}_{zz} similarly decrease in the same exponential fashion. As the correlation falls exponentially, it can be fit to an exponential function

$$\tilde{c}_{xx}(r) = Ae^{-\frac{r}{\xi}}, \quad (5.14)$$

where ξ has the same units as r and is known as the *correlation length*. The correlation length is simply a measure of how far the information of spin direction travels along the spin chain. Since there is a correlation length for each value of h and in each direction, the correlation length can be computed and plotted as a function of h . This is shown in Fig. 5.11 for $N = 60$ and in Fig. 5.12 in $N = 80$ which has more data points. It is noteworthy that in both plots of the correlation length, the value peaks at $h \approx h_c$, and that as N increases, the height of the peak increases⁸. This suggests that as the system approaches its critical point the correlation between components of the system becomes very large. Physically this means that near the critical point the behaviour of the system can change very rapidly since the information can effectively influence a larger number of spins. As h increases past h_c , the correlation length decreases, as in general the correlation is not expected to propagate far along the chain.

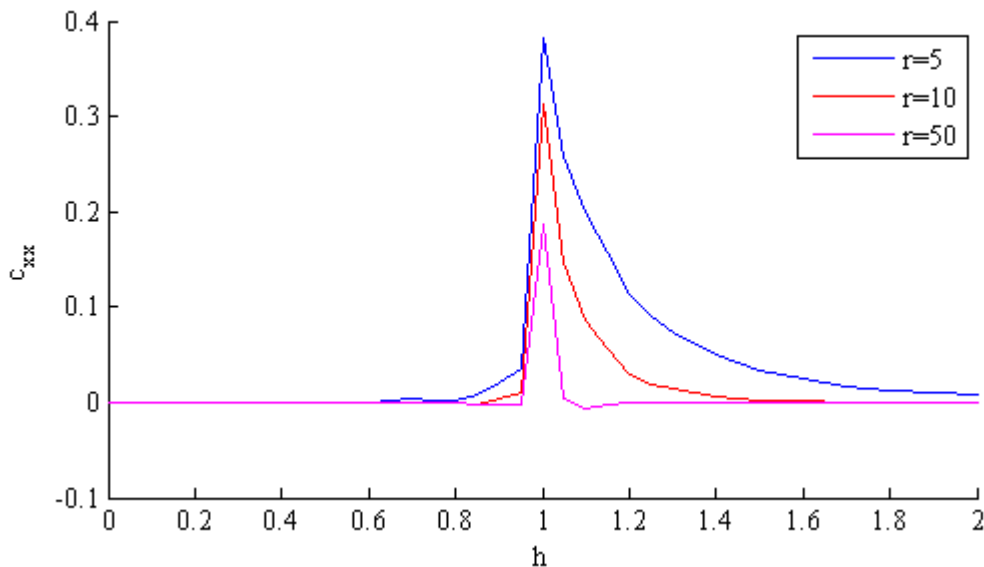


Figure 5.9: The x correlation, according to Eq. (5.13), as a function of h for $r = 5$, $r = 10$, and $r = 50$ in the ground state for $N = 100$ and $D = 10$.

⁸Although only two datasets are shown, the other datasets were plotted and found to agree with the assessment that as N increases, the height of the peak increases.

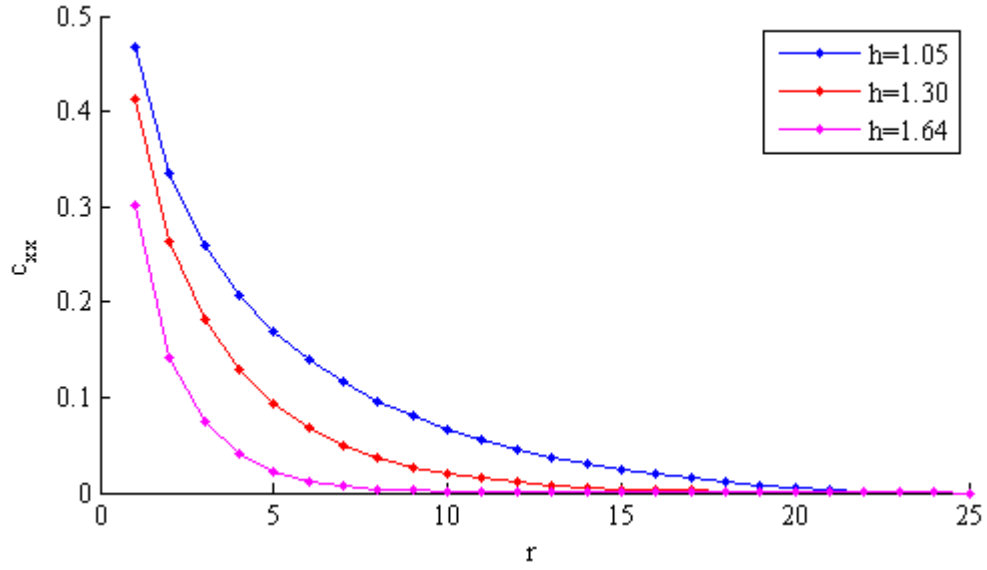


Figure 5.10: The x correlation, according to Eq. (5.13), as a function of r for $h = 1.05$, $h = 1.30$, and $h = 1.64$ in the ground state for $N = 50$ and $D = 8$.

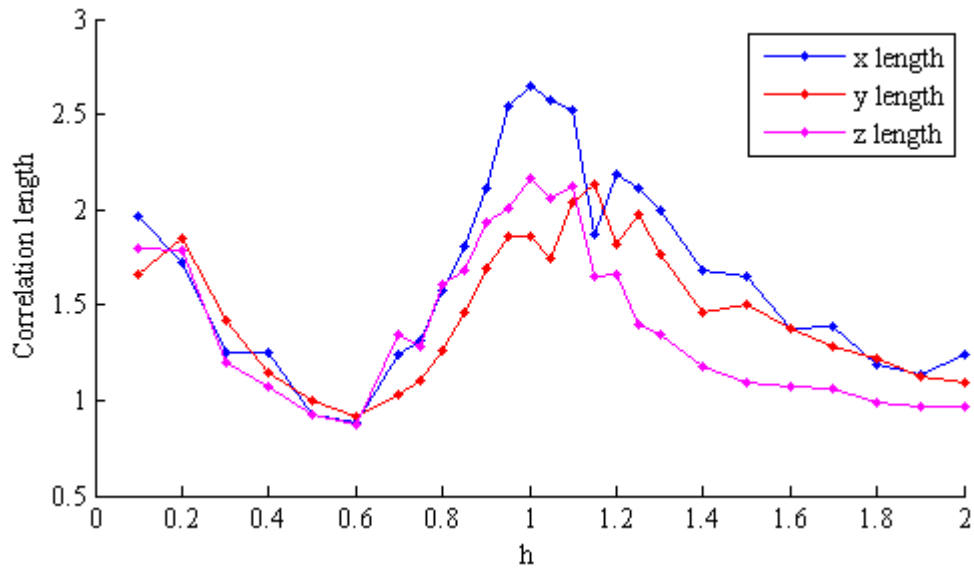


Figure 5.11: The correlation length, ξ , as a function of h in the ground state for $N = 60$ and $D = 8$.

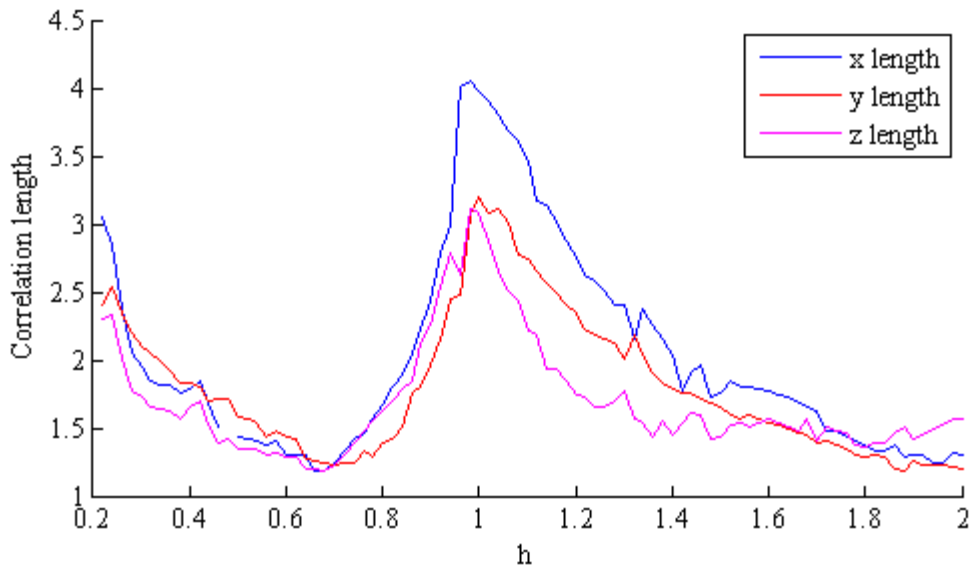


Figure 5.12: The correlation length, ξ , as a function of h in the ground state for $N = 80$ and $D = 8$.

5.3 Numerical Issues

Unlike with other models, an efficient, accurate simulation of the Ising model requires special attention to the conditioning of the N_{eff} tensor when $h = 0$ ⁹. The reason for this is manifest in Eq. (5.3), where the ground state is degenerate. This degeneracy causes some of the eigenvalues of N_{eff} to be very small, which greatly reduces the stability of solving Eq. (4.9). To achieve a stable algorithm it was necessary to ensure that N_{eff} had sufficiently large eigenvalues, or alternatively stated, was well conditioned.

5.3.1 Wrap Around

In the open boundary conditions case the minimization sweep went from site 1 to $N - 1$, then from N to 2, varying each site and performing a local minimization. Using this technique it was possible to make $N_{eff} = \mathbb{I}$. In the periodic boundary conditions case, the minimization sweep was still implemented to iterate from 1 to $N - 1$, then from N to 2, despite the fact that N_{eff} can no longer be made to be the identity. It was observed in running the algorithm that while the minimization converged for sites away from the ends of the sweep, near the ends of the sweep, namely sites 1 and N the minimization was not converging and N_{eff} became poorly conditioned.

The proposed remedy to reduce the end effects was to complete an additional singular value decomposition when sweeping from site 1 to $N - 1$ at site N and at site 1 when sweeping from N to 2. The non-unitary tensor from site N will then be contracted with site 1. This effectively *wraps around* the information in the N_{eff} and H_{eff} more consistently. In implementing this change, however, it is also necessary to recalculate the H_{eff} and N_{eff} tensors from site 1 to N before continuing the sweep from site N to 2. This is necessary because the wrap around of the

⁹The Heisenberg model, for instance, can be simulated without the modifications presented in this section [11].

tensor causes the information to be updated in site 1 which is included in all minimizations in the sweep. The same wrap around technique is applied to site 1.

This technique when implemented with variable bond length, discussed in Section 5.3.2, did in fact increase the stability of N_{eff} . There were, however, still some instances where N_{eff} would have small eigenvalues and the algorithm would not converge.

5.3.2 Variable Bond Length

Another source of poor conditioning of the N_{eff} tensor was when the bond length, D , was set too large. In theory, this should present no issue because as $D \rightarrow \chi$ the MPS representation becomes an exact representation of the wavefunction. It was observed, however, that as D was set unnecessarily large, meaning that a small D sufficiently represents the system, the N_{eff} tensor becomes sparse. As N_{eff} becomes more sparse it also becomes more poorly conditioned.

The solution to this issue was to allow the bond length to decrease during computation as necessary. Specifically when each tensor is decomposed using the singular value decomposition if a singular value of the tensor is below a specified tolerance, then the singular value is estimated to be 0 and the bond length is effectively decreases. This amounts to a change in the `svd2.m` program in Appendix B as shown in Program 5.1.

This change used with the wrap around modification did increase the stability of N_{eff} in many instances. Near $h = 0$, the bond length also tended to decrease to $D \approx 2$ (depending on the size of the system). This is because for $h = 0$ there is a well defined ground state, Eq. (5.3), and a longer bond length was not necessary to fully represent the system.

5.3.3 Subspace Projection

Upon discussion with Valentin Murg, a colleague of Verstraete, he suggested an alternative method to improve the conditioning of N_{eff} with a method similar to that presented in Section 4.1.3. The concept is to find the eigenvalues of N_{eff} , then identify which eigenvalues are below a specified tolerance. Then using the eigenvectors of the corresponding eigenvalues less than the tolerance, N_{eff} is projected into the subspace orthogonal to these eigenvectors. This ensures that in the projected subspace N_{eff} will not be poorly conditioned and if the tolerance is small, then the accuracy will also be maintained. The transformation of N_{eff} and H_{eff} are simply

$$\tilde{N}_{eff} = \mathcal{P}_2^T N_{eff} \mathcal{P}_2, \quad (5.15)$$

$$\tilde{H}_{eff} = \mathcal{P}_2^T H_{eff} \mathcal{P}_2, \quad (5.16)$$

where \mathcal{P}_2 is the projector into the subspace orthogonal to the eigenvalues less than the tolerance. Then the found MPS is projected back,

$$|\psi\rangle = \mathcal{P}_2 |\tilde{\psi}\rangle. \quad (5.17)$$

This method was found to consistently make N_{eff} well conditioned without sacrificing any accuracy in the computation. Furthermore, this technique is not as computationally demanding as Verstraete's transformation and it is conceptually more transparent. As a result, this was the method used in code in Appendix B and plots generated in Chapter 5.

Program 5.1 Program that allows for variable bond lengths when decomposing each tensor.

```
function [U,S,V]=svd2_variable(T)

tol=1e-2;
[m,n]=size(T);

% 1->N Case
if m>=n
    [U,S,V]=svd(T,0);
    singVals=diag(S);
    newLength = length(singVals) - 1;
    if (min(singVals) < tol && newLength >= 1)
        fprintf('Changing bond length to %i.\n' ,newLength);
        S=S(1:newLength,1:newLength);
        V=V(1:newLength,1:newLength);
        U=U(:,1:newLength);
    end

% N->1 Case
else
    [V,S,U]=svd(T',0);
    singVals=diag(S);
    newLength = length(singVals) - 1;
    if (min(singVals) < tol && newLength >= 1)
        fprintf(' Changing bond length to %i.\n' ,newLength);
        S=S(1:newLength,1:newLength);
        U=U(1:newLength,1:newLength);
        V=V(:,1:newLength);
    end
end

V=V';
N=sqrt(trace(S*S));
S=S/N;
```

Chapter 6

Conclusion

The objective of this thesis was to study and understand numerical techniques used in simulating many body quantum systems. In particular, methods useful for analyzing one dimensional spin chains were studied. In order to simulate quantum systems efficiently and accurately it was necessary to make the appropriate approximations after parameterizing the wavefunction as a matrix product state. The approximation made was limiting the extent of entanglement in the system via bounding the Schmidt number, χ , of the system by a chosen bond length D . This modification changed the time complexity of the simulation from exponential to polynomial, namely Nd^2D^3 for open boundary conditions and Nd^2D^5 for periodic boundary conditions, where N is the number of spins in the system, D is the bond length, and d is the dimension of the spins. As an extension, since limiting the entanglement allowed for an efficient and accurate simulation of some one dimensional spin chains it was determined that entanglement is a necessary resource for genuine quantum computation.

The primary method developed was the variational MPS method which used a minimization equation to find the ground state MPS and energy of a system. The method was derived from the variational principle and used a sweeping technique to iterate from site to site in the tensor network. Once the energy converged, the simulation was deemed to have found the ground state energy. An additional constraint that a found wavefunction needed to be orthogonal to an already determined ground state MPS led to finding the first excited state energy and MPS using the same VMPS method.

These methods were applied to the one dimensional Ising model for which there is an analytical solution of the energy levels as a function of the field strength, h . The results of the simulation agreed within the prespecified tolerance to the exact solution for both the ground state and first excited state energies. Furthermore, in analyzing the computed ground state MPS, the x magnetization was identified as an order parameter of the system with a critical point at $h_c \approx 1$. Using this MPS, the magnetization in the y and z directions were also computed as well as the correlations as a function of h and r , the distance along the spin chain. The x magnetization and correlation were found to decrease as a function of h while the corresponding z quantities were found to increase. The y values remained at 0 in each case. The correlations were also examined as a function of r and found to decrease exponentially, defining a correlation length as the coefficient of the decay. The correlation length was a measure of the distance that spin information travels around the chain. Moreover, it was found that at the critical point, h_c , the correlation length peaked before decaying for all larger values of h .

In working with the Ising model, a number of numerical instabilities were discovered in the algorithms and consequently solved by projecting the equations in a subspace with sufficiently large eigenvalues. This ensured the well conditioning of the tensors so that they equation could be solved accurately.

Through this study, not only was an involved technical understanding of the simulation of quantum systems gained, but also a deeper understanding of quantum systems themselves was

gained. In particular, the behaviour of the Ising model was studied in detail and found to exhibit very non-trivial behaviour despite its seeming simplicity. The techniques developed and the results found in this thesis are of use to further advances in the numerical methods utilizing matrix product states and to deeper studies of the intricacies of the one dimensional Ising model in a transverse field.

Bibliography

- [1] G. Vidal, Phys. Rev. Lett. **93**, 040502 (2004).
- [2] R. Feynman, *Feynman Lectures on Computation*. Westview Press (2000).
- [3] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge (2000).
- [4] G. Vidal, Phys. Rev. Lett. **91**, 147902 (2003).
- [5] S. R. White, Phys. Rev. Lett. **69**, 2863 (1992).
- [6] I. Affleck, T. Kennedy, E. H. Lieb, and H. Tasaki, Phys. Rev. Lett. **59**, 799 (1987); Commun. Math. Phys. **155**, 477 (1988).
- [7] M. Fannes, B. Nachtergaele, and R. F. Werner, Commun. Math. Phys. **144**, 443 (1992); J. Funct. Anal. **120**, 511 (1994).
- [8] S. Östlund and S. Rommer, Phys. Rev. Lett. **75**, 3537 (1995).
- [9] F. Verstraete, D. Porras, J. I. Cirac, Phys. Rev. Lett. **93**, 227205 (2004).
- [10] D. Perez-Garcia, F. Verstraete, M.M. Wolf, and J.I. Cirac, [arXiv:quant-ph/0608197](https://arxiv.org/abs/quant-ph/0608197) (2007).
- [11] F. Verstraete, V. Murg, and K. Cirac, Advances in Physics, 57:2,143-224 (2008).
- [12] D. Porras, F. Verstraete, and J. I. Cirac, Phys. Rev. B **73**, 014410 (2006).
- [13] M. D. Sturge, *Statistical and Thermal Physics*. A K Peters (2003).
- [14] A. Ekert and P. L. Knight, Am. J. Phys. **63**, 415 (1995).
- [15] D. Griffiths, *Introduction to Quantum Mechanics*. Pearson Education (2005).
- [16] G. Vidal, Phys. Rev. Lett. **98**, 070201 (2007).
- [17] E. Lieb, T. Schultz, and D. Mattis, Ann. Phys. **60**, 407 (1961).
- [18] T. Wei, *Quantum Entanglement: Geometric Quantification and Applications to Multi-Partite States and Quantum Phase Transitions*, [arXiv:0905.2467v1](https://arxiv.org/abs/0905.2467v1) (2004).
- [19] H. Gould and J. Tobochnik, *Thermal and Statistical Physics*. Princeton University Press (2010).

Appendix A

Mathematics

A.1 Schmidt Decomposition

The Schmidt decomposition is a way of decomposing a quantum system $|\psi\rangle$ into a tensor product of two quantum systems $|i_A\rangle$ and $|i_B\rangle$ with coefficients λ_i . The formulation and proof of the decomposition is courtesy of Nielsen and Chuang [3].

Theorem:

For any (pure state) $|\psi\rangle$ which consists of two composite systems, A and B , there exist orthonormal states for A , $|i_A\rangle$, and for B , $|i_B\rangle$ such that

$$|\psi\rangle = \sum_i \lambda_i |i_A\rangle |i_B\rangle, \quad (\text{A.1})$$

where λ_i are non-negative real numbers, known as Schmidt coefficients, satisfying $\sum_i \lambda_i^2 = 1$.

Proof:

Let the composite systems, A and B , be of the same dimension for simplicity and have fixed orthonormal bases, $|j\rangle$ and $|k\rangle$, respectively. The state $|\psi\rangle$ can be represented as

$$|\psi\rangle = \sum_{jk} a_{jk} |j\rangle |k\rangle, \quad (\text{A.2})$$

where a is a matrix of complex numbers $\{a_{jk}\}$. The matrix can then be decomposed using the singular value decomposition which states that any matrix, a , can be decomposed as $a = usv$, where u and v are unitary matrices and s is a diagonal matrix whose entries are non-negative and the singular values of a . Thus

$$|\psi\rangle = \sum_{ijk} u_{ji} s_{ii} v_{ik} |j\rangle |k\rangle. \quad (\text{A.3})$$

Let $|i_A\rangle = \sum_j u_{ji} |j\rangle$, $|i_B\rangle = \sum_k v_{ik} |k\rangle$, and $\lambda_i = s_{ii}$, then

$$|\psi\rangle = \sum_i \lambda_i |i_A\rangle |i_B\rangle. \quad (\text{A.4})$$

Since u is unitary and the basis $|j\rangle$ is orthonormal the basis $|i_A\rangle$ is orthonormal. Similarly the basis $|i_B\rangle$ is orthonormal. \square

A.2 Variational Principle

The variational principle upon which the VMPS method is based states

$$E_{gs} \leq \langle \psi | H | \psi \rangle, \quad (\text{A.5})$$

for any normalized wavefunction $|\psi\rangle$. The proof of the principle is straightforward and based upon Griffith's proof [15].

Proof:

Let $|\psi\rangle$ be any normalized wavefunction. Since the eigenfunctions of the Hamiltonian, H , form a complete orthonormal set $|\psi\rangle$ can be expressed as a linear combination of them

$$|\psi\rangle = \sum_n c_n |\psi_n\rangle, \quad (\text{A.6})$$

where $H\psi_n = E_n\psi_n$. From the normalization of $|\psi\rangle$

$$1 = \langle \psi | \psi \rangle = \left(\sum_m c_m^* \langle \psi_m | \right) \left(\sum_n c_n |\psi_n\rangle \right) = \sum_m \sum_n c_m^* c_n \langle \psi_m | \psi_n \rangle = \sum_n |c_n|^2. \quad (\text{A.7})$$

Similarly,

$$\langle \psi | H | \psi \rangle = \left(\sum_m c_m^* \langle \psi_m | \right) \left(H \sum_n c_n |\psi_n\rangle \right) = \sum_m \sum_n c_m^* E_n c_n \langle \psi_m | \psi_n \rangle = \sum_n E_n |c_n|^2. \quad (\text{A.8})$$

By definition the ground state, E_{gs} is the smallest eigenvalue of H , so $E_{gs} \leq E_n$ thus

$$\langle \psi | H | \psi \rangle \geq E_{gs} \sum_n |c_n|^2 = E_{gs}. \quad (\text{A.9})$$

Therefore the variational principle holds for all $|\psi\rangle$. \square

Appendix B

MATLAB Code

The code included implements the VMPS method and is based on the code found in [11]. The original code was written for open boundary conditions and non-trivial modifications have been made to adapt the code to periodic boundary conditions. The code is included with the permission of the original authors of [11]. The top level function is `isingProg(N,D,hs)` which will write the values to a file, from which the various quantities can be plotted.

Program B.1 (Modified from [11])

```
function [P]=calcprojector_onesite(B,Cleft,Cright)
% -----
% [P]=calcprojector_onesite(B,Cleft,Cright)
%
% Computes the projector, P, given the tensor
% network {Cleft,Cright} and ket B.
%
% Parameters
%   B      = The ket of the site that is varied
%   Cleft  = A tensor of the left side of
%            tensor network
%   Cright = A tensor of the right side of
%            tensor network
%
% Returns
%   P      = m-by-n projector
% -----

y=contracttensors(Cleft,6,3,B,3,1);
y=contracttensors(y,7,[2,6,3,4,5],Cright,6,[2,3,4,5,6]);
y=permute(y,[1,3,2]);
y=reshape(y,[prod(size(y)),1]);

Q=orth([y,eye(size(y,1))]);
P=Q(:,2:end);
```

Program B.2 (Modified from [11])

```

function [X,numindX]=contracttensors(X,numindX,indX,Y,numindY,indY)
% -----
% [X,numindX]=contracttensors(X,numindX,indX,Y,numindY,indY)
%
% Contracts indX of tensor X with indY of tensor Y.
%
% Parameters
%   X          = A tensor
%   numindX    = The number of indices of X
%   indX       = The index of X to contract
%   Y          = A tensor
%   numindY    = The number of indices of Y
%   indY       = The index of Y to contract
%
% Returns
%   X          = The contracted tensor of X and Y
%   numindX    = The number of indices of X
% -----

Xsize=ones(1,numindX); Xsize(1:length(size(X)))=size(X);
Ysize=ones(1,numindY); Ysize(1:length(size(Y)))=size(Y);

indXl=1:numindX; indXl(indX)=[];
indYr=1:numindY; indYr(indY)=[];

sizeXl=Xsize(indXl);
sizeX=Xsize(indX);
sizeYr=Ysize(indYr);
sizeY=Ysize(indY);

if prod(sizeX)~=prod(sizeY)
    error('indX and indY are not of same dimension.');
```

```

end

if isempty(indYr)
    if isempty(indXl)
        X=permute(X,[indX]);
        X=reshape(X,[1,prod(sizeX)]);
        Y=permute(Y,[indY]);
        Y=reshape(Y,[prod(sizeY),1]);
        X=X*Y;
        Xsize=1;
        return;
    end
end

```

```

else
    X=permute(X,[indXl,indX]);
    X=reshape(X,[prod(sizeXl),prod(sizeX)]);
    Y=permute(Y,[indY]);
    Y=reshape(Y,[prod(sizeY),1]);
    X=X*Y;
    Xsize=Xsize(indXl);
    X=reshape(X,[Xsize,1]);
    return
end
end

X=permute(X,[indXl,indX]);
X=reshape(X,[prod(sizeXl),prod(sizeX)]);
Y=permute(Y,[indY,indYr]);
Y=reshape(Y,[prod(sizeY),prod(sizeYr)]);

X=X*Y;
Xsize=[Xsize(indXl),Ysize(indYr)];
numindX=length(Xsize);
X=reshape(X,[Xsize,1]);

```

Program B.3 (Modified from [11])

```

function [mps]=createrandommps(N,D,d)
% -----
% [mps]=createrandommps(N,D,d)
%
% Creates a random MPS.
%
% Parameters
%   N   = The number of spins
%   D   = The bond length
%   d   = The spin dimension
%
% Returns
%   mps = The random MPS
% -----

mps=cell(1,N);
for i=1:N
    mps{i}=randn(D,D,d)/sqrt(D);
end

```

Program B.4 (Written by Matthew Low)

```
function E = exactIsing(r,h,N,isGround)
% -----
% E = exactIsing(r,h,N,isGround)
%
% Computes the exact solution to the ground state
% and first excited state energies.
%
% Parameters
%   r      = The r value in the XY model (ising r=1)
%   h      = The field value
%   N      = The number of spins
%   isGround = 1 means compute the ground state,
%             0 means compute the first excited state
%
% Returns
%   E      = Energy
% -----

E = 0;
if (isGround == 1)
    for m=0:(N-1)
        E = E - sqrt((h-cos(2*pi*(m+0.5)/N))^2 + r^2*sin(2*pi*(m+0.5)/N)^2);
    end
else
    E = E + (h-1);
    for m=1:(N-1)
        E = E - sqrt((h-cos(2*pi*m/N))^2 + r^2*sin(2*pi*m/N)^2);
    end
end
end
```

Program B.5 (Modified from [11])

```
function [Cstorage]=initCstorage(mpsB,mpoX,mpsA,N,D)
% -----
% [Cstorage]=initCstorage(mpsB,mpoX,mpsA,N,D)
%
% Initializes storage necessary to compute the
% projector for the first excited state.
%
% Parameters
%   mpsB = The ground state MPS
%   mpoX = The operator (if any)
```

```

%   mpsA = The MPS to calculate
%   N     = The number of spins
%   D     = The bond length
%
% Returns
%   Cstorage = The tensor network used to
%             store the information
% -----

Cstorage=cell(1,N+1);
idD=eye(D);
idid=kron(idD,idD);
idid=reshape(idid,[D 1 D D 1 D]);
Cstorage{1}=idid;
Cstorage{N+1}=idid;

for i=N:-1:2
    if isempty(mpoX), X=[]; else X=mpoX{i}; end
    Cstorage{i}=updateCright(Cstorage{i+1},mpsB{i},X,mpsA{i});
end

```

Program B.6 (Modified from [11])

```

function [Hstorage]=initHstorage(mps,hset,d,D)
% -----
% [Hstorage]=initHstorage(mps,hset,d,D)
%
% Initializes storage necessary to compute Heff
%
% Parameters
%   mps = The MPS to calculate
%   hset = The Hamiltonian
%   d   = The spin dimension
%   D   = The bond length
%
% Returns
%   Hstorage = The tensor network used to
%             store the information
% -----

[M,N]=size(hset);
Hstorage=cell(M,N+1);
idD=eye(D);
idid=kron(idD,idD);
idid=reshape(idid,[D 1 D D 1 D]);

```

```

for m=1:M, Hstorage{m,1}=idid; Hstorage{m,N+1}=idid; end
for j=N:-1:2
    for m=1:M
        h=reshape(hset{m,j},[1,1,d,d]);
        Hstorage{m,j}=updateCright(Hstorage{m,j+1},mps{j},h,mps{j});
    end
end
end

```

Program B.7 (Written by Matthew Low)

```

function [Nstorage]=initNstorage(mps,N,d,D)
% -----
% [Nstorage]=initNstorage(mps,N,d,D)
%
% Initializes storage necessary to compute Neff
%
% Parameters
%   mps = The MPS to calculate
%   N    = The number of spins
%   d    = The spin dimension
%   D    = The bond length
%
% Returns
%   Nstorage = The tensor network used to
%              store the information
% -----

Nstorage=cell(1,N+1);
idD=eye(D);
idid=kron(idD,idD);
idid=reshape(idid,[D 1 D D 1 D]);
id=eye(d);
id=reshape(id,[1,1,d,d]);
Nstorage{1}=idid;
Nstorage{N+1}=idid;

for j=N:-1:2
    Nstorage{j}=updateCright(Nstorage{j+1},mps{j},id,mps{j});
end

```

Program B.8 (Written by Matthew Low)

```

function isingProg(N,D,hs)
% -----
%   isingProg(N,D,hs)
%
%   Computes the economical singular value
%   decomposition of the matrix T.
%
%   Parameters
%   N       = The number of spins
%   D       = Schmidt number approximation
%   hs      = Values of the field h
% -----

% Initiliaze variables
hlen=length(hs);
e0s=zeros(1,hlen);           % Ground state energies
e1s=zeros(1,hlen);           % First excited state energies
ex0s=zeros(1,hlen);          % Exact ground state energies
ex1s=zeros(1,hlen);          % Exact first excited state energies

mx0s=zeros(1,hlen);          % Magnetizations in x-direction
my0s=zeros(1,hlen);          % Magnetizations in y-direction
mz0s=zeros(1,hlen);          % Magnetizations in z-direction

rlen=floor(N/2);
cxx_ir=zeros(rlen,hlen);     % Correlation function: C_xx(r)
cyy_ir=zeros(rlen,hlen);     % Correlation function: C_yy(r)
czz_ir=zeros(rlen,hlen);     % Correlation function: C_zz(r)

cxx_i=zeros(rlen,hlen);      % Correlation function: C_xx(r)
cyy_i=zeros(rlen,hlen);      % Correlation function: C_yy(r)
czz_i=zeros(rlen,hlen);      % Correlation function: C_zz(r)

cxx_r=zeros(rlen,hlen);      % Correlation function: C_xx(r)
cyy_r=zeros(rlen,hlen);      % Correlation function: C_yy(r)
czz_r=zeros(rlen,hlen);      % Correlation function: C_zz(r)

ts=zeros(1,hlen);            % Time per computation
tsum=0;

fprintf('---- Summary ----\n');
fprintf(' N = %g\n D = %g\n Range : %g -> %g\n',N,D,min(hs),max(hs));

```

```

% Initialize variables to compute magnetizations and correlations
sx=[0,1;1,0]; sy=[0,-1i;1i,0];
sz=[1,0;0,-1]; id=eye(2);
mxset=cell(N,N);           % <Mx>
myset=cell(N,N);           % <My>
mzset=cell(N,N);           % <Mz>

cxxset_ir=cell(1,rlen);    % <Sx_i Sx_i+r>
cyyset_ir=cell(1,rlen);    % <Sy_i Sy_i+r>
czzset_ir=cell(1,rlen);    % <Sz_i Sz_i+r>
cxxset_i=cell(1,rlen);     % <Sx_i>
cyyset_i=cell(1,rlen);     % <Sy_i>
czzset_i=cell(1,rlen);     % <Sz_i>
cxxset_r=cell(1,rlen);     % <Sx_i+r>
cyyset_r=cell(1,rlen);     % <Sy_i+r>
czzset_r=cell(1,rlen);     % <Sz_i+r>

for j=1:rlen
    cxxset_ir{j}=cell(N,N);    cyyset_ir{j}=cell(N,N);
    czzset_ir{j}=cell(N,N);    cxxset_i{j}=cell(N,N);
    cyyset_i{j}=cell(N,N);     czzset_i{j}=cell(N,N);
    cxxset_r{j}=cell(N,N);     cyyset_r{j}=cell(N,N);
    czzset_r{j}=cell(N,N);
end

for j=1:N
    for k=1:N
        mxset{j,k}=id;        myset{j,k}=id;        mzset{j,k}=id;
        for l=1:rlen
            cxxset_ir{l}{j,k}=id;    cyyset_ir{l}{j,k}=id;
            czzset_ir{l}{j,k}=id;    cxxset_i{l}{j,k}=id;
            cyyset_i{l}{j,k}=id;     czzset_i{l}{j,k}=id;
            cxxset_r{l}{j,k}=id;     cyyset_r{l}{j,k}=id;
            czzset_r{l}{j,k}=id;
        end
    end
    mxset{j,j}=sx;    myset{j,j}=sy;    mzset{j,j}=sz;
    for k=1:rlen
        cxxset_ir{k}{j,j}=sx;    cxxset_ir{k}{j,mod(j+k-1,N)+1}=sx;
        cyyset_ir{k}{j,j}=sy;    cyyset_ir{k}{j,mod(j+k-1,N)+1}=sy;
        czzset_ir{k}{j,j}=sz;    czzset_ir{k}{j,mod(j+k-1,N)+1}=sz;
        cxxset_i{k}{j,j}=sx;    cxxset_r{k}{j,mod(j+k-1,N)+1}=sx;
        cyyset_i{k}{j,j}=sy;    cyyset_r{k}{j,mod(j+k-1,N)+1}=sy;
        czzset_i{k}{j,j}=sz;    czzset_r{k}{j,mod(j+k-1,N)+1}=sz;
    end
end
end

```

```

% Calculate energies, magnetizations, and correlations
for j=1:hlen
    tic;
    fprintf('Step %g of %g\n',j,hlen);
    [mps0, mps1, e0s(j), e1s(j)] = mpsIsing(N, D, hs(j));
    ex0s(j) = exactIsing(1,hs(j),N,1);
    ex1s(j) = exactIsing(1,hs(j),N,0);

    fprintf('Calculating magnetizations.\n');
    mx0s(j) = expectationvalue(mps0,mxset)/N;
    my0s(j) = expectationvalue(mps0,myset)/N;
    mz0s(j) = expectationvalue(mps0,mzset)/N;

    for k=1:rlen
        fprintf('Calculating correlations for r=%i.\n',k);
        cxx_ir(k,j) = expectationvalue(mps0,cxxset_ir{k})/N;
        cyy_ir(k,j) = expectationvalue(mps0,cyyset_ir{k})/N;
        czz_ir(k,j) = expectationvalue(mps0,czzset_ir{k})/N;

        cxx_r(k,j) = expectationvalue(mps0,cxxset_r{k})/N;
        cyy_r(k,j) = expectationvalue(mps0,cyyset_r{k})/N;
        czz_r(k,j) = expectationvalue(mps0,czzset_r{k})/N;

        cxx_i(k,j) = expectationvalue(mps0,cxxset_i{k})/N;
        cyy_i(k,j) = expectationvalue(mps0,cyyset_i{k})/N;
        czz_i(k,j) = expectationvalue(mps0,czzset_i{k})/N;
    end

    ts(j)=toc;
    tsum=tsum+ts(j);
    fprintf('Time = %g sec\n',ts(j));
end

% Save values to file
fprintf('Done. Writing to file.\n');
save(sprintf('ising_files/ising_N%g_D%g',N,D), ...
    'hs','e0s','ex0s','e1s','ex1s','ts','N','D', ...
    'mx0s','my0s','mz0s','cxx_ir','cyy_ir','czz_ir', ...
    'cxx_i','cyy_i','czz_i','cxx_r','cyy_r','czz_r');

```

Program B.9 (Modified from [11])

```

function [E,mps]=minimizeE(hset,D,precision,mpsB)
% -----
% [E,mps]=minimizeE(hset,D,precision,mpsB)
%
% Computes the minimization sweep to find the
% ground state or first excited state energy
% and MPS.
%
% Parameters
%   mpsB      = [] or the ground state MPS
%   hset      = The Hamiltonian
%   precision = The spin dimension
%   D         = The bond length
%
% Returns
%   E        = The energy
%   mps      = The MPS
% -----

[M,N]=size(hset);
d=size(hset{1,1},1);
mps=createrandommps(N,D,d);
mps=prepare(mps);

% Storage initialization
Hstorage=initHstorage(mps,hset,d,D);
Nstorage=initNstorage(mps,N,d,D);
if ~isempty(mpsB), Cstorage=initCstorage(mps,[],mpsB,N,D); end
P=[];

% Optimization sweeps
while 1
    Evalues=[];

    % Sweep: 1 -> N-1
    for j=1:(N-1)

        % Projector calculation
        if ~isempty(mpsB)
            B=mpsB{j};
            Cleft=Cstorage{j};
            Cright=Cstorage{j+1};
            P=calcpjector_onsite(B,Cleft,Cright);
        end
    end
end

```

```

end

% Local minimization
Hleft=Hstorage(:,j);
Hright=Hstorage(:,j+1);
Nleft=Nstorage{j};
Nright=Nstorage{j+1};
hsetj=hset(:,j);
[A,E]=minimizeE_onesite(hsetj,Hleft,Hright,P,Nleft,Nright);
[A,U]=prepare_onesite(A,'lr');
mps{j}=A;
Evalues=[Evalues,E];

% Storage update
for m=1:M
    h=reshape(hset{m,j},[1,1,d,d]);
    Hstorage{m,j+1}=updateCleft(Hleft{m},A,h,A);
    id=reshape(eye(d),[1,1,d,d]);
    Nstorage{j+1}=updateCleft(Nleft,A,id,A);
end
if ~isempty(mpsB)
    Cstorage{j+1}=updateCleft(Cleft,A,[],B);
end
end

% Sweep: N -> 2
for j=N:(-1):2

% Projector calculation
if ~isempty(mpsB)
    B=mpsB{j};
    Cleft=Cstorage{j};
    Cright=Cstorage{j+1};
    P=calcprojector_onesite(B,Cleft,Cright);
end

% Local minimization
Hleft=Hstorage(:,j);
Hright=Hstorage(:,j+1);
Nleft=Nstorage{j};
Nright=Nstorage{j+1};
hsetj=hset(:,j);
[A,E]=minimizeE_onesite(hsetj,Hleft,Hright,P,Nleft,Nright);
[A,U]=prepare_onesite(A,'rl');
mps{j}=A;
Evalues=[Evalues,E];

```

```

% Storage update
for m=1:M
    h=reshape(hset{m,j},[1,1,d,d]);
    Hstorage{m,j}=updateCright(Hright{m},A,h,A);
    id=reshape(eye(d),[1,1,d,d]);
    Nstorage{j}=updateCright(Nright,A,id,A);
end
if ~isempty(mpsB)
    Cstorage{j}=updateCright(Cright,A,[],B);
end
end

% If energies are converging, complete MPS and return
if (std(Evalues)/abs(mean(Evalues))<precision)
    mps{1}=contracttensors(mps{1},3,2,U,2,1);
    mps{1}=permute(mps{1},[1,3,2]);
    break;
end
end
end

```

Program B.10 (Modified from [11])

```

function [A,E]=minimizeE_onesite(hsetj,Hleft,Hright,P,Nleft,Nright)
% -----
% [A,E]=minimizeE_onesite(hsetj,Hleft,Hright,P,Nleft,Nright)
%
% Computes the minimization sweep to find the
% ground state or first excited state energy
% and MPS.
%
% Parameters
%   hsetj = The Hamiltonian
%   Hleft  = The left side of the Heff tensor
%   Hright = The right side of the Heff tensor
%   P      = [] or the projector
%   Nleft  = The left side of the Neff tensor
%   Nright = The right side of the Neff tensor
%
% Returns
%   A = Orthogonal MPS tensor
%   E = The energy
% -----

DAL=size(Hleft{1},1);

```

```

DAr=size(Hright{1},1);
d=size(hsetj{1},1);
M=size(hsetj,1);
id=eye(d);

% Compute Heff
Heff=0;
for m=1:M
    Heffm=contracttensors(Hleft{m},6,[2,4,5,6],Hright{m},6,[2,4,5,6]);
    Heffm=contracttensors(Heffm,5,5,hsetj{m},3,3);
    Heffm=permute(Heffm,[1,3,5,2,4,6]);
    Heffm=reshape(Heffm,[DA1*DAr*d,DA1*DAr*d]);
    Heff=Heff+Heffm;
end

% Compute Neff
Neff=contracttensors(Nleft,6,[2,4,5,6],Nright,6,[2,4,5,6]);
Neff=contracttensors(Neff,5,5,id,3,3);
Neff=permute(Neff,[1,3,5,2,4,6]);
Neff=reshape(Neff,[DA1*DAr*d,DA1*DAr*d]);

% Projection on orthogonal subspace to fs
if ~isempty(P)
    Heff=P'*Heff*P;
    Neff=P'*Neff*P;
end

% Enforce hermiticity
Neff=(Neff+Neff')/2.0;
Heff=(Heff+Heff')/2.0;

% Improve Neff conditioning
tol=1e-10;
[vs,ds]=eig(Neff);
ds=diag(ds);
for j=1:length(ds)
    if ds(j) >= tol
        ind=j-1;
        break;
    end
end
W=vs(:,1:ind);
Q=orth([W,eye(size(W,1))]);
P2=Q(:,ind+1:end);

% Projection into well conditioned subspace

```

```

Neff=P2'*Neff*P2;
Heff=P2'*Heff*P2;

% Enforce hermiticity
Neff=(Neff+Neff')/2.0;
Heff=(Heff+Heff')/2.0;

% Optimization
[vs ds]=eig(Heff,Neff);
E=ds(1,1);
A=vs(:,1);

% Project back into original space
A=P2*A;
if ~isempty(P), A=P*A; end
A=A/sqrt(A'*A);
A=reshape(A,[DA1,DAr,d]);

```

Program B.11 (Modified from [11])

```

function [mps0, mps1, E0, E1] = mpsIsing(N, D, h, precision)
% -----
% [mps0, mps1, E0, E1] = mpsIsing(N, D, h, precision)
%
% Computes the ground state and first excited state
% of the Ising model given a field strength.
%
% Parameters
%   N      = The number of spins
%   D      = The bond length
%   h      = The field strength
%   precision = The spin dimension
%
% Returns
%   mps0 = The ground state MPS
%   mps1 = The first excited state MPS
%   E0   = The ground state energy
%   E1   = The first excited state energy
% -----

% Pauli matrices
sx=[0,1;1,0]; sy=[0,-1i;1i,0]; sz=[1,0;0,-1]; id=eye(2);

% Define hamiltonian
M=2*N;

```



```

hset=cell(M,N);
for m=1:M, for j=1:N, hset{m,j}=id; end; end
for j=1:(N-1)
    hset{2*(j-1)+1,j}=-sx; hset{2*(j-1)+1,j+1}=sx;
    hset{2*(j-1)+2,j}=-h*sz;
end
hset{2*N-1,1}=-sx; hset{2*N-1,N}=sx;
hset{2*N,N}=-h*sz;

% Ground state energy
randn('state',0)
[E0,mps0]=minimizeE(hset,D,precision*1e-2,[]); % modified
fprintf('E0 = %g\n',E0);

% First excited state
[E1,mps1]=minimizeE(hset,D,precision,mps0);
fprintf('E1 = %g\n',E1);

```

Program B.12 (Modified from [11])

```

function [mps]=prepare(mps,direction)
% -----
% [mps]=prepare(mps,direction)
%
% Prepares a MPS using gauge transformations such
% that Neff is the identity for the first spin.
%
% Parameters
%   mps      = The MPS to prepare
%   direction = 'lr' or 'rl'
%
% Returns
%   mps = The MPS
% -----

% Default preparation
if nargin < 2
    direction = 'rl';
end

N=length(mps);
switch direction
    case 'lr'
        for i=1:N-1
            [mps{i},U]=prepare_onsite(mps{i},'lr');

```

```

        mps{i+1}=contracttensors(U,2,2,mps{i+1},3,1);
    end
    case 'rl'
        for i=N:-1:2
            [mps{i},U]=prepare_onesite(mps{i},'rl');
            mps{i-1}=contracttensors(mps{i-1},3,2,U,2,1);
            mps{i-1}=permute(mps{i-1},[1,3,2]);
        end
    end
end

```

Program B.13 (Modified from [11])

```

function [B,U,DB]=prepare_onesite(A,direction)
% -----
% [B,U,DB]=prepare_onesite(A,direction)
%
% Performs a singular value decomposition on
% the tensor A.
%
% Parameters
%   A          = The MPS tensor
%   direction = 'lr' or 'rl'
%
% Returns
%   B  = Unitary tensor
%   U  = Non-unitary
%   DB = The number of singular values
% -----

[D1,D2,d]=size(A);
switch direction
    case 'lr'
        A=permute(A,[3,1,2]); A=reshape(A,[d*D1,D2]);
        [B,S,U]=svd2(A); DB=size(S,1);
        B=reshape(B,[d,D1,DB]); B=permute(B,[2,3,1]);
        U=S*U;
    case 'rl'
        A=permute(A,[1,3,2]); A=reshape(A,[D1,d*D2]);
        [U,S,B]=svd2(A); DB=size(S,1);
        B=reshape(B,[DB,d,D2]); B=permute(B,[1,3,2]);
        U=U*S;
end

```

Program B.14 (Modified from [11])

```
function [U,S,V]=svd2(T)
% -----
% [U,S,V]=svd2(T)
%
% Computes the economical singular value
% decomposition of the matrix T.
%
% Parameters
%   T    = A m-by-n matrix
%
% Returns
%   U    = Orthogonal m-by-m matrix
%   S    = Diagonal n-by-n matrix
%   V    = Orthogonal n-by-n matrix
% -----

[m,n]=size(T);

if m>=n           % LR Case
    [U,S,V]=svd(T,0);
else             % RL Case
    [V,S,U]=svd(T',0);
end

V=V';
N=sqrt(trace(S*S));
S=S/N;
```
