Time Evolution of a Device for Remote Detection on Atomic Spin Chains using Matrix Product States

Justin Bouwmeester - 4709098

Bachelor Thesis
TU Delft
Faculty EEMCS & TNW
BSc Applied Mathematics & Applied Physics

July 2020
Supervisors:
Dr. J.M. Thijssen & Dr. N.V. Budko

Abstract

In this thesis the matrix product state(MPS) formalism is applied to simulate a device created for remote detection in atomic spin chains. Although simulation of the full device quickly becomes infeasible for direct algorithms, MPS allowed for highly accurate simulations while requiring only modest computational resources. It has been verified that the simulation conserves energy and maintains normalization. Furthermore, the simulation is shown to successfully perform both a coherent real-time evolution and iteration towards the ground state. Thus MPS is a very promising tool that has potential to serve as a guide for further experiment as well as a tool to understand experimental results.

Contents

1 Introduction

In physics, experiment and theory, including computation, are complementary ways of studying phenomena and systems in nature. An interesting such system, which is the topic of this thesis, is a chain of iron atoms which has been studied experimentally in the Otte group in Delft with an emphasis on the time evolution of atomic spins[1]. One challenge of simulating such a quantum system is that the dimensions of the Hilbert space scales exponentially with system size, rendering direct simulations unfeasible for large systems.

In this thesis we simulate the system used in the experiment for the remote detection on atomic spin chains. The device is a one-dimensional chain of Fe atoms that have spin 2[1]. The chain consists of three parts: an input, an output, and a reset. Due to the number of atoms in the system, direct simulation is limited to the input chain. This has been done in the Otte group and their results will be used to verify an advanced, more powerful method used to simulate the complete device.

In order to simulate the complete device efficiently the matrix product state (MPS) formalism is used. This formalism allows us to limit the size of the Hilbert space we are simulating, resulting in a smaller portion that scales polynomially as opposed to exponentially[2]. A crucial parameter, χ, of the method controls the truncation of the Hilbert space. Both real-time evolution of the system and iteration to the ground state have been applied to the system.

This work is a follow-up of the work previously done by Pim Vree, who has previously used MPS to find the ground state energy of a spin-$\frac{1}{2}$ Ising chain[3].

This thesis is structured as follows: chapter 2 contains the experiment, chapter 3 consists of some mathematical concepts required to derive the MPS formalism. In chapter 4 we set up the MPS formalism, and then we explain the time-evolution methods and computations in chapter 5. The simulation results can be found in chapter 6, followed by a conclusion in chapter 7.

2 Experiment

The aim to the experiment is to perform remote detection on an atomic spin chain with a scanning tunneling microscope(STM)[1]. Often experimentalists want to measure the effect of applying an excitation to an atom at a different point in the chain. Because of the speed at which excitations move through a spin chain, it is often impossible to apply an excitation with the STM tip and to move the tip to perform a measurement multiple atoms away. The experiment therefore introduces a memory component that allows measurement of results seconds after the excitation is applied. The experiment has been carried out by R. Elberste and D. Coffey in the Otte group of TU Delft.

The device consists of three parts: an input, an output, and a reset. This can be seen in figure 1. In this case the input contains $n_{in} = 5$, the ouput contains $n_{out} = 8$, and the reset contains $n_{reset} = 3$ Fe atoms.

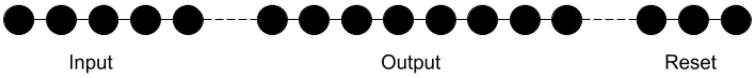


Figure 1: A chain of length 16 with $n_{in} = 5, n_{out} = 8, n_{reset} = 3$. Full lines indicate a coupling of J_i and dashed lines indicate a coupling of J'.

The input of the device is where the excitation is applied. The output is used as the memory component and stores the result for later measurement. The reset is used to return the system to its initial state.

The chain that was tested consists of Fe atoms that behave as spin-2 particles, so each particle has 5 basisstates. Neighbouring particles inside of the same parts are all coupled anti-ferromagnetically with a coupling of $J = 0.7\ meV$. Neighbouring atoms between different parts of the system are coupled ferromagnetically with a weaker coupling of $J' = -0.05\ meV$. A magnetic field has been applied in the z-direction to fix this as quantization axis.

The experimentalists require that the length of the full chain is even. This is to ensure that the ground state is degenerate, which is important as a system will always move to the ground state due to losses. If the ground state is not degenerate then the system will lose its memory property because of always returning to the same state. They furthermore require that the length of the input and reset chain is odd and that the length of the output chain is even. By doing so they obtain a system where the difference between the degenerate ground states only occurs in the output lead. The effect of this is that after an excitation of the input chain, it will relax to its original state while the output chain can end up in either degenerate state depending on the excitation.

The chain that is studied consists of an input chain of length $3 \leq n_{in} \leq 9$, and output chain of length $n_{out} = 8$ and a reset of length $n_{reset} = 3$.

Hamiltonian

The experimental Hamiltonian consists of three term; a Zeeman splitting, an anisotropy term, and a nearest neighbour coupling term. For a chain of length L the Hamiltonian is

$$\hat{H} = \sum_i^L (\hat{H}_i^{\text{Zeeman}} + \hat{H}_i^{\text{Anis}}) + \sum_i^{L-1} \hat{H}_i^{\text{Coupling}}$$

with

$$\begin{aligned}\hat{H}_i^{\text{Zeeman}} &= g\mu_b \vec{B}_i \cdot \vec{S}_i \\ \hat{H}_i^{\text{Anis}} &= D S_i^{z2} + E(S_i^{x2} - S_i^{y2}) \\ \hat{H}_i^{\text{Coupling}} &= J_i(\vec{S}_i \cdot \vec{S_{i+1}}).\end{aligned}$$

The values of each constant in the Hamiltonian are listed in the table below. The definition and explanation of the spin matrices will be given in section 3.3.

Table 1: Definition, explanation and numerical value of all the variables that are used in our model.

Variable	Value	Unit	Explanation
g	2.11	dimensionless	g-factor
μ_B	0.0578838	meVT^{-1}	Bohr magneton
B_i	0.5-1	T	Magnetic field strength
D	-1.77	meV	Anisotropy parameter
E	0.33	meV	Anisotropy parameter
J_i	0.7	meV	coupling between neighbouring states
J'	-0.05	meV	coupling between input-output-reset

As previously noted, the degeneracy in the ground state of the chain under this Hamiltonian can be fully described by the output chain. For this purpose the 5 $\hat{S}_z$ eigenstates of each site are labelled $|-2\rangle, |-1\rangle, ..., |2\rangle$. In the ground state the output chain can be in two-states: $N_A = (2, -2, 2, -2, 2, -2, 2, -2)$ or $N_B = (-2, 2, -2, 2, -2, 2, -2, 2)$. The probability that the state of the output chain switches from N_A to N_B or vice versa will be referred to as the switching probabilities. In the experiment, the system is initialized in the ground state described by N_A.

If the spin of two neighbouring particles in different chains is the same they are called "happy". Because of ferromagnetic coupling between different chains this is energetically favourable. If the spins are different however, they are referred to as "unhappy": this is energetically unfavorable. If an excitation is applied to the input chain, then the N_B state will be energetically favourable over the N_A state until the system relaxes to the ground state. All simulations will be performed with the input in the happy state unless mentioned otherwise.

Throughout the experiment, the S^z operator is measured. This is used to estimate the switching probabilities. For example if the system is in state N_A then we expect for a measurement of S^z on the first site of the output to give us the value $2\hbar$. If we are in state N_B we would expect for a measurement of S^z on the first site of the output to instead give us the value $-2\hbar$.

3 Mathematical Background

In this chapter we outline the main mathematical concepts that are required in setting up a matrix product state(MPS) formalism. In particular the singular value decomposition and the concept of matricization. We furthermore construct the spin matrices and introduce error estimation through Richardson extrapolation.

All proofs related to the singular value decomposition can be found in the book "Foundations of Data Science"[4].

3.1 The Singular Value Decomposition

Singular Vectors

Let A be a complex $n \times m$ matrix of rank r. In order to derive the singular value decomposition we first define the singular vectors of a A. We define $\mathbf{v_1}$, the first right singular vector of A, as follows:

$$\mathbf{v_1} = \underset{|\mathbf{v}|=1}{\arg\max} |A\mathbf{v}|.$$

The arg max function returns the value, or in this case vector, for which the argument is maximal. We use $|\dots|$ to denote the euclidean norm. We furthermore define $\sigma_1(A) = |A\mathbf{v_1}|$ as the first singular value.

The second right singular vector is then chosen similarly, but under an additional condition of being orthogonal to the first singular vector:

$$\mathbf{v_2} = \underset{\mathbf{v} \perp \mathbf{v_1}, |\mathbf{v}|=1}{\arg\max} |A\mathbf{v}|.$$

We denote $\sigma_2(A) = |A\mathbf{v_2}|$ as the second singular value.

In general the k^{th} right singular vector is defined as:

$$\mathbf{v_k} = \underset{\mathbf{v} \perp \mathbf{v_1}, \mathbf{v_2}, \dots, \mathbf{v_{k-1}}, |\mathbf{v}|=1}{\arg\max} |A\mathbf{v}|$$

with singular value $\sigma_k(A) = |A\mathbf{v_k}|$.

As A is of rank r we note that there are r different singular vector and singular values. By construction we also note that $\mathbf{v_1}, \mathbf{v_2}, \dots, \mathbf{v_r}$ are orthogonal.

Using the right singular vector $\mathbf{v_i}$ we can also define the left singular vector $\mathbf{u_i}$:

$$\mathbf{u_i} = \frac{1}{\sigma_i(A)} A\mathbf{v_i}.$$

Theorem 1. *The left singular vectors of A, $\mathbf{u_1}, \mathbf{u_2}, \dots, \mathbf{u_r}$ are orthogonal.*

The orthogonality of both the left singular vectors and the right singular vectors will be important for performing fast calculations using matrix product states.

Singular Value Decomposition

We can now define the compact singular value decomposition:

Theorem 2. *(Compact Singular Value Decomposition) Let A be a $n \times m$ matrix with right singular vectors $\mathbf{v_1}, \mathbf{v_2}, \dots, \mathbf{v_r}$, left singular vectors $\mathbf{u_1}, \mathbf{u_2}, \dots, \mathbf{u_r}$, and corresponding singular values $\sigma_1, \sigma_2, \dots, \sigma_r$ then:*

$$A = \sum_{i=1}^{r} \sigma_i \mathbf{u_i} \mathbf{v_i}^\dagger$$

We can write this sum in matrix notation as $A = U\Lambda V^\dagger$ where Λ is a diagonal matrix containing the singular values and U and V are matrices whose columns are the left and right singular vectors respectively. The † symbol is used to denote the Hermitian transpose. The following diagram shows the structure of the matrices involved in the SVD:

$$\underset{(n \times m)}{\begin{bmatrix} \quad\quad\quad & \\ \\ \\ \\ \\ \\ \end{bmatrix}} = \underset{(n \times r)}{\begin{bmatrix} \quad\quad & \\ \\ \\ \\ \\ \\ \end{bmatrix}} \underset{(r \times r)}{\begin{bmatrix} \quad\quad & \\ \\ \\ \end{bmatrix}} \underset{(r \times m)}{\begin{bmatrix} \quad\quad\quad\quad & \\ \\ \\ \end{bmatrix}}$$

$$= U\Lambda V^\dagger.$$

This matrix decomposition of A is not unique, however we can always construct the matrices such that Λ contains the singular values in decreasing order. For the rest of this thesis we will assume that this is the case.

Approximation

The SVD allows us to represent the matrix A in a more compact form through the means of obtaining the best rank k approximation of the matrix. We define A_k as follows:

$$A_k = \sum_{i=1}^{k} \sigma_i \mathbf{u_i} \mathbf{v_i}^\dagger$$

This is clearly a matrix of rank k.

Under the Frobenius norm A_k is the best rank k approximation for A. This can be seen in the following theorem:

Theorem 3. *For any matrix B of at most rank k:*

$$||A - A_k||_F \leq ||A - B||_F$$

Schmidt Decomposition

We can use the singular value decomposition to obtain the following theorem[5]:

Theorem 4. *(Schmidt decomposition) Let $H = H_1 \otimes H_2$ be a Hilbert space with H_1 and H_2 finite dimensional Hilbert spaces with not necessarily the same dimension. Let $w \in H$ normalized, then there exist orthonormal sets $\{u_1, ..., u_r\} \in H_1$ and $\{v_1, ..., v_r\} \in H_2$ such that:*

$$w = \sum_{i=1}^{r} \alpha_i u_i \otimes v_i$$

where α_i are real, strictly positive, and normalized such that $\sum_i |\alpha_i|^2 = 1$. We define r as the Schmidt rank of the state.

3.2 Matricization

In general our wavefunction is described by a tensor. However, in order to apply the singular value decomposition we require a matrix instead. Therefore, we need to transform the tensor into a matrix that is appropriate for our calculations. The method we use is through matricization/unfolding. Let $X \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_n}$ with $I_i \in \mathbb{N}$ for all $1 \leq i \leq n$. The index i_l runs from 1 to I_l. We define the k-mode matricization by mapping the element $(i_1, ..., i_{k-1}, i_k, i_{k+1}, ..., i_n)$ to (i_k, j) with[6]:

$$j = 1 + \sum_{m=1, m \neq k}^{n} ((i_m - 1) \prod_{l=1, l \neq k}^{m-1} I_l) \tag{1}$$

Our resultant matrix will have dimensions $I_k \times (I_1 \cdots I_{k-1} I_{k+1} \cdots I_n)$. For ease of notation we will write the resultant matrix of k-mode matricization of $X_{i_1 i_2 \dots i_n}$ as $X_{i_k,(i_1 \dots i_{k-1} i_{k+1} \dots i_n)}$.

In order to visualize this operation we introduce mode-k fibers[7]. The mode-k fibers are the result of fixing each parameter save the k^{th}. For a second order tensor the mode-1 fibers are the columns for example. Figure 2 shows the three different mode-k fibers for a 3^{rd} order tensor.

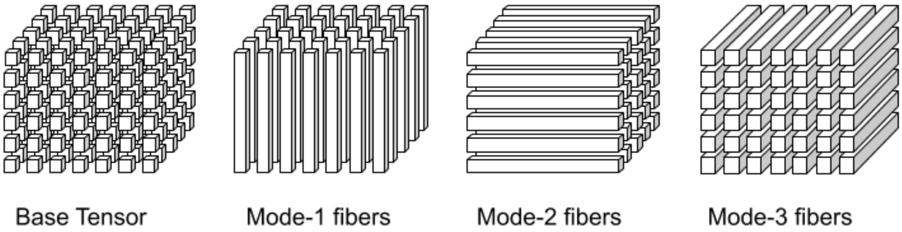


Figure 2: The different mode-k fibers for a third order tensor. Each cube in the base tensor is a value.

In practice mode-k matricization rearranges the mode-k fibers into a matrix. We can see this from equation (1): each element in a specific mode-k fiber will share the same j value. We see that the columns of the resultant matrix contain the fibers in reverse lexicographical order [8]. Figure 3 shows mode-2 matricization of a $(4 \times 5 \times 3)$ tensor into a (5×12) matrix.

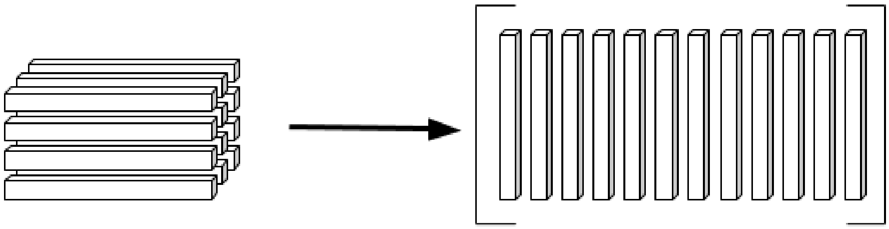


Figure 3: Rearranging mode-2 fibers for mode-2 matricization of a third order tensor.

Canonical Matricization

Outside of k-mode matricization we also employ mode-(1,2,..,k) matricization, also known as mode-k canonical matricization[8]. In this form of matricization we map an element $(i_1, ..., i_{k-1}, i_k, i_{k+1}, ..., i_n)$ to (l, j) with:

$$l = 1 + \sum_{m=1}^{k} ((i_m - 1) \prod_{l=1}^{m-1} I_l)$$
$$j = 1 + \sum_{m=k+1}^{n} ((i_m - 1) \prod_{l=k+1}^{m-1} I_l)$$

Our resultant matrix will be of dimensions $(I_1 \cdots I_k) \times (I_{k+1} \cdots I_n)$. As notation we write the mode-k canonical matricization of $X_{i_1 i_2 \dots i_n}$ as $X_{(i_1 \dots i_k),(i_{k+1} \dots i_n)}$, analogously to mode-k matricization.

Slices

Outside of matricization, we also introduce the concept of slices of third order tensors[7]. A slice of a third order tensor is obtained by keeping one index constant. Slices were the first index of the tensor is kept constant are known as horizontal slices. For the second and third index the names are lateral and frontal slices respectively. This is shown in figure 4.

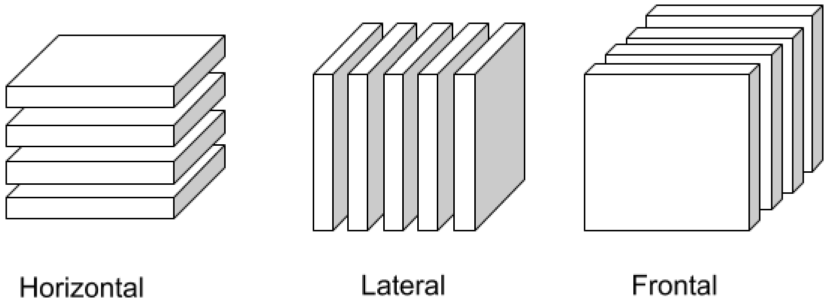


Figure 4: The three types of slices that can be obtained from a third order tensor.

3.3 Spin Matrices

In chapter 2 we have seen that the Hamiltonian primarily consists of the spin operators $\hat{S}_x, \hat{S}_y, \hat{S}_z$. In order to perform calculations with these operators we need to obtain their matrix form. The spin operator $\hat{S}_z$ is defined as seen here[9]:

$$\hat{S}_z \left|s, m_s\right\rangle = \hbar m_s \left|s, m_s\right\rangle$$

We can see that $|s, m_s\rangle$ are eigenstates of the $\hat{S}_z$ operator. To obtain the elements of our S_z matrix we can simply perform the following inner product:

$$S_{z\,a,b} = \left\langle s, m_a\right| \hat{S}_z \left|s, m_b\right\rangle .$$

Here a and b are the indices of the S_z matrix.

To obtain the S_x and S_y matrix we first define the $\hat{S}_+$ and $\hat{S}_-$ operators. They are defined as follows[9]:

$$\begin{aligned}\hat{S}_+ \left|s, m_s\right\rangle &= \hbar\sqrt{s(s+1) - m_s(m_s+1)}\left|s, m_s+1\right\rangle \\ \hat{S}_- \left|s, m_s\right\rangle &= \hbar\sqrt{s(s+1) - m_s(m_s-1)}\left|s, m_s-1\right\rangle\end{aligned}$$

Here we note that since $|m_s| \leq s$ we find that $\hat{S}_+ \left|s, m_s = s\right\rangle = 0$ and $\hat{S}_- \left|s, m_s = -s\right\rangle = 0$. the matrices are constructed in the same way as for S_z. Using these matrices we can obtain the S_x and S_y matrices through the following two equations[9]:

$$\begin{aligned}S_x &= \frac{S_+ + S_-}{2} \\ S_y &= \frac{S_+ - S_-}{2i}\end{aligned}$$

As the experiment concerns itself with spin 2 particles, these are listed below:

$$S_x = \frac{\hbar}{2}\begin{bmatrix} 0 & 2 & 0 & 0 & 0 \\ 2 & 0 & \sqrt{6} & 0 & 0 \\ 0 & \sqrt{6} & 0 & \sqrt{6} & 0 \\ 0 & 0 & \sqrt{6} & 0 & 2 \\ 0 & 0 & 0 & 2 & 0 \end{bmatrix}$$

$$S_y = \frac{\hbar}{2}\begin{bmatrix} 0 & -2i & 0 & 0 & 0 \\ 2i & 0 & -\sqrt{6}i & 0 & 0 \\ 0 & \sqrt{6}i & 0 & -\sqrt{6}i & 0 \\ 0 & 0 & \sqrt{6}i & 0 & -2i \\ 0 & 0 & 0 & 2i & 0 \end{bmatrix}$$

$$S_z = \hbar\begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -2 \end{bmatrix}$$

This matrix representation of the system is not unique. The following commutation relations must hold for any three spin matrices[9]:

$$[S_x, S_y] = i\hbar S_z$$
$$[S_y, S_z] = i\hbar S_x$$
$$[S_z, S_x] = i\hbar S_y$$

with

$$[A, B] = AB - BA,$$

the commutator between A and B.

Richardson error estimation

If we use the numerical method $Q(h)$ to approximate the unknown value M, then we can assume that the following holds[10]:

$$M - Q(h) = c_p h^p + \mathcal{O}h^{p+1}.$$

with $c_p \neq 0$ and $p \in \mathbb{N}$. This formula allows us to determine the value of p. We firstly note that for sufficiently small values of h the effect of the $\mathcal{O}(h^{p+1})$ part is negligible. We can then compute $Q(h), Q(2h)$ and $Q(4h)$ to obtain the following:

$$Q(2h) - Q(4h) = c_p(2h)^p(2^p - 1),$$
$$Q(h) - Q(2h) = c_p(h)^p(2^p - 1).$$

So we quickly obtain[10]:

$$\frac{Q(2h) - Q(4h)}{Q(h) - Q(2h)} = 2^p.$$

This formula will be used to test the order of the errors in time evolution of the chain. This allows us to see if time evolution is implemented properly or not. We will state the expected order of error in chapter 5. If Richardson estimation gives similar results to the expected error, then we know that the implementation of the numerical methods works properly.

4 Matrix Product States

It is our aim to simulate the time-evolution of long spin chains. The technique of MPS is very helpful in this context. In this chapter we use MPS to drastically reduce the dimension of the Hilbert space we consider in our calculations.

Let us consider a one-dimensional chain of particles with length L. For each site we denote the orthonormal basis by $\{|n_i\rangle\}$. We can write a given wave function as:

$$|\Psi\rangle = \sum_{n_1,n_2,...,n_L} c_{n_1,n_2,...,n_L} |n_1, n_2, ..., n_L\rangle .$$

With $|n_1, n_2, ..., n_L\rangle = |n_1\rangle \otimes |n_2\rangle \otimes ... \otimes |n_L\rangle$. Here we store our $c_{n_1,n_2,...,n_L}$ in the L^{th}-order tensor C. If each site can be in d states, then C will contain d^L elements. Thus, in order to simulate an arbitrary given state we are required to store an amount of information that scales exponentially with the length of the chain. This rapidly becomes unfeasible. One method to approximate and simulate this wave function without storage requirements scaling exponentially with length is through matrix product states.

4.1 The Matrix Product States Formalism

The matrix product states formalism heavily relies on the singular value decomposition. Our aim is to write an element $c_{n_1,n_2,...,n_L}$ of the tensor C as

$$c_{n_1,n_2,...,n_L} = \Gamma_{n_1}\Lambda_1\Gamma_{n_2}\Lambda_2...\Gamma_{n_L} \tag{2}$$

introducing a matrix Γ_{n_i} for each i and corresponding n_i[11]. This is known as Vidal's decomposition.

Let $C \in \mathbb{R}^{I_1,I_2,...,I_L}$ be our coefficient tensor with I_i the number of states of site i. We will label the states $n_i = 1, 2, ..., I_i$. For convenience we denote it as $C^{(1)}$. Now let $M = C_{[1]} \in \mathbb{R}^{I_1\times(I_2...I_L)}$ be the mode-1 matricization of $C^{(1)}$. We denote the rank of M as R_1. We now apply a singular value decomposition on M to obtain[11]:

$$m_{n_1,(n_2n_3...n_L)} = \sum_{r_1=1}^{R_1} U_{n_1,r_1}\Lambda^{(1)}_{r_1,r_1}V^{\dagger}_{r_1,(n_2n_3...n_L)}.$$

We note $U \in \mathbb{R}^{I_1\times R_1}$, $\Lambda \in \mathbb{R}^{R_1\times R_1}$, $V^\dagger \in \mathbb{R}^{R_1\times(I_2I_3...I_L)}$. We now find that each row of U is related only to a specific value of n_1. We define each A_{n_1} vector by its respective row in U. This yields us the I_1 A_{n_1} matrices that belong to our state. We quickly note that these are all row vectors of length R_1

We now define $C^{(2)} = \Lambda^{(1)}V^\dagger$. We let $M^{(2)} = C_{[2]} \in \mathbb{R}^{(R_1I_2)\times(I_3...I_L)}$ with $C_{[2]}$ the mode-2 canonical matricization of $C^{(2)}$. We denote the rank of $M^{(2)}$ as R_2 and apply a singular value decomposition on $M^{(2)}$ to obtain[11]:

$$m_{(r_1n_2),(n_3...n_L)} = \sum_{r_2=1}^{R_2} U_{(r_1n_2),r_2}\Lambda^{(2)}_{r_2,r_2}V^{\dagger}_{r_2,(n_3...n_L)}.$$

We note that $R_2 \leq \min(R_1I_2, I_3I_4...I_n)$. We now reshape U into a 3^{rd} order tensor in $\mathbb{R}^{I_2\times R_1\times R_2}$. The horizontal slices of this tensor will be denoted as A_{n_2} based on the value of the index n_2 that the slice belongs to. It is clear that $A_{n_2} \in \mathbb{R}^{R_1\times R_2}$ and that we have I_2 of them, one for each eigenstate of site 2.

We let $C^{(3)} = \Lambda^{(2)}V^{\dagger}$ and repeat this process until we have moved through the entire chain. For each site of the chain that is not at an end we obtain a matrix and for the right end of the chain we obtain a column vector. Upon completion we are left with:

$$c_{n_1,n_2,...,n_L} = A_{n_1}A_{n_2}...A_{n_L}$$

This is known as the left canonical decomposition. To change it into Vidal's decomposition we use the $\Lambda^{(i)}$ matrices obtained in the process of writing the left canonical decomposition. Between each A_{n_i} and $A_{n_{i+1}}$ we add $\Lambda^{(i)}(\Lambda^{(i)})^{-1}$. We now write $\Gamma_{n_1} = A_{n_1}$ and $\Gamma_{n_i} = (\Lambda^{(i-1)})^{-1}A_{n_i}$. This yields:

$$\begin{aligned} c_{n_1,n_2,...,n_L} &= A_{n_1}A_{n_2}...A_{n_L} \\ &= A_{n_1}\Lambda^{(1)}(\Lambda^{(1)})^{-1}A_{n_2}\Lambda^{(2)}...(\Lambda^{(L-1)})^{-1}A_{n_L} \\ &= \Gamma_{n_1}\Lambda^{(1)}\Gamma_{n_2}\Lambda^{(2)}...\Gamma_{n_L} \end{aligned}$$

It is clear that this is the same form as equation (2). The wave function in matrix product state form becomes[11]:

$$|\Psi\rangle = \sum_{n_1,...,n_L} \Gamma_{n_1}\Lambda_1\Gamma_{n_2}\Lambda_2...\Gamma_{n_L}\,|n_1,n_2,...,n_L\rangle$$

Properties

Vidal's decomposition has several properties that turn out to be very helpful in simulations.

We firstly see that the Λ matrices are all diagonal. This directly follows from the singular value decomposition. We also know that the singular values inside each Λ matrix are in decreasing order. Furthermore normalization of the wave function means that $||\Lambda||_F = 1$, so we have $\sum_i |\lambda_i|^2 = 1$. We refer to the number of non-zero elements in a Λ matrix as its bond dimension.

The orthogonality of the left and right singular vectors also results in orthogonality in Vidal's decomposition. By the singular value decomposition we know that[11]:

$$\sum_{i=1}^{L}(\Gamma_{n_i})^{\dagger}\Gamma_{n_i} = \mathbb{I}$$

$$\sum_{i=1}^{L}\Gamma_{n_i}(\Gamma_{n_i})^{\dagger} = \mathbb{I}$$

with $\mathbb{I}$ the identity matrix. These are known as the left-orthonormal and right-orthonormal constraints respectively. These constraints will be very important for time evolution and calculating the expectation of operators.

In Vidal's decomposition each Λ_i is a $R_i \times R_i$ matrix and each Γ_i is a $R_{i-1} \times R_i$ matrix with $R_0 = R_L = 1$. As such it is interesting to see what the maximum value of each R_i is. For each R_i we find:

$$R_i \leq \min\left(R_{i-1}I_i, I_iR_{i+1}\right)$$

If we observe a chain where each site can be in d states we will see that a pyramid like structure appears. This can be seen below:

$$\underset{(1\times R_1)}{\begin{bmatrix} \cdot & \cdot & \cdot \end{bmatrix}} \underset{(R_1\times R_1)}{\begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix}} \underset{(R_1\times R_2)}{\begin{bmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix}} \underset{(R_2\times R_2)}{\begin{bmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix}} \cdots$$

$$\cdots \underset{(R_{L-2}\times R_{L-2})}{\begin{bmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix}} \underset{(R_{L-2}\times R_{L-1})}{\begin{bmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix}} \underset{(R_{L-1}\times R_{L-1})}{\begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix}} \underset{(R_{L-1}\times 1)}{\begin{bmatrix} \cdot \\ \cdot \\ \cdot \end{bmatrix}}$$

We note that in the worst case scenario we have $\max\limits_i\{R_i\} = d^{\frac{L}{2}}$ if the chain is even length and $\max\limits_i\{R_i\} = d^{\frac{L-1}{2}}$ if the chain length is odd. As such the size of our largest Λ_i is at worst $d^{\frac{L}{2}} \times d^{\frac{L}{2}}$. This indicates that we still require storage space that scales with d^L. This is not surprising as we have only rewritten the state into matrix product state form. In order to reduce the amount of storage space used we have to make an approximation that is based on the area law.

4.2 Entropy and the Area Law

Entropy

In order to understand the area law we first need to have a notion of entropy. Entropy is a measure of the uncertainty in the system[11]. We define the Von Neumann entropy of a quantum mechanical system as

$$S = -Tr(\rho \log_2(\rho)) \tag{4}$$

with $Tr()$ the trace and ρ the density matrix of the system defined as $\rho = |\Psi\rangle\langle\Psi|$. Since $|\Psi\rangle$ is a d^L column vector containing the coefficients $C_{n_1,\dots,n_L}$, we know that ρ is a $d^L \times d^L$ matrix.

The Von Neumann entropy gives us information about the entanglement of a bipartite system, that is, a system consisting of two parts. If we split our system into two parts A and B then the entropy of entanglement between the subsystems is given by

$$\begin{aligned} S(\rho_A) &= -Tr(\rho_A \log_2(\rho_A)) \\ S(\rho_B) &= -Tr(\rho_B \log_2(\rho_B)) \\ S(\rho_A) &= S(\rho_B) = S \end{aligned}$$

with $\rho_A = Tr_B(\rho)$ and $\rho_B = Tr_A(\rho)$ the reduced density matrices. Tr_A and Tr_B denote partial traces over the eigenstates of system A or B respectively. We apply an Schmidt decomposition to rewrite our state as a sum of orthonormal states of A and B as follows:

$$|\Psi\rangle = \sum_{i=1}^{r} \lambda_i |u_i\rangle_A \otimes |v_i\rangle_B .$$

Here λ_i are the singular values of the Λ matrix that is between part A and B. Together with equation (4), this leads to[11]:

$$S = -\sum_i Tr(|\lambda_i|^2 \log_2(|\lambda_i|^2))$$

From this formula we see that there is a relation between the entropy of the system and the singular values. We note that for a system with low entropy the sum will generally have fewer large terms. If we are in an state with $S = 0$ for example, we require $\lambda_1 = 1$ and $\lambda_i = 0$ for $i > 1$. This tells us that if we work with states that have low entanglement we will have singular values that quickly decrease.

Area law

For a general system, the Von Neumann entropy of the system follows a volume law[12]. This means that if we double the volume of the system we find that the entropy of the system also doubles. This then implies that if we perform calculations with a two times longer chain we will find that there are on average twice as many larger singular values. Under specific conditions we find something different though: if we are near the ground state of a system described by a gapped local Hamiltonian the entropy of the system follows an area law instead. Gapped means that there is a finite energy gap between the ground state and first excited state and local implies that interactions only occur over short distances. These two conditions are both met by the Hamiltonian described in the experiment as all interactions are only nearest neighbour interactions and there is a finite energy gap between all non-degenerate states. So, if we are near the ground state of the experiment we know that the entropy follows an area law.

The reason it is very important for the system to follow an area law is because it means that the entropy is only determined by the area of the boundary of the system. For a 1-D chain this boundary is always the same size meaning that for states that follow an area law $S = constant$. If the system scales at least linearly, then it cannot be approximated efficiently by MPS[12]. Furthermore if we start with a state near the ground state it will remain near the ground state during time evolution. This means that we remain in the stay in a state that follows the area law. meaning that we can restrict our available states from the entire Hilbert space to only area law states. This is depicted in figure 5:

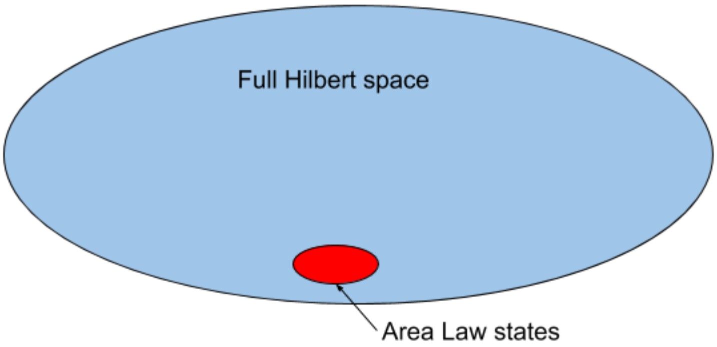


Figure 5: A depiction of the area law states inside of the full Hilbert space.

Truncation

To reduce our available states from the full Hilbert space to states that follow the area law we use an an important empirical observation: for area law states the singular values tend to decrease exponentially fast[13]. Although this is not rigorously proven, it has been proven that states with exponentially decaying singular values follow an area law[14]. If the empirical observation holds then smaller singular values can be neglected in simulations. We introduce a truncation parameter χ. For each Λ at most the first χ singular values kept. The error associated with this truncation can be bounded as follows:

$$|| \, |\Psi\rangle - |\Psi_{Trunc}\rangle \, ||_2^2 \leq 2 \sum_{i=1}^{L} \epsilon_i(\chi).$$

$\epsilon_i(\chi)$ is the truncation error of site i. $||.||_2$ denotes the L^2-norm defined as:

$$||\vec{z}||_2 = \sqrt{|z_1|^2 + |z_2|^2 + ... + |z_n|^2}$$

with $\vec{z}$ an n-dimensional vector.

The following figure shows the first 50 singular values of the ground state of the system and a random state. The chain used is a $5-8-3$ chain and the singular values depicted are those from Λ_8, the center of the chain. j is used as the index of the singular value in the matrix.

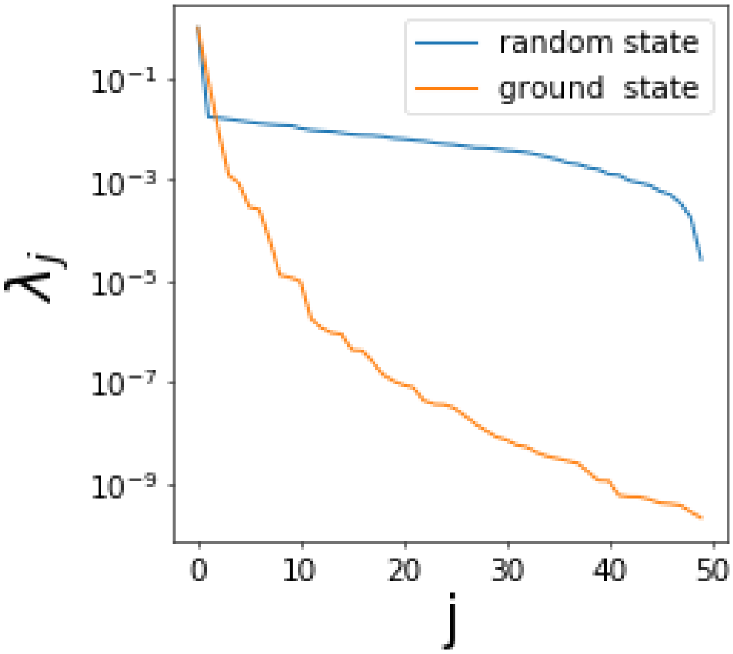


Figure 6: The singular values of the center Λ matrix for a chain of L=16. The ground state and a random state are plotted.

We can see how the singular values of the ground state rapidly decrease whilst the singular values of the random state remain around the same order of magnitude. As such we can assume that for the experimental Hamiltonian the exponential decrease of singular values holds. For $\chi = 15$ we already find that the approximation of the ground state is very accurate, only losing singular values that are of order 10^{-6}. Meanwhile the approximation for the random state remains inaccurate for all first 50 singular values, indicating how this truncation is only effective for states that follow the area law.

4.3 Graphical Representation

One of the benefits of the matrix product state representation is that we can easily represent is graphically. The following figure shows a graphical representation of a given matrix product state.

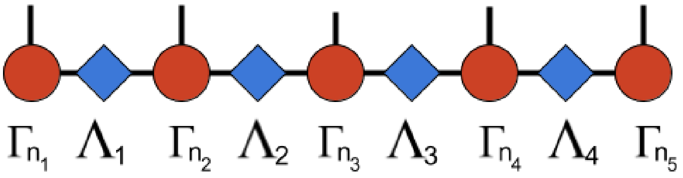


Figure 7: A graphical representation of a matrix product state.

We can see that the Γ matrices are shown in red and the Λ matrices are blue. The indices of each matrix are shown by lines. Each Γ_{n_i} matrix has one line free, this line represents the n_i index. Since the Λ_i are matrices they have two indices they also have two lines, these are both used in matrix multiplication with the adjacent Γ_{n_i} and $\Gamma_{n_{i+1}}$. All Γ_{n_i} matrices save the first and last have three indices as each eigenstate of the site has a matrix associated with it. For the first and last site there is no matrix, there is a row vector and column vector respectively instead.

5 Time Evolution and Computations

In order to simulate how a system behaves over a given period in time we develop a time evolution method that is applicable with MPS and that scales polynomially in time. We furthermore discuss calculations in the MPS formalism such as the dot product.

Time evolution of a quantum wavefunction with a time independent Hamiltonian is described by the following differential equation:

$$i\hbar\frac{d}{dt}\left|\Psi(t)\right\rangle = \hat{H}\left|\Psi(t)\right\rangle$$

This is a system of d^L equations. We find that:

$$\left|\Psi(t+\Delta t\right\rangle = e^{-\frac{i\hat{H}\Delta t}{\hbar}}\left|\Psi(t)\right\rangle . \tag{5}$$

The problem with evaluating the time evolution directly is that it scales exponentially in time for longer chains. As such we have to employ a different method of time evolution. For this purpose we use time-evolving block decimation (TEBD). This algorithm allows for time evolution that scales polynomially with system size. As previously mentioned $\hbar = 1$, meaning that we will remove it from further calculations for clarity.

5.1 Time-Evolving Block Decimation

The idea behind TEBD is to make use of the fact that the Hamiltonian only contains nearest neighbour interactions to split the Hamiltonian into parts[2]. Normally the Hamiltonian of a system with L sites and d eigenstates per site will have dimensions $d^L \times d^L$. This clearly makes direct calculations for long chains impossible. Through splitting up the Hamiltonian we can handle this calculation.

We start off by splitting $\hat{H}$ into two Hamiltonians: $\hat{H}_{odd}$ and $\hat{H}_{even}$. For convenience we will write $\hat{H} = \sum_{i=1}^{L}\hat{K}^{[i]} + \sum_{i=1}^{L-1}\hat{K}^{[i,i+1]}$ as a shorthand. Equation (6) shows how we split $\hat{H}$ for a chain of even length:

$$\hat{H} = \sum_{i=1}^{L}\hat{K}^{[i]} + \sum_{i=1}^{L-1}\hat{K}^{[i,i+1]} = \hat{H}_{odd} + \hat{H}_{even} \tag{6a}$$

$$\hat{H}_{odd} = \frac{1}{2}\sum_{i=2}^{L-1}\hat{K}^{[i]} + \sum_{\substack{i=1\\odd}}^{L-1}\hat{K}^{[i,i+1]} + \hat{K}^{[1]} + \hat{K}^{[L]} \tag{6b}$$

$$\hat{H}_{even} = \frac{1}{2}\sum_{i=2}^{L-1}\hat{K}^{[i]} + \sum_{\substack{i=2\\even}}^{L-1}\hat{K}^{[i,i+1]} \tag{6c}$$

For a chain of odd length $\hat{K}^{[L]}$ will instead be added into $\hat{H}_{even}$. The idea behind the split is to ensure that the nearest neighbour interactions are split between $\hat{H}_{odd}$ and $\hat{H}_{even}$ in alternating fashion while the interactions that only depend on one atom are split symmetrically between $\hat{H}_{odd}$ and $\hat{H}_{even}$. For the end of the chain the complete single site interactions are added to the Hamiltonian that has a nearest neighbour interaction with the site itself. Since $\widehat{H_{odd}}$ always contains $\hat{K}^{[1,2]}$ we always add $\hat{K}^{[1]}$ to it for example.

We can try to visualize the split by observing which elements of each matrix are non-zero. We plot $\hat{H}, \hat{H}_{odd}$, and $\hat{H}_{even}$ for a chain with $L = 4$.

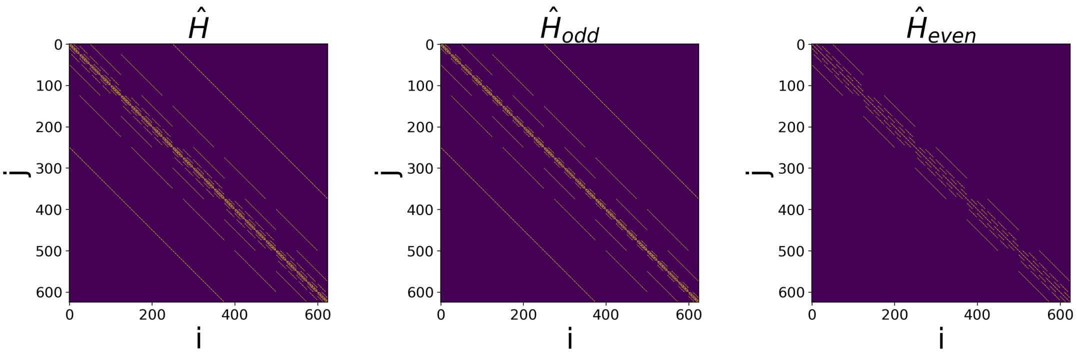


Figure 8: The locations where the three matrices are non-zero. Non-zero components are marked yellow. $L = 4$,$d = 5$.

We clearly see the effect of each nearest neighbour interaction. $\hat{H}_{even}$ contains $\hat{K}^{[2,3]}$ and $\hat{H}_{odd}$ contains $\hat{K}^{[1,2]}$ and $\hat{K}^{[3,4]}$. We recognize this in the patterns, with nearest neighbour interactions in $\hat{H}_{odd}$ being split into a part near the diagonal and a part farther away. Meanwhile interactions of $\hat{H}_{even}$ predominantly occur between this. We also note that in both cases the diagonals are nearly completely non-zero, as a result of single site interactions.

Suzuki-Trotter

We now use $\hat{H}_{odd}$ and $\hat{H}_{even}$ to split the $e^{-i\hat{H}\Delta t}$ in two parts. Because the operators do not commute this will result in an error. This is known as the Suzuki-Trotter 1 decomposition. The approximation looks as follows[15]:

$$e^{(A+B)x} = e^{Ax}e^{Bx} + \mathcal{O}(x^2)$$

We notice that this will result in a global error that is $\mathcal{O}(x)$. Applying the Suzuki-Trotter decomposition yields:

$$e^{-i\hat{H}\Delta t} = e^{-i\hat{H}_{odd}\Delta t}e^{-i\hat{H}_{even}\Delta t}.$$

We now split $\hat{H}_{odd}$ and $\hat{H}_{even}$ even further to obtain the individual nearest neighbour interactions of every adjacent pair together with their single site interactions. We write these as $\hat{O}^{[i,i+1]} = \hat{K}^{[i,i+1]} + \frac{1}{2}(\hat{K}^{[i]} + \hat{K}^{[i+1]})$. For the edges we once again use the full $\hat{K}^{[1]}$ and $\hat{K}^{[L]}$ instead. We write:

$$\hat{H}_{odd} = \sum_{\substack{i=1 \\ odd}}^{L-1} \hat{O}^{[i,i+1]}$$

$$\hat{H}_{even} = \sum_{\substack{i=2 \\ even}}^{L-1} \hat{O}^{[i,i+1]}$$

The benefit of this is that all of the $\hat{O}^{[i,i+1]}$ in $\hat{H}_{odd}$ commute with each other. The same holds for $\hat{H}_{even}$. Therefore, we split both $\hat{H}_{even}$ and $\hat{H}_{odd}$ up into their components and calculate each component individually without further errors. Our final calculation for the time evolution is as follows:

$$e^{-i\hat{H}\Delta t} = \prod_{\substack{i=1 \\ odd}}^{L-1} e^{-i\hat{O}^{[i,i+1]}\Delta t} \prod_{\substack{i=2 \\ even}}^{L-1} e^{-i\hat{O}^{[i,i+1]}\Delta t}$$

For the application of each $O^{[i,i+1]}$ operator onto a given state we note that the operator only depends on the state of particles i and $i+1$. Therefore, we know that the operator will only have effects on Γ_i, Λ_i, and Γ_{i+1}. During the calculation of $O^{[i,i+1]}$ we can thus restrict ourselves to the subspace spanned by these sites. This means we only have to apply a size $d^2 \times d^2$ operator as opposed to a full size $d^L \times d^L$ operator. Through this entire method we have thus been able to eliminate the exponential scaling in the size of the Hamiltonian that occurs. Instead of using one $d^L \times d^L$ matrix we now use L $d^2 \times d^2$ matrices.

This can also be visualized graphically. Because the order in which the operators in $\hat{H}_{odd}$ are applied does not matter, they are placed on the same line. This is also the case for $\hat{H}_{even}$. Figure 9 shows this form of time evolution. For clarity we use $T^{[i,i+1]} = e^{-i\hat{O}^{[i,i+1]}\Delta t}$.

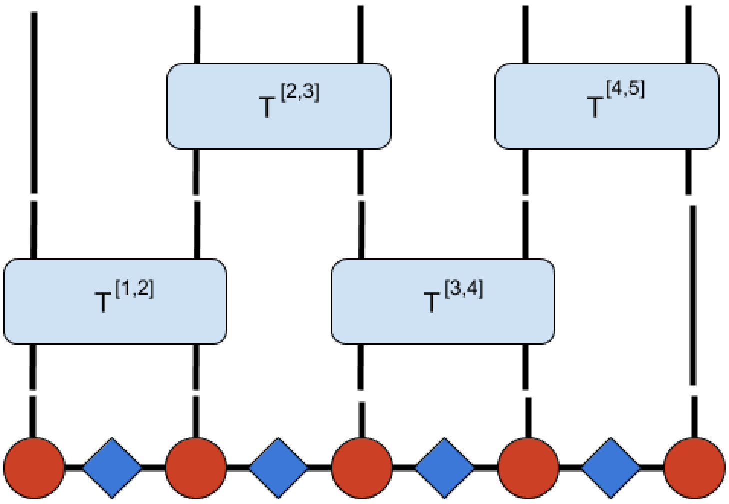


Figure 9: Odd-even time evolution of a matrix product state. $L = 5$

This form of time evolution is aptly named odd-even time evolution. A few decades ago, it was known as a "checkerboard decomposition" where discrete time steps and spin sites together span a square lattice, which is viewed as a checkerboard of which the black fields contain the interactions treated in the calculation.

Crank-Nicolson

In order to accelerate the calculations we approximate the two-site operator $T^{[i,i+1]}$. One property that we require of the approximation of our time evolution operator is that it is unitary. This is required to preserve normalization of the state. The Crank-Nicolson operator is used a well-known discrete form which preserves unitarity[16]:

$$e^{-\frac{i\hat{O}^{[i,i+1]}\Delta t}{\hbar}} = \frac{\mathbb{I} - \frac{i\hat{O}^{[i,i+1]}\Delta t}{2}}{\mathbb{I} + \frac{i\hat{O}^{[i,i+1]}\Delta t}{2}} + \mathcal{O}(\Delta t^3)$$

This yields a global error of $\mathcal{O}(\Delta t^2)$. This also implies that for small Δt, we expect for the error to be dominated by the Suzuki-Trotter decomposition.

Truncation error

Time evolution gradually increases the bond dimension of the Λ matrices in a system[2]. As such we constantly truncate the state down to $\chi \times \chi$. This also means that for smaller values of χ we will obtain a further error every time step.

By combining these errors we see that for large values of χ the global error will scale with $\mathcal{O}(\Delta t)$. In this case Richardson estimation should yield $p = 1$. For small values of χ the global error is highly dependant on the truncation error, this means we will likely find that Richardson estimation yields different values for p.

5.2 Computations in the matrix product state formalism

As of now we have shown how we can reduce the time complexity from exponential in L to polynomial. This section focuses on performing the actual calculations on a matrix product state.

Application of a Single-Site Operator

Let us define the single site operator $\hat{O}_i = \mathbb{I}_1 \otimes \mathbb{I}_2 \otimes ... \otimes \mathbb{I}_{i-1} \otimes \hat{O} \otimes \mathbb{I}_{i+1} \otimes ... \otimes \mathbb{I}_L$. In order to calculate the effect of this single-site operator we need to calculate the full matrix product. Doing this calculation naively would be relatively slow. Because the operator only changes one site we can do this calculation efficiently by first writing it in a simplified form. We split the wavefunction into three parts, the sites to the left of where the operator acts, the site where the operators acts, and the sites to the right of where the operator acts. To do so we perform a Schmidt decompositions. We obtain:

$$\begin{aligned} |\Psi\rangle = &\sum_{n_i} \sum_{\alpha_{i-1},\alpha_i} \Lambda_{i-1}^{\alpha_{i-1},\alpha_{i-1}} \Gamma_{n_i}^{\alpha_{i-1},\alpha_i} \Lambda_i^{\alpha_i,\alpha_i} \left|\eta_{\alpha_{i-1}}\right\rangle_A |n_i\rangle \left|\eta_{\alpha_i}\right\rangle_B \\ &= \sum_{n_i} \sum_{\alpha_{i-1},\alpha_i} \Theta_{n_i}^{\alpha_{i-1},\alpha_i} \left|\eta_{\alpha_{i-1}}\right\rangle_A |n_i\rangle \left|\eta_{\alpha_i}\right\rangle_B \end{aligned} \tag{7a}$$

$$\left|\eta_{\alpha_i}\right\rangle_A = \sum_{n_1,...,n_i} \sum_{\alpha_1,...,\alpha_{i-1}} \Gamma_{n_1}^{\alpha_1} \Lambda_1^{\alpha_1,\alpha_1} \Gamma_{n_2}^{\alpha_1,\alpha_2} \Lambda_2^{\alpha_2,\alpha_2} ... \Gamma_{n_i}^{\alpha_{i-1},\alpha_i} |n_1, n_2, ..., n_i\rangle \tag{7b}$$

$$\left|\eta_{\alpha_i}\right\rangle_B = \sum_{n_{i+1},...,n_L} \sum_{\alpha_{i+1},...,\alpha_L} \Gamma_{n_{i+1}}^{\alpha_i,\alpha_{i+1}} \Lambda_{i+1}^{\alpha_{i+1}\alpha_{i+1}} ... \Gamma_{n_L}^{\alpha_{L-1},\alpha_L} |n_i, n_{i+1}, ..., n_L\rangle . \tag{7c}$$

We now update the middle part by using the following unit operator: $\mathbb{I} = \sum_{n_i'} |n_i'\rangle \langle n_i'|$ and let $\hat{O}_i$ act on it.

$$\begin{aligned} \hat{O}_i |\Psi\rangle &= \sum_{n_i} \sum_{\alpha_{i-1},\alpha_i} \sum_{n_i'} \Theta_{n_i}^{\alpha_{i-1},\alpha_i} \langle n_i'| \hat{O}_i |n_i\rangle \left|\eta_{\alpha_{i-1}}\right\rangle_A |n_i'\rangle \left|\eta_{\alpha_i}\right\rangle_B \\ &= \sum_{\alpha_{i-1},\alpha_i} \sum_{n_i'} \tilde{\Theta}_{n_i'}^{\alpha_{i-1},\alpha_i} \left|\eta_{\alpha_{i-1}}\right\rangle_A |n_i'\rangle \left|\eta_{\alpha_i}\right\rangle_B \end{aligned}$$

with:

$$\tilde{\Theta}_{n_i'}^{\alpha_{i-1},\alpha_i} = \sum_{n_i} \Theta_{n_i'}^{\alpha_{i-1},\alpha_i} \langle n_i'| \hat{O}_i |n_i\rangle . \tag{8}$$

We simply have to return to MPS form using a transformation:

$$\Gamma_{n_i} \to \tilde{\Gamma}_{n_i} = (\Lambda_{i-1})^{-1} \tilde{\Theta}_{n_i} (\Lambda_i)^{-1}$$

We see that the application of a single site operator only ends up changing Γ_{n_i} to $\tilde{\Gamma}_{n_i}$.

Application of a Two-Site Operator

The application of a two-site operator $\hat{O}_{i,i+1}$ understandably is very similar to that of a single site operator. We once again split the chain into three pieces[17]:

$$\begin{aligned} |\Psi\rangle = &\sum_{n_i,n_{i+1}} \sum_{\alpha_{i-1},\alpha_i,\alpha_{i+1}} \Lambda_{i-1}^{\alpha_{i-1},\alpha_{i-1}} \Gamma_{n_i}^{\alpha_{i-1},\alpha_i} \Lambda_i^{\alpha_i,\alpha_i} \Gamma_{n_{i+1}}^{\alpha_i,\alpha_{i+1}} \Lambda_{i+1}^{\alpha_{i+1},\alpha_{i+1}} \left|\eta_{\alpha_{i-1}}\right\rangle_A |n_i, n_{i+1}\rangle \left|\eta_{\alpha_{i+1}}\right\rangle_B \\ &= \sum_{n_i,n_{i+1}} \sum_{\alpha_{i-1},\alpha_{i+1}} \Theta_{n_i,n_{i+1}}^{\alpha_{i-1},\alpha_{i+1}} \left|\eta_{\alpha_{i-1}}\right\rangle_A |n_i, n_{i+1}\rangle \left|\eta_{\alpha_{i+1}}\right\rangle_B , \end{aligned} \tag{9}$$

followed by entering the unit operator $\mathbb{I} = \sum_{n_i',n_{i+1}'} \left|n_i', n_{i+1}'\right\rangle \left\langle n_i', n_{i+1}'\right|$ and applying the operator $\hat{O}_{i,i+1}$.

$$\begin{aligned} \hat{O}_{i,i+1} |\Psi\rangle = &\sum_{n_i,n_{i+1}} \sum_{\alpha_{i-1},\alpha_{i+1}} \sum_{n_i',n_{i+1}'} \Theta_{n_i,n_{i+1}}^{\alpha_{i-1},\alpha_{i+1}} \left\langle n_i', n_{i+1}'\right| \hat{O}_{i,i+1} |n_i, n_{i+1}\rangle \left|\eta_{\alpha_{i-1}}\right\rangle_A \left|n_i', n_{i+1}'\right\rangle \left|\eta_{\alpha_{i+1}}\right\rangle_B \\ = &\sum_{\alpha_{i-1},\alpha_{i+1}} \sum_{n_i',n_{i+1}'} \tilde{\Theta}_{n_i',n_{i+1}'}^{\alpha_{i-1},\alpha_{i+1}} \left|\eta_{\alpha_{i-1}}\right\rangle_A \left|n_i', n_{i+1}'\right\rangle \left|\eta_{\alpha_{i+1}}\right\rangle_B . \end{aligned}$$

Here we introduced:

$$\tilde{\Theta}^{\alpha_{i-1},\alpha_{i+1}}_{n'_i,n'_{i+1}} = \sum_{n_i,n_{i+1}} \Theta^{\alpha_{i-1},\alpha_{i+1}}_{n_i,n_{i+1}} \left\langle n'_i, n'_{i+1}\right| \hat{O}_{i,i+1} \left|n_i, n_{i+1}\right\rangle \tag{10}$$

analogously to the single-site operator.

The main difference between the application of a two-site operator and a single-site operator is that we have to perform a SVD before we can return to MPS form. As such we reshape $\tilde{\Theta}$ to a matrix using canonical matricization and perform a SVD.

$$\tilde{\Theta}^{\alpha_{i-1},\alpha_{i+1}}_{n'_i,n'_{i+1}} \rightarrow \tilde{\Theta}_{(n'_i\alpha_{i-1}),(n'_{i+1}\alpha_{i+1})} \rightarrow U_{(n'_i\alpha_{i-1}),\alpha'_i}\tilde{\Lambda}^{\alpha'_i,\alpha'_i}_i V_{\alpha'_i,(n'_{i+1}\alpha_{i+1})}$$

We transform U and V into third order tensors and take the horizontal slices analogously to the derivation of MPS. This yields us the U_{n_i} matrices and the $V_{n_{i+1}}$ matrices. We truncate these resulting matrices and $\tilde{\Lambda}_i$ to be $\chi \times \chi$ if required. Λ_i is updated to $\tilde{\Lambda}_i$. The updated Γ_{n_i} and $\Gamma_{n_{i+1}}$ matrices are obtained as follows:

$$\tilde{\Gamma}_{n_i} = (\Lambda_{i-1})^{-1}U_{n_i}$$
$$\tilde{\Gamma}_{n_{i+1}} = V_{n_{i+1}}(\Lambda_{i+1})^{-1}$$

Figure 10 depicts this process:

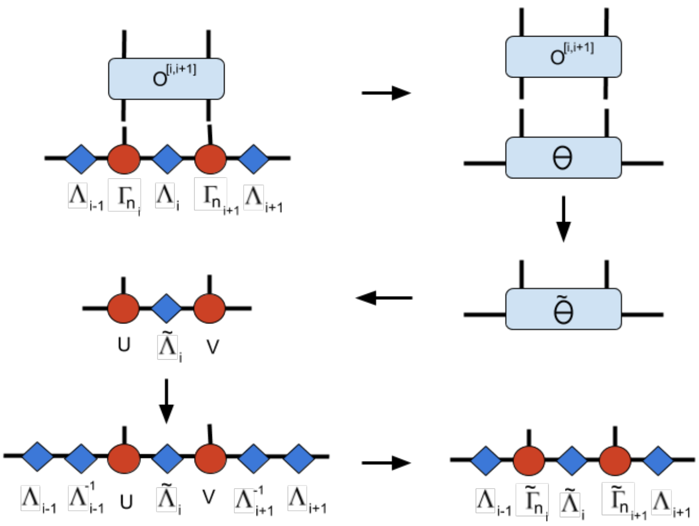


Figure 10: The application of a two-site operator to sites i and i+1.

Expectation of a Single-site and Two-site Operator

To find the expectation of a single-site operator we are required to calculate $\langle\Psi|\,\hat{O}_i\,|\Psi\rangle$. To do so we utilize Θ and $\tilde{\Theta}$ from equations (7a) and (8). The expectation is:

$$\langle\Psi|\,\hat{O}_i\,|\Psi\rangle = \sum_{\alpha_{i-1},\alpha_i}\sum_{n_i}(\Theta^{\alpha_{i-1},\alpha_i}_{n'_i})^\dagger\tilde{\Theta}^{\alpha_{i-1},\alpha_i}_{n'_i}$$

Likewise we find the expectation of a two-site operator using the Θ and $\tilde{\Theta}$ from equations (9) and (10).

$$\langle\Psi|\,\hat{O}_{i,i+1}\,|\Psi\rangle = \sum_{\alpha_{i-1},\alpha_{i+1}}\sum_{n_i,n_{i+1}}(\Theta^{\alpha_{i-1},\alpha_{i+1}}_{n'_i,n'_{i+1}})^\dagger\tilde{\Theta}^{\alpha_{i-1},\alpha_{i+1}}_{n'_i,n_{i+1}}$$

The graphical representation of these calculations is visible in figure 11.

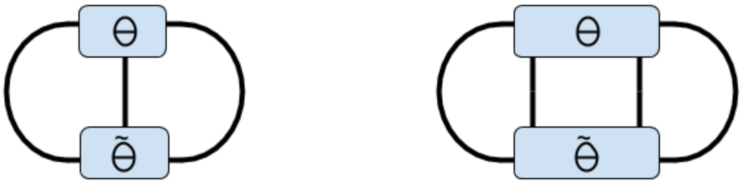


Figure 11: The expectation of a single-site operator and a two-site operator.

Dot product

In order to verify that a given state is normalized we require a fast method to calculate a dot product. We will look at the general calculation of $\langle\Psi_1|\Psi_2\rangle$ with $|\Psi_1\rangle$ and $|\Psi_2\rangle$ different states.

Although the simplest method would be to perform the calculation in one step, this is not the optimal method as computation costs scale exponentially with system size. We instead use the zip-up method to perform the calculation in polynomial time[2].

The zip-up method consists of starting at the edge of the chain and creating a matrix M_1. We start at the left edge. We denote $|\Psi_i\rangle = \sum_{n_1,...,n_L} \Gamma_{n_1,i}\Lambda_{1,i}\Gamma_{n_2,i}\Lambda_{2,i}...\Gamma_{n_L,i}\,|n_1,n_2,...,n_L\rangle_i$.

$$M_1^{\alpha_1',\alpha} = \sum_{n_1}(\Gamma^{\alpha_1'}_{n_1,1})^\dagger\Gamma^{\alpha_1}_{n_1,2}(\Lambda^{\alpha_1',\alpha_1'}_{1,1})^\dagger\Lambda^{\alpha_1,\alpha_1}_{1,2}.$$

We then create M_2 by adding site 2 to the product:

$$M_2^{\alpha_1',\alpha} = \sum_{n_2}\sum_{\alpha_1',\alpha_1} M_1^{\alpha_1',\alpha}(\Gamma^{\alpha_1'\alpha_2'}_{n_2,1})^\dagger\Gamma^{\alpha_1\alpha_2}_{n_2,2}(\Lambda^{\alpha_2',\alpha_2'}_{2,1})^\dagger\Lambda^{\alpha_2,\alpha_2}_{2,2}.$$

This process repeats until we reach the end of the chain. Figure 12 shows the difference between the methods.

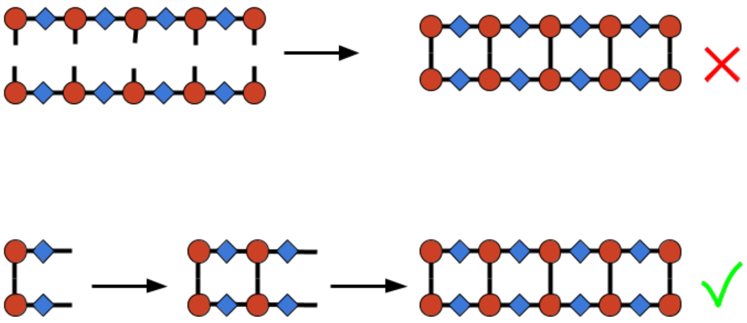


Figure 12: The naive method of calculating the dot product compared to the more efficient zipping method.

6 Data Analysis

In this chapter we analyze the results of simulations using MPS. We start with verifying that the global error is $\mathcal{O}(\Delta t)$, followed by a comparison between direct simulation for a short chain and MPS. We will analyse the energy of the system and the expectation of the operator $\hat{S}_z$ to get a better understanding of the systems behaviour.

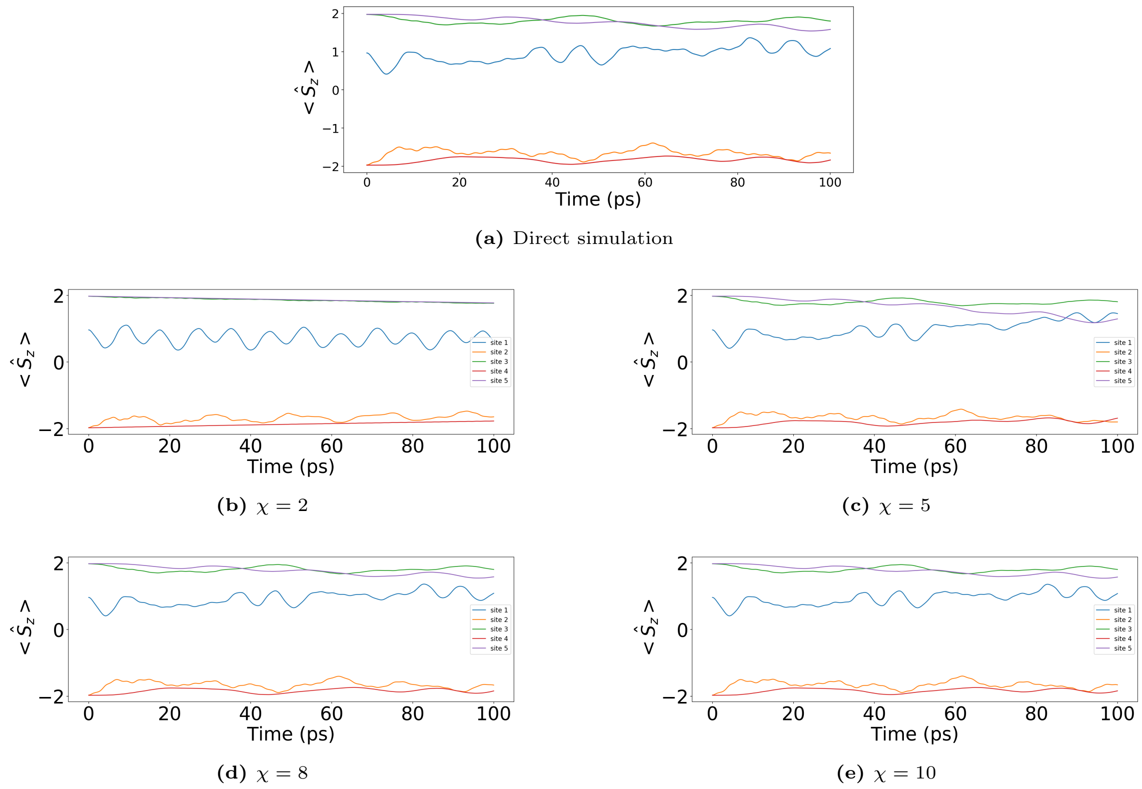
The code for the MPS simulations can be found on the gitlab page in appendix A.

6.1 Richardson and comparison

The error we obtain through Richardson error estimation is expected to be of order 1 for large enough χ. If the truncation error can be neglected then the primary source of error will be due to the Suzuki-Trotter expansion, which has global error $\mathcal{O}(\Delta t)$. We retrieve this result. If χ is chosen to be relatively small however, this result is not retrieved, most likely due to the truncation. In this case $p \approx 0.4$.

For the comparison between direct simulations and simulations using MPS we use the simulations done by the experimentalists themselves. Because direct simulations are only possible for short chains, we simulate the first 5 atoms of a $5-8-3$ chain. Unless explicitly stated we consider the input chain to have length 5. The effects of the output and reset chain are approximated by a constant coupling. Although this is not necessarily realistic, it allows for verification that the MPS method obtains correct results.

Figure 13 compares the expectation of $\hat{S}_z$ for a direct simulation and 4 MPS simulations. In these plots an excitation $\hat{S}_-$ is applied to atom 1, the left-most atom. The 4 MPS simulations have been chosen to highlight the rate of converge of the method.



(a) Direct simulation

(b) $\chi = 2$

(c) $\chi = 5$

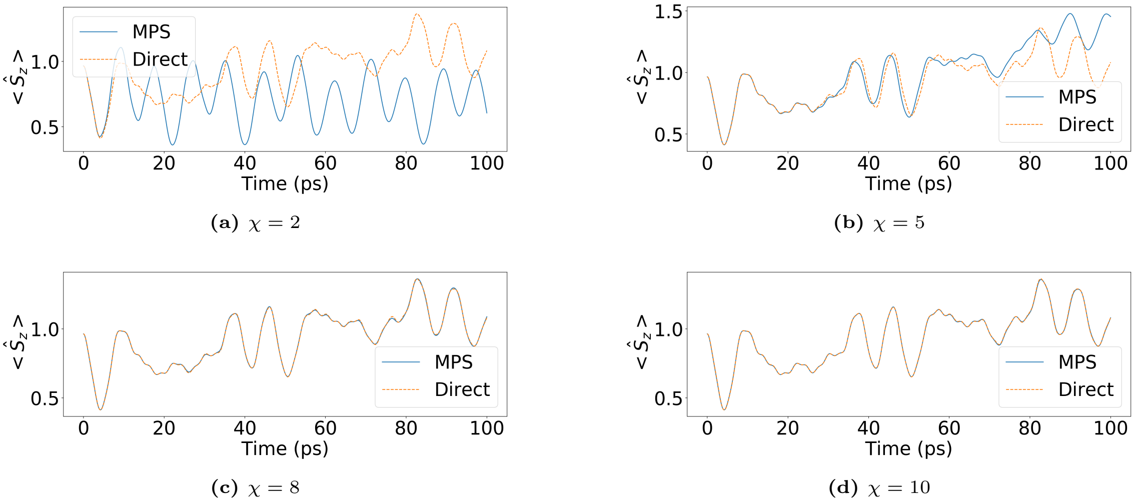
(d) $\chi = 8$

(e) $\chi = 10$

Figure 13: Matrix product state simulations for a variety of χ values. These simulations were completed with T=2500 timesteps

We notice that increasing χ rapidly yields to better convergence. For $\chi = 2$ we recover practically nothing from the direct simulation whilst $\chi = 8$ already yields nearly indistinguishable results if plotted as above. This indicates that the MPS formalism is very effective for simulating this chain. It also verifies that the excitation does not cause us to shift into a state that does not follow an area law. This is important since we can only simulate area law states.

Below are four plots in which we focus on the measurement of the leftmost particle. In each plot we compare the results from the direct simulation with that of the MPS simulation.



(a) $\chi = 2$

(b) $\chi = 5$

(c) $\chi = 8$

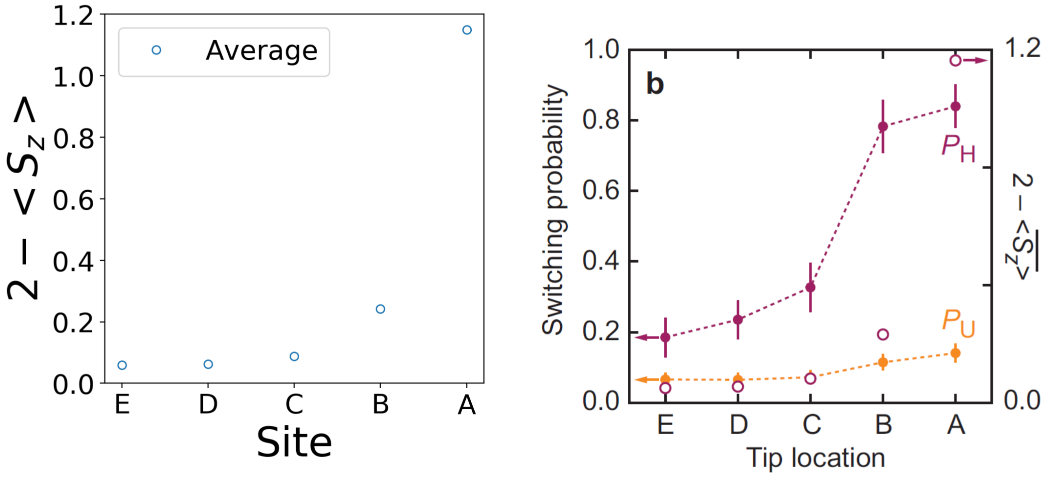
(d) $\chi = 10$

Figure 14: Comparison of expectation of $\hat{S}_z$ of site 1 from MPS simulation to direct simulation for varying χ. These simulations were completed with T=2500 timesteps.

Figure 14 gives us a clearer picture of how quickly the MPS simulations converge. Once again $\chi = 2$ gives highly inaccurate results, but we can now also easily note that the results for $\chi = 5$ diverge relatively quickly. For values as low as $\chi = 8$ we find that the results are highly accurate, only for a few points can we see the divergence. The results for $\chi = 10$ are nearly indistinguishable from the direct simulations, showcasing how MPS simulations can achieve high fidelity with relatively few singular values.

Switching probabilities

We can use these results to determine which excitations are likely to cause the chain to flip. For this purpose we time average $\langle \hat{S}_z \rangle$ for the atom nearest to the output for the first 20 ps and plot 2 minus the result. We compare MPS simulation to direct simulation.



(a) MPS Simulation

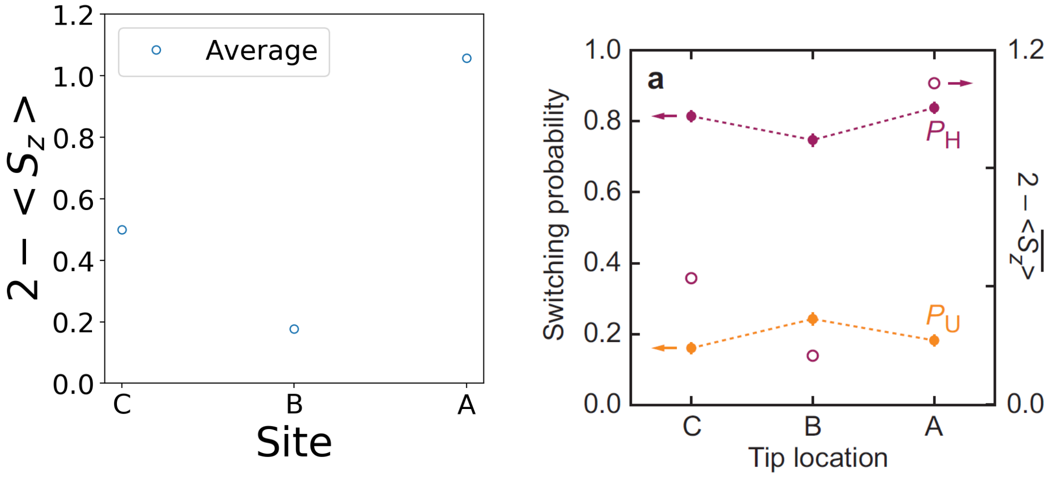
(b) Experimental results and direct calculation. Rings: Direct simulation results. Red dots: switching probability for happy input. Yellow dots: switching probability for unhappy input.

Figure 15: The effect of applying an excitation at the given atom on the atom nearest to the output. The sites are labeled depending on how far they are from the output.

In figure 15b we see two dotted lines that connect points and a number of empty rings. The rings are results from a direct simulation and the points are experimental results. The red points show the switching probability if an excitation is applied to a happy input state and the yellow points show the switching probability if an excitation is applied to an unhappy input state.

The sites are labeled depending on how far away they are from the output chain, with A being the nearest and E being the farthest. It is not surprising that an excitation of atom A yields a significant chance for the expectation of atom A to drop. We see that the effect of the excitations decreases as they move further from the output chain. This implies that we also find a lower switching probability from excitations further from the output chain. This is in line with experimental results.

We can also create the same plot with an input chain of length 3 instead. The result of doing so is shown below:



(a) MPS Simulation

(b) Experimental results and direct calculation. Rings: Direct simulation results. Red dots: switching probability for happy input. Yellow dots: switching probability for unhappy input.

Figure 16: The effect of applying an excitation at the given atom on the atom nearest to the output. The sites are labeled depending on how far they are from the output. $L = 3$

We notice that the behaviour is different from an input chain of length 5. Instead of a gradual decrease we find that an excitation on atom C actually has more effect on atom A than an excitation on atom B. This is also in line with experimental results.

6.2 Analysis of full chain

The main benefit of the MPS formalism is of course that we can simulate long chains. Whereas we previously only simulated a chain of length 5 where the output chain and reset chain were approximated by a constant coupling, we will now perform the calculations with a full chain.

We start of by approximating the ground state of the system. To do so we utilize imaginary time evolution. If we replace t by $-i\tau$ in equation (5) we find that each eigenstate $|\Psi_n\rangle$ inside of our full state is multiplied by $e^{-E_n\tau}$. Thus we immediately note that the coefficient of the eigenstate with the lowest energy decreases the slowest or increases the fastest depending on the sign of the energy. Therefore, we can iterate to the ground state of the system using imaginary time evolution and normalization. This is important because the real experiment starts with a system in the ground state. we have to ensure that we iterate to the ground state before applying an excitation.

The figure below shows how the energy in the chain decreases as we move towards the ground state. The behaviour is expected to be close to exponential, only deviating from exponential due to renormalization of the state.

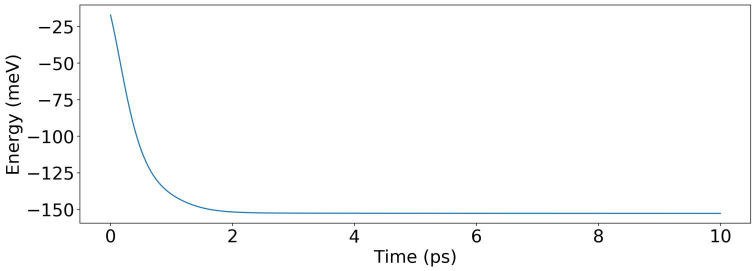


Figure 17: Energy of the chain throughout imaginary time evolution to the ground state. The Initial state is a random state.

In this case we start with a state that is completely random. This means that the initial energy will generally be high and that the state does not follow an area law. Nevertheless we see the expected behaviour of imaginary time evolution: near exponential decay to the ground state.

Since we approximately know what the ground state looks like we can start with a system that is in a state near the ground state. This allows for significantly faster convergence. It also allows us to choose a state that evolves towards the N_A ground state. If we begin with a random state we will end up in a superposition of state N_A and N_B. As the experiment starts in the N_A state, this would result in problems for the simulation. With this in mind, the state that we will perform imaginary time evolution on is the $(2, -2, 2, ...)$ state.

Normalization

In order to ensure that the error related to truncation is not too high we can observe the norm of the state. In a real quantum system this will always be equal to 1 because we require that $|\langle\Psi|\Psi\rangle|^2 = 1$. Depending on our truncation parameter χ we can start to deviate from it. If we want to ensure that our simulations are realistic we require a large enough χ that our normalization remains relatively close to 1.

In the following figure we show how the norm of a state changes during time evolution for several values of χ. The state that we are plotting is a state where an excitation is applied to the first site.

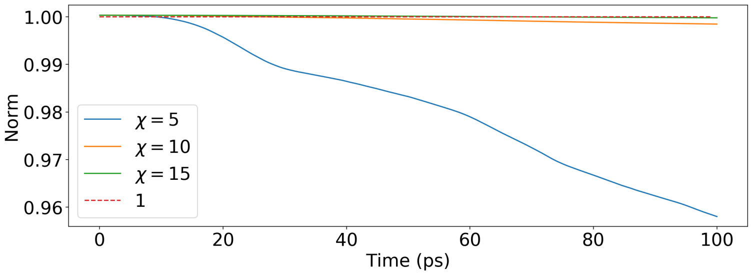


Figure 18: The normalization of a state throughout time evolution for $\chi = 5, 10, 15$.

We see the expected behaviour. For $\chi = 5$ we find that normalization is not maintained while $\chi = 10$ and $\chi = 15$ maintain normalization significantly better. We note that at the start of the simulation the norm of the state is actually larger than 1. This is of no significant concern however as the error is in the order of 10^{-4}. We find that $\chi = 10$ is a reasonable truncation parameter for most calculations. Thus it will be used unless stated otherwise. The last property that we have to verify is that the energy in the chain is conserved throughout real time evolution.

Energy conservation

Because the chain we are simulating is a closed system, conservation of energy requires us to find a constant total energy. It is thus important to ensure that this is approximately true. Due to errors from the Suzuki-Trotter decomposition and the Crank-Nicolson method we will of course slightly oscillate around the true value. Furthermore, truncation of the state will affect the energy. With this in mind we will show the energy in two situations: with and without normalization. This gives us an idea of what the effect of truncation is on the energy of the system while simultaneously allowing us to see if energy is conserved. Figure 19 shows this.

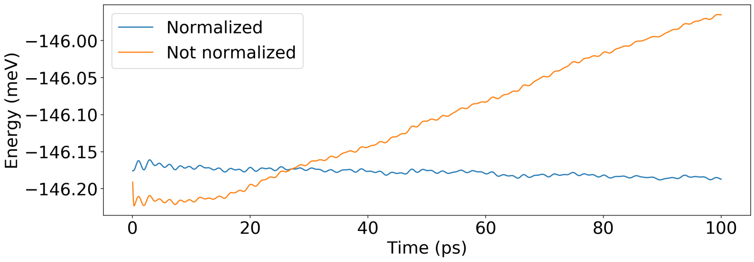


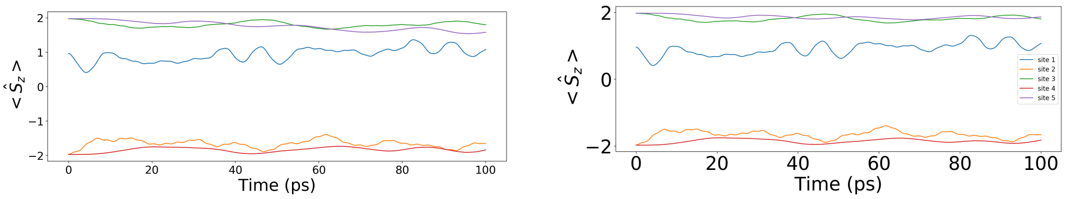
Figure 19: Total energy inside of a 5-8-3 chains after an excitation on site 1.

We see behaviour that should occur. Both of the lines oscillate a bit due to errors as expected. We can clearly see the effect of losing normalization on the orange line, with the energy slowly moving towards zero and the normalization of the state does the same. The normalized state remains a lot closer to constant, with the value slowly becoming more negative. This decrease indicates that the truncation has a tendency to remove higher energy eigenstates. This is unsurprising considering that the area law holds for low energy states.

We can also use figures 17 and 19 to find the energy change that is related to an excitation of site 1. We find $\Delta E = 6.55\ meV$, once again confirming that an excitation on site 1 does not increase the energy significantly. As such the area law holds for this excited state, as was previously seen from the analysis of the normalization of the state.

Excitation on particle 1

The first simulation of the experiment we want to perform is an excitation of particle 1. This allows us to see how accurate the direct calculation with the approximation of constant coupling was.



(a) Direct

(b) Full 5-8-3 chain with MPS

Figure 20: Comparison between direct simulation of the input chain and MPS simulation of the full chain.

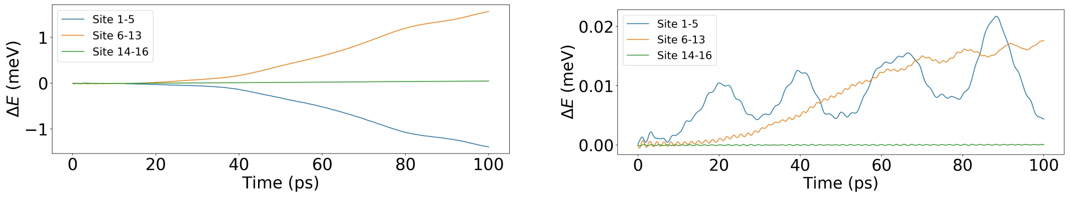
We see that up to 40 *ps* the behaviour is nearly identical. beyond this time we observe that the measurements for site 5, the site closest to the output, are starting to differ. This is in line with what we would expect. The error from assuming a constant coupling between the input and output should primarily be noticeable at site 5. We also see the differences for site 4, however these are already quite subtle. We furthermore find that the behaviour of the rest of the chain is modelled very well by the direct simulation.

Energy transfer

One result that the experimentalists could not easily explain was related to the switching probability of a system with the input chain in the unhappy state. If the system is initially in the happy state, then they expected for an excitation to be able to move the system to an unhappy state. for a system where the input chain is in an unhappy state, their expectation was that the system would not switch after an excitation as switching would not be energetically favourable. They did however find that there was a switching probability.

In order to try to explain what occurs we can look at the energy transfer through the chain. It could be that part of the excitation moves from the input chain to the output, possibly affecting the state of the output. If this is the case we would expect for there to be an increase in energy in the output chain. If this does not occur then we can rule out energy tunneling into the output as a major contribution to the switching probability.

We will observe the energy in each of the three parts of the chain; the input, the output, and the reset. In doing so we only consider the single-site operators and the two-site operators that act between two sites of the same part. The energy in the coupling between parts is neglected. This means that the total energy that is plotted is not conserved. Furthermore, we only plot the relative difference in the energy of the chain between the start of the simulation and the given point in time. This means that the energy of each part of the chain starts at zero, allowing us to see the transfer of energy more clearly. We perform the calculation for both the situation where the input chain is happy and where the input chain is unhappy. Because the energy changes are small for the excitation of the unhappy chain, we continuously normalize the state to limit the effects of truncation. We additionally use $\chi = 15$ and decrease the size of the time-steps for calculations on the unhappy chain. Figure 21 shows how energy moves though the 5-8-3 chain.



(a) Happy

(b) Unhappy

Figure 21: How energy moves through a 5-8-3 chain after an excitation on site 1 for the situation where the input is happy and unhappy.

For the excitation in the happy chain it is immediately clear that the energy in the input chain and the energy in the output chain nearly mirror each other. They both behave very similarly as you would expect from energy conservation.

The excitation in the unhappy chain behaves very differently. The energy scales are quite small, so the oscillations are likely due to numerical errors as opposed to the actual behaviour of the chain. We clearly see that the total plotted energy is not conserved, as previously mentioned this could be related to neglecting the coupling between chains for energy calculations.

The energy transfer after 100 ps is approximately equal to 1.5 meV for the happy chain. This is about a fourth of the energy of the full excitation. We thus find that most of the energy from the excitation remains inside of the input chain. If we focus on the first 20 ps we find that less than 0.02 meV has transferred into the output chain. This is a small fraction of the total energy of the excitation, meaning it likely has little effect on the switching rate.

For the unhappy chain we find an even lower amount of energy transfer. Due to the size of the numerical errors it is not realistically possible to obtain a good measure for the amount of energy that has transferred to the output chain. We find that the mean energy transferred to the output chain is 0.0001 meV. This is certainly small enough that we can assume that nearly no energy moves from the input chain to the output chain. This is in line with the results about in the experiment, where an excitation on site 1 results in a low switching rate from unhappy to happy.

Interestingly enough the energy of the reset chain does not change significantly. This appears to imply that the excitation never reaches the reset chain. We can verify this quite easily by comparing the relative energy change of the output chain to that of the first half of the output chain. We do this for the situation where the input is happy. This can be seen in the following figure:

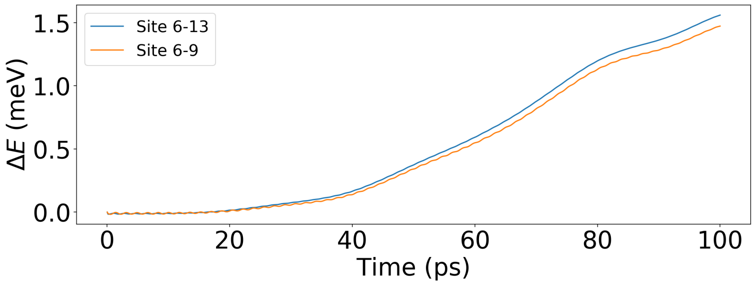
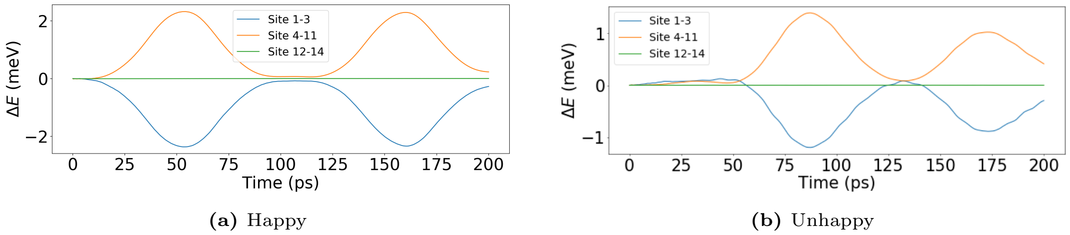


Figure 22: The relative energy change of the output chain compared to that of the first half of the output chain.

There is only a very slight amount of energy that ends up in the right half of the output chain. This is a sign that the reset chain has relatively little effect on the result of the excitation. For the sake of the experiment this is very important to verify, as the point of the reset chain is to reset the system.

We can do a similar analysis for a 3-8-3 chain. In the case of a 3-8-3 chain the excitation results in a very different looking transfer of energy through the chain. Figure 23 shows how energy transfers through a 3-8-3 chain. We plot this for 200 ps instead of 100 ps:



(a) Happy

(b) Unhappy

Figure 23: How energy moves through a 3-8-3 chain after an excitation on site 1.

The energy appears to transfer from the input to the output and back nearly periodically. This goes against expectations. We would expect for the excitation to travel through the full chain and affect the reset chain, but instead it appears to simply return from the output chain to the input chain. The period of this oscillation is approximately 100 ps.

There furthermore appears to be a gradual decrease in the energy in the input chain and a gradual increase in energy of the output chain. This tells us that outside of the periodic behaviour there could also be additional transfer of energy. It may however also be part of a significantly slower oscillation. Further analysis regarding the shape of the oscillation is beyond the scope of this project.

We find that for a 3-8-3 chain with happy input the energy transfer peaks at around 2 meV, about a third of the excitation. After 20 ps approximately 0.24 meV has transferred to the output chain. This is approximately 3.7% of the energy of the excitation. This is of a similar order of magnitude as the energy of the coupling, indicating that it could very well have a notable effect.

The unhappy chain can also provide us with some information. For the first 20 ps the mean energy transferred is equal to 0.012 meV. This is a reasonable amount more than for an unhappy 5-8-3 chain. That being said, it is still an order of magnitude smaller than the energy in the coupling between sites. It is unlikely that this effect alone would explain the unexpectedly high switching rate from unhappy to happy. To understand if this energy transfer has a significant effect, further analysis would have to be done.

7 Conclusions

Our results show that the MPS formalism is a very powerful method for simulating long chains of quantum particles that would otherwise be impossible to analyse numerically. It allows for relatively fast and accurate simulations of systems near the ground state using a truncation parameter as low as $\chi = 10$ for the spin-2 chains analysed. We found that normalization is nearly completely maintained for high enough χ, losing only about 0.2% of the state after 100 *ps* for χ=10. Energy is furthermore nearly completely conserved, with losses that appear to be connected to loss of normalization. The method reproduces the results of direct simulation and provides further insights into physical properties such as energy transfer through the system. Although the current algorithm is limited to one dimensional systems near the ground state, we consider MPS highly advantageous to direct simulations where applicable.

In order to further improve MPS simulations one may consider including decoherence into the method. In real-life experiments decoherence damping will always play a role, this is ignored by our current implementation of MPS. These effects may be explored through a variety of methods such as quantum jumps. Especially for long duration simulations the effects of adding damping and decoherence can prove to be greatly beneficial for comparison with experiments, where these cannot be switched off.

Furthermore, an interesting extension is that to two dimensions, which is possible for highly anisotropic interaction or with weak nonlocal interactions. For this, swap operators can be introduced. Using MPS on 2-dimensional systems has already been utilized to obtain the ground state of systems, however time-evolution of 2-dimensional systems using MPS is still relatively unexplored.

A last topic of further research would be to fully analyze the transfer of energy through a chain with the help of MPS. This could allow for a better understanding of how to limit these effects. Unintended energy transfer to the output chain can result in noise, undermining the purpose of the device.

Appendix

Appendix A: The code

All code used for the simulations can be found in the following gitlab repository:
`https://gitlab.com/JustinBGIT/justin-bep-matrix-product-states/-/tree/master`
The code allows for real time evolution and imaginary time evolution for a chain of arbitrary size.

Bibliography

[1] R.J.G. Elbertse, D. Coffey, J. Gobeil, et al. Remote detection and recording of atomic-scale spin dynamics. *Communications physics*, 2020.

[2] S. Paekal et al. Time-evolution methods for matrix-product states. *Annals of Physics*, 2019.

[3] P. Vree. Computing the Ground-State Energy of the Ising Model in a Transverse Field with Matrix Product States. 2019.

[4] A. Blum, J Hopcroft, and R. Kannan. *Foundations of Data Science.* Cambridge University Press, 2020.

[5] M.M. Wilde. *Quantum Information Theory.* Cambridge University Press, 2013.

[6] Jean Kossaifi. Tensor unfolding.

[7] T.G. Kolda and B.W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, September 2009.

[8] A. Cichocki et al. Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions. *Foundations and Trends in Machine Learning*, 9(4-5):249–429, September 2017.

[9] D.J Griffiths. *Introduction to Quantum Mechanics, Second Edition.* Pearson. Pearson, 2013.

[10] C. Vuik, F.J. Vermolen, M.B. van Gijzen, and M.J. Vuik. *Numerical Methods for Ordinary Differential Equations.* Delft Academic Press, 2016.

[11] J.A. Bengua. Matrix product state decomposition in machine learning and signal processing. 2016.

[12] J. Eisert, M. Cramer, and M. B. Plenio. Colloquium: Area laws for the entanglement entropy. *Rev. Mod. Phys.*, 82:277–306, Feb 2010.

[13] Guifré Vidal. Efficient simulation of one-dimensional quantum many-body systems. *Phys. Rev. Lett.*, 93:040502, Jul 2004.

[14] F.G.S.L. Brandão and M. Horodecki. Exponential decay of correlations implies area law. *Communications in Mathematical Physics*, 333:761–798, 2015.

[15] M. Hatano and N. Suzuki. *Quantum Annealing and Other Optimization Methods.* Springer, 2005.

[16] Zuojing Chen. Efficient modeling techniques for time-dependent quantum system with applications to carbon nanotubes. 2010.

[17] A. Melo. Numerical study of a superconducting qubit for the realization of quantum Ising chains using matrix product state techniques. 2017.