
BACHELORARBEIT
Institut für Theoretische Physik
Computational Physics

2D QUANTUM BOWLING USING
MATRIX PRODUCT STATES

Advisor:
Hans Gerd Evertz, Ao.Univ.-Prof. Dipl.-Phys. Dr.rer.nat.

MICHAEL SCHERBELA

Mat.Nr. 1130975

Sommersemester 2014

Contents

1	Introduction	4
2	The Heisenberg Model	5
3	Motivation to use MPS	6
4	Matrix Product states in 1D	7
4.1	Representation of a single basis state	7
4.2	Representation of an arbitrary linear combination	7
4.3	Left-orthonormalization of Matrix Product State	8
4.4	Right-orthonormalization of Matrix Product State	9
5	Schmidt decomposition	10
6	Pseudoinverse	11
7	Canonical Form of MPS	12
8	Operations on MPS	13
8.1	Application of a one-site operator	14
8.2	Application of two-site operators	14
8.3	Expectation values of one-site operators	16
8.4	Truncation of Schmidt Values	18
8.5	Time development	18
9	Implementation and validation of 1D-Simulation	20
10	Quantum Bowling in 1D	22
11	Extension to 2 dimensions	24
12	Validation of 2D implementation	26
13	Quantum Bowling in 2D	27
13.1	2D Quantum-Bowling at high J_z	27
13.2	Transmission of a single particle through a 2D wall	28
13.3	Transmission rate of a single particle through a 2D wall	29
13.4	Error estimation of transmission rate	31
14	Summary	32

Abstract

Strongly correlated systems are highly interesting, both because of their properties, as well as their computational complexity. Recent advances in computational physics, in particular Matrix Product States (MPS) and the Time Evolving Block Decimation (TEBD) have made accessible the time evolution of these systems, at least in one dimension.

In this work, TEBD is used to simulate the time evolution of the XXZ Heisenberg model in one and two dimensions. In particular, the scattering of impurities at clusters of particles is examined. It is shown that scattering is different in 2D compared to 1D and the dependence of the transmission rate on the coupling energies is simulated.

Kurzfassung

Stark korrelierte Systeme sind von besonderem Interesse, da sie zu neuen Materialeigenschaften, beispielsweise Magnetismus und Supraleitung, führen; die Berechnung der Dynamik dieser Systeme ist jedoch äußerst komplex. Mit dem Formalismus der Matrixproduktzustände sowie der sogenannten Time Evolving Block Decimation (TEBD) wurde in den letzten Jahren ein effizienter Algorithmus zur Simulation der Zeitentwicklung eindimensionaler Systeme gefunden.

In dieser Arbeit wird der TEBD-Algorithmus verwendet, um die Zeitentwicklung des XXZ-Heisenberg Modell in ein und zwei Dimensionen zu simulieren. Insbesondere wird die Streuung von einzelnen Teilchen an gebundenen Clustern von Teilchen betrachtet. Es wird gezeigt, dass im Gegensatz zur eindimensionalen Streuung, bei der einfallende Teilchen vollständig transmittiert werden, bei der zweidimensionalen Betrachtung einfallende Teilchen nur teilweise transmittiert werden. Die Abhängigkeit dieser Transmissionsrate von den beteiligten Wechselwirkungen wird simuliert.

1 Introduction

The microscopic dynamics of atoms, and thus the macroscopic properties of materials, are described by quantum mechanics. Many interesting effects (e.g. BCS superconductivity) do crucially depend on the interaction between particles [1]. Studying these strongly correlated systems can provide valuable insight and might for example eventually help at understanding high temperature superconductors. The systematic investigation of the time evolution of such highly correlated systems has recently become accessible, both experimentally and theoretically.

Experimentally, manipulation and imaging of single ultra cold atoms confined by an optical lattice has been achieved, which allows to systematically analyze the behavior and time evolution of such systems. [2, 3].

On the other hand numerical methods which allow efficient simulations have emerged. It has been shown that at least systems with little entanglement can be efficiently simulated on classical computers [4] using Matrix Product States (MPS), which has led to the development of the Time Evolving Block Decimation (TEBD).

Recently MPS and TEBD have been used to investigate the time evolution of strings of flipped spins on a background of oppositely aligned spins for the 1D Heisenberg model [5]. It has been shown that at strong interactions, these strings form bound states that only decay slowly over time (Fig. 11). Furthermore similar methods have been used to observe the dynamics of a single impurity being scattered on such a string of flipped spins [6].

Fig. 1r shows that on a 1D lattice, an incident particle will be fully transmitted through the wall, resembling the well known Newton Cradle, but the wall will peculiarly shift by 2 sites.

The aim of this work is to further investigate the scattering of impurities on walls, by considering a 2D lattice. It is shown that unlike in the 1D case, a 2D barrier obstructs the movement of an incoming particle depending on the ratio between hopping and bonding energies.

Sec. 2 defines the Heisenberg Model used throughout this work, while Sec. 3 shows the need for an efficient way of simulating it. Sec. 4 - 8 give an introduction into the representation of a system using Matrix Product States and its advantages for simulating the time evolution of such a system. In Sec. 9 - 10 the algorithm for simulating 1D systems (TEBD using MPS) is shown and its implementation validated. Sec. 11-13 finally extend the previous work to 2D and discuss the dynamics of scattered particles in 2D.

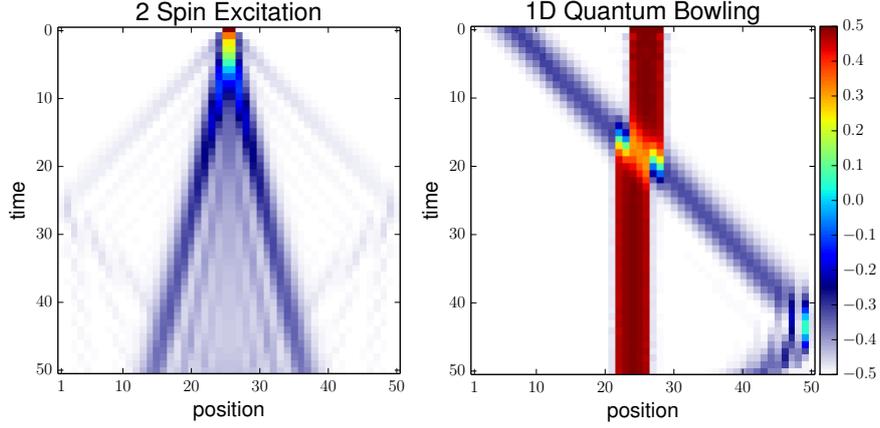


Figure 1: Expectation value of \hat{S}^z for MPS-Simulation of time evolution of flipped spins (impurities) on a lattice of aligned spins.

Left: Flipping 2 neighboring spins leads to 2 distinct signals: A fast propagating, free branch and a slower, bound branch. [5]

Right: "Quantum Bowling" in 1D: An incident particle hits a wall and is fully transmitted as a hole. The wall shifts by 2 sites. [6]

2 The Heisenberg Model

Throughout this work the anisotropic spin $\frac{1}{2}$ Heisenberg Model for 1 and 2 dimensions will be considered. For a one-dimensional chain with closed boundary conditions the Hamiltonian is given by

$$\hat{H} = \sum_{j=1}^{L-1} J^{xy} \left(\hat{S}_j^x \hat{S}_{j+1}^x + \hat{S}_j^y \hat{S}_{j+1}^y \right) + J^z \hat{S}_j^z \hat{S}_{j+1}^z \quad (1)$$

One can define the operators

$$\begin{aligned} \hat{S}_j^+ &:= \hat{S}_j^x + i\hat{S}_j^y \\ \hat{S}_j^- &:= \hat{S}_j^x - i\hat{S}_j^y \end{aligned} \quad (2)$$

which have the property of creating and annihilating $|\uparrow\rangle$ -states: $\hat{S}_j^+ |\downarrow_j\rangle = |\uparrow_j\rangle$, $\hat{S}_j^- |\uparrow_j\rangle = |\downarrow_j\rangle$. Using the equalities 2, one can write the Hamiltonian as:

$$\hat{H} = \sum_{j=1}^{L-1} \frac{J^{xy}}{2} \left(\hat{S}_j^+ \hat{S}_{j+1}^- + \hat{S}_j^- \hat{S}_{j+1}^+ \right) + J^z \hat{S}_j^z \hat{S}_{j+1}^z \quad (3)$$

Eq. 3 reveals an alternative way of looking at the Hamiltonian. The first two terms (with the prefactor J^{xy}) describe a hopping from position j to position $j+1$ and vice versa. The

J^z term describes repulsion/attraction (depending on the sign on J^z) between adjacent spins. Especially when considering a "background" of spins pointing in one direction with few spins pointing in the opposite direction it is convenient to speak of the flipped spins as "particles" that can hop to adjacent sites and form "bonds" with other particles due to the additional energy J^z . Indeed the model as described by Eq. 3 is equivalent to a tight binding system of spinless fermions with nearest neighbor attraction or repulsion by the so called Jordan-Wigner transformation. [8]

The extension of this model to more dimensions is achieved by adding the interactions with neighboring sites along the additional dimension to the Hamiltonian. Instead of 2 next neighbors, every site has now 4 next neighbors to interact with. For a rectangular 2D grid of spins of shape $L_x \times L_y$ the Hamiltonian is

$$\begin{aligned} \hat{H} = \sum_{j=1}^{L_x-1} \sum_{k=1}^{L_y-1} J_{\parallel}^{xy} \left(\hat{S}_{j,k}^x \hat{S}_{j+1,k}^x + \hat{S}_{j,k}^y \hat{S}_{j+1,k}^y \right) + \\ J_{\perp}^{xy} \left(\hat{S}_{j,k}^x \hat{S}_{j,k+1}^x + \hat{S}_{j,k}^y \hat{S}_{j,k+1}^y \right) + \\ J_{\parallel}^z \hat{S}_{j,k}^z \hat{S}_{j+1,k}^z + \\ J_{\perp}^z \hat{S}_{j,k}^z \hat{S}_{j,k+1}^z \end{aligned} \quad (4)$$

As can be seen from Eq. 4, the Hamiltonian has 4 parameters: 2 parameters that determine hopping of particles along the x and y direction of the system (J_{\parallel}^{xy} , J_{\perp}^{xy}) and 2 parameters that specify the bond strength between adjacent particles along the x and y direction (J_{\parallel}^z , J_{\perp}^z). To avoid confusion with the index of orientation of the Spins, the directions x and y are labeled as parallel \parallel and perpendicular \perp .

3 Motivation to use MPS

A single spin can be described by 2 complex numbers, e.g. the coefficients for $|\downarrow\rangle$ and $|\uparrow\rangle$. To describe a system of 2 spins, 4 coefficients are needed, e.g the coefficients for $|\downarrow\downarrow\rangle$, $|\downarrow\uparrow\rangle$, $|\uparrow\downarrow\rangle$ and $|\uparrow\uparrow\rangle$. To fully describe a system of N spins, 2^N coefficients are necessary. This exponential growth in complexity severely limits the size of systems that can be treated using full diagonalization of the system's Hamiltonian.

If the system obeys certain symmetries (e.g. the Hamiltonian in Eq. 1 conserves the total spin of the System in z direction) one can choose only basis states that obey that symmetry. Although choosing a clever basis allow significantly larger systems to be simulated (see Sec. 9), more complex systems are still out of reach using exact diagonalization of the Hamiltonian.

Matrix Product States (MPS), together with Time Evolving Block Decimation (TEBD) on the contrary, allow the simulation of the time evolution of significantly larger systems as long as the entanglement remains small [7].

4 Matrix Product states in 1D

The chapters 4 to 8 outline the mathematical basis of MPS and follow the presentations in [7, 9].

A chain of L spins can be described by 2^L coefficients $c_{s_1 s_2 \dots s_L}$, where each index s_j has two possibilities, corresponding to $|\downarrow\rangle$ or $|\uparrow\rangle$ of the spin at position j . In a Matrix Product State (MPS) each of these coefficients is described as a product of L matrices $M^{[j]s_j}$.

$$c_{s_1 s_2 \dots s_L} = M^{[1]s_1} M^{[2]s_2} \dots M^{[L]s_L} \quad (5)$$

4.1 Representation of a single basis state

When describing a product state (only 1 of all 2^L coefficients is nonzero) these matrices are all of dimension $[1 \times 1]$. As an example, the state $|\psi\rangle = |\downarrow\downarrow\uparrow\uparrow\downarrow\rangle$ can be represented by the following set of $[1 \times 1]$ matrices:

j	1	2	3	4	5
$ \psi\rangle$	\downarrow	\downarrow	\uparrow	\uparrow	\downarrow
$M^{[j]\uparrow}$	0	0	1	1	0
$M^{[j]\downarrow}$	1	1	0	0	1

It is apparent that the only coefficient that is nonzero, is the coefficient of the desired basis state.

4.2 Representation of an arbitrary linear combination

To describe a general state, which is a linear combination of n basis states, diagonal matrices of shape $[n \times n]$ can be used. The first and last matrices have to be of shape $[1 \times n]$ and $[n \times 1]$ respectively, so that the full product will yield a scalar coefficient. As an example, the state $c_1 |\downarrow\uparrow\downarrow\rangle + c_2 |\downarrow\downarrow\uparrow\rangle$ can be represented by the following set of matrices:

j	1	2	3	4
$M^{[j]\uparrow}$	$\begin{pmatrix} 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \end{pmatrix}$
$M^{[j]\downarrow}$	$\begin{pmatrix} c_1 & c_2 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 1 \end{pmatrix}$

Multiplying the matrices, it is obvious that only 2 of the 16 possible coefficients are nonzero. This can be extended to linear combinations of an arbitrary number of basis states by using the following scheme:

$$M_{kk}^{[j]\uparrow} = \begin{cases} 1 & \text{if } s_j \text{ at basis state } k \text{ is } \uparrow \\ 0 & \text{if } s_j \text{ at basis state } k \text{ is } \downarrow \end{cases} \quad (6)$$

and vice versa for $M_{kk}^{[j]\downarrow}$. [7] shows that by using diagonal block matrices instead of diagonal matrices, not only basis states can be added, but also arbitrary states in MPS form. Such a diagonal representation, however, can be very inefficient, which can immediately be seen by considering a linear combination of two identical states:

$$|\psi\rangle = \frac{1}{2} (|\psi\rangle + |\psi\rangle) \quad (7)$$

When storing this state in Eq. 7 as a linear combination, it obviously needs an excessive amount of memory compared to its MPS representation as a product state. In fact the size of the MPS matrices can often be drastically reduced by compressing them as described in Sec. 8.4.

4.3 Left-orthonormalization of Matrix Product State

The example in Section 4.2 shows that an MPS representation is not unique. For example one could shift the coefficients c_1 and c_2 to any other of the matrices, without changing the result after performing all matrix multiplications. It is therefore possible to impose additional constraint on the matrices, with one possible choice being

$$\sum_{s_n} M^{s_n \dagger} M^{s_n} = \mathbb{1} \quad (8)$$

which leads to a *left orthonormal* state [1]. Any MPS can be left orthonormalized by the following algorithm [7]:

1. Starting with an arbitrary MPS, the auxiliary row index and the spin index s_1 of the first matrix are combined to form one index (e.g. by concatenating the matrices

corresponding to different spin indices along the first dimension):

$$\begin{aligned} c_{s_1 s_2 \dots s_L} &= \sum_{\{\alpha\}} M_{1,\alpha_1}^{[1]s_1} M_{\alpha_1,\alpha_2}^{[2]s_2} \dots M_{\alpha_{L-1},1}^{[L]s_L} \\ c_{s_1 s_2 \dots s_L} &= \sum_{\{\alpha\}} \Theta_{(1,s_1),\alpha_1}^{[1]} M_{\alpha_1,\alpha_2}^{[2]s_2} \dots M_{\alpha_{L-1},1}^{[L]s_L} \end{aligned} \quad (9)$$

2. Then a singular value decomposition (SVD) of the matrix Θ is performed resulting in

$$\Theta_{(1,s_1),\alpha_1}^{[1]} = \sum_{\beta} U_{(1,s_1),\beta} S_{\beta\beta} V_{\beta\alpha_1}^\dagger \quad (10)$$

where U and V are unitary matrices and S is a diagonal matrix containing the singular values. [10]

3. The matrix U is now split back into individual matrices A (one for each value of s_1). The remaining matrix product SV^\dagger is multiplied from the left onto the next matrix $M^{[2]}$ which yields:

$$c_{s_1 s_2 \dots s_L} = \sum_{\{\alpha\}\beta} A_{1,\beta}^{[1]s_1} \widetilde{M}_{\beta,\alpha_2}^{[2]s_2} \dots M_{\alpha_{L-1},1}^{[L]s_L} \quad (11)$$

As U is unitary it holds that

$$\begin{aligned} U^\dagger U &= \mathbb{1} \\ \sum_{\alpha_n, s_n} U_{\beta, (\alpha_{n-1}, s_n)}^* U_{(\alpha_{n-1}, s_n), \beta'} &= \delta_{\beta, \beta'} \\ \sum_{\alpha_n, s_n} A_{\alpha_{n-1}, \beta}^{[n]s_n*} A_{\alpha_{n-1}, \beta'}^{[n]s_n} &= \delta_{\beta, \beta'} \\ \sum_{s_n} A^{[n]s_n^\dagger} A^{[n]s_n} &= \mathbb{1} \end{aligned} \quad (12)$$

which is the the property required by Equation (8).

This procedure is now repeated from step 1 with the second Matrix \widetilde{M} , up to the last matrix in the chain. When performing the SVD on the last matrix, the product SV^\dagger will be a scalar containing the norm of the state [7]. Upon dropping it, the state is then left orthonormalized.

4.4 Right-orthonormalization of Matrix Product State

Similarly to Eq. 8 one could impose the relation

$$\sum_{s_n} M^{s_n} M^{s_n^\dagger} = \mathbb{1} \quad (13)$$

which leads to a *right orthonormalized* state. To right orthonormalize an arbitrary MPS an algorithm similar to Sec. 4.4 is used.

1. Starting from the right end of the MPS, the auxiliary column index and the spin index are combined to form one index (e.g. by concatenating the matrices corresponding to different spin indices along the second dimension):

$$\begin{aligned} c_{s_1 s_2 \dots s_L} &= \sum_{\{\alpha\}} M_{1,\alpha_1}^{[1]s_1} M_{\alpha_1,\alpha_2}^{[2]s_2} \dots M_{\alpha_{L-1},1}^{[L]s_L} \\ c_{s_1 s_2 \dots s_L} &= \sum_{\{\alpha\}} M_{1,\alpha_1}^{[1]s_1} M_{\alpha_1,\alpha_2}^{[2]s_2} \dots \Theta_{\alpha_{L-1},(1,s_L)}^{[L]} \end{aligned} \quad (14)$$

2. Similar to Sec. 4.3 the resulting matrix Θ is now SV-decomposed: $\Theta = USV^\dagger$
3. The Matrix US is multiplied from the right onto the Matrix $M^{[L-1]}$. After splitting the column index of V^\dagger one obtains the right orthonormalized matrix B .

$$B_{\beta,\alpha_n}^{[L]s_L} = V_{\beta,(\alpha_n,s_L)}^\dagger \quad (15)$$

Iterating this procedure from right to left along the chain produces a fully right orthonormalized MPS.

5 Schmidt decomposition

Let $|\psi\rangle$ be an element of the Hilbert space S , which we divide arbitrarily into two subspaces A and B so that $S = A \otimes B$. The state $|\psi\rangle$ can be represented by

$$\begin{aligned} |\psi\rangle &= \sum_{\{S\}} c_S |S\rangle \\ |\psi\rangle &= \sum_{\{A\}} \sum_{\{B\}} M_{AB} |A\rangle |B\rangle \end{aligned} \quad (16)$$

The coefficient M_{AB} can be interpreted as a matrix element. SV decomposing $M = U\lambda V^\dagger$ and inserting it into 16 leads to

$$\begin{aligned} |\psi\rangle &= \sum_{\{A\}} \sum_{\{B\}} \sum_{\gamma} U_{A,\gamma} \lambda_{\gamma} V_{\gamma,B}^\dagger |A\rangle |B\rangle \\ |\psi\rangle &= \sum_{\gamma} \lambda_{\gamma} \left(\sum_{\{A\}} U_{\{A\},\gamma} |A\rangle \right) \left(\sum_{\{B\}} V_{B,\gamma}^* |B\rangle \right) \end{aligned} \quad (17)$$

Since U and V are unitary, due to the general properties of the SVD [10], the two braced expressions in Eq. 17 are basis transformations to a new basis

$$\begin{aligned}
|\alpha\rangle_\gamma &:= \sum_{\{A\}} U_{A,\gamma} |A\rangle \\
|\beta\rangle_\gamma &:= \sum_{\{B\}} V_{B,\gamma}^* |B\rangle \\
|\psi\rangle &= \sum_\gamma \lambda_\gamma |\alpha\rangle_\gamma |\beta\rangle_\gamma
\end{aligned} \tag{18}$$

The new basis functions are called *Schmid states* for the given partition of S into A and B and the λ_γ are the corresponding Schmidt Values. Since they were obtained as singular values, they are real valued [10]. If the original state was normalized, it holds that

$$\begin{aligned}
\langle\psi|\psi\rangle &= 1 \\
\sum_{\gamma\gamma'} \lambda_\gamma^2 \langle\alpha_\gamma|\alpha_{\gamma'}\rangle \langle\beta_\gamma|\beta_{\gamma'}\rangle &= 1 \\
\sum_{\gamma\gamma'} \lambda_\gamma^2 \delta_{\gamma\gamma'} \delta_{\gamma\gamma'} &= 1 \\
\sum_\gamma \lambda_\gamma^2 &= 1
\end{aligned} \tag{19}$$

6 Pseudoinverse

The inverse of a matrix M is defined by $MM^{-1} = \mathbb{1}$. Using the SVD, the inverse of M can be calculated by

$$M^{-1} = (U\lambda V^\dagger)^{-1} = V\lambda^{-1}U^\dagger \tag{20}$$

If $\lambda_{jj} = 0$, the corresponding matrix element $\lambda_{jj}^{-1} = \frac{1}{\lambda_{jj}}$ is not defined. However, if $\lambda_{jj} = 0$ this means that the matrix M does not act upon that subspace, which can be excluded from the invsere matrix [9]. This leads to the following definition of λ^{-1} :

$$\lambda_{jj}^{-1} = \begin{cases} \frac{1}{\lambda_{jj}} & \lambda_{jj} \neq 0 \\ 0 & \lambda_{jj} = 0 \end{cases} \tag{21}$$

The matrix MM^{-1} is now the identity for all directions M acts upon, and zero for all others. In any numerical implementation, the decision in Eq. 21 will be taken by comparing λ_{jj} against some small constant ε (e.g. $\varepsilon = 10^{-15}$).

7 Canonical Form of MPS

An extremely useful form of a matrix product state is the so called *canonical form* [7]. In this representation, the matrices for two consecutive sites are linked by a diagonal matrix containing the Schmidt values for the partition of the system at this site (see Fig. 2 for a graphical representation). This representation allows to efficiently evaluate the action of operators that only act locally, as shown in Sec. 8. When having a matrix product state that has been left orthonormalized up to site k and right orthonormalized from the right end up to site $k + 1$, the matrix combining the two sides will contain the Schmidt values for a partition at that bond.

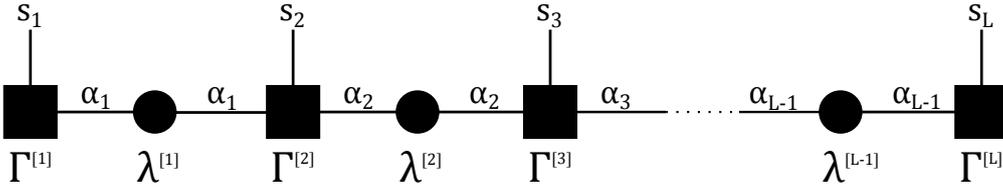


Figure 2: Schematic representation of a canonical MPS. Square matrices represent Γ matrices for each site, circles the diagonal λ matrices containing the Schmidt values. Lines represent indices, a closed line indicates that this index is being summed over.

Every matrix product state can be brought into the canonical form by the following algorithm:

1. Left orthonormalize the state according to Sec. 4.3.
2. Start right orthonormalizing the state according to Sec. 4.4. At every site the Schmidt values (i.e. the matrix λ) of the bond are the singular values obtained by the SVD and are stored.

$$c_{s_1 s_2 \dots s_L} = \sum_{\{\alpha\}} \sum_{\beta} A_{1,\alpha_1}^{[1]s_1} A_{\alpha_1,\alpha_2}^{[2]s_2} \dots A_{\alpha_{L-2},\alpha_{L-1}}^{[L-1]s_{L-1}} U_{\alpha_{L-1},\beta} \lambda_{\beta,\beta} V_{\beta,(1,s_L)}^\dagger \quad (22)$$

3. Like during ordinary right orthonormalization the Matrix $U\lambda$ is multiplied onto the remaining matrices to the left of the chain, forming a new matrix \tilde{A} .

4. To explicitly pull out the matrix λ and obtain Γ , the identity is inserted in the form of $\mathbb{1} = \lambda\lambda^{-1}$ and the inverse is multiplied onto the matrix V^\dagger which becomes \tilde{V}^\dagger . In this operation λ^{-1} is the pseudoinverse of λ as defined in Sec. 6.

$$c_{s_1 s_2 \dots s_L} = \sum_{\{\alpha\}} \sum_{\beta} A_{1,\alpha_1}^{[1]s_1} A_{\alpha_1,\alpha_2}^{[2]s_2} \dots \tilde{A}_{\alpha_{L-2},\beta}^{[L-1]s_{L-1}} \lambda_{\beta,\beta} \left(\lambda_{\beta,\beta}^{-1} V_{\beta,(1,s_L)}^\dagger \right) \quad (23)$$

5. Index splitting of \widetilde{V}^\dagger , yields the Γ matrices for that site.

Iterating this procedure across the entire chain from right to left will transform the MPS into canonical form shown in Eq. 24, where the auxiliary indices were omitted for clarity, and depicted graphically in Fig. 2.

$$c_{s_1 s_2 \dots s_L} = \Gamma^{[1]s_1} \lambda^{[1]} \Gamma^{[2]s_2} \lambda^{[2]} \dots \lambda^{[L-1]} \Gamma^{[L]s_L} \quad (24)$$

From the canonical form a fully left orthonormalized set of matrices A can be obtained by using $A^{[j]} = \lambda^{[j-1]} \Gamma^{[j]}$. To get the right orthonormalized matrices B the singular values are factored out on the right side: $B^{[j]} = \Gamma^{[j]} \lambda^{[j]}$. Inserting these expressions into the orthonormalization conditions Eq. 8 and 13 yield the orthonormalization conditions for the canonical MPS form:

$$\sum_{s_j} \Gamma^{[j]s_j \dagger} (\lambda^{[j-1]})^2 \Gamma^{[j]s_j} = \mathbb{1} \quad (25)$$

$$\sum_{s_j} \Gamma^{[j]s_j} (\lambda^{[j]})^2 \Gamma^{[j]s_j \dagger} = \mathbb{1} \quad (26)$$

These conditions can also be depicted graphically as shown in Eq. 3.

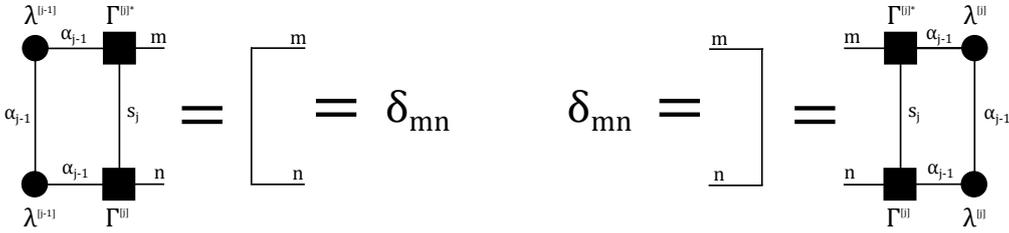


Figure 3: Schematic representation of the orthonormalization Equations 25, 26. Squares represent the Γ matrices, circles the diagonal λ matrices. Horizontal lines are auxiliary, matrix indices, vertical indices are spin indices. A closed line implies summation over that index.

8 Operations on MPS

MPS allows for an efficient way of evaluating operators that only act locally. In this work, operators that act on a single site (e.g. expectation value of \hat{S}_j^z) and operators that act on two consecutive sites are being used.

8.1 Application of a one-site operator

To evaluate how a system changes under the application of an operator, one needs to evaluate how the coefficients c_j of its basis vectors change.

$$c'_i = \sum_j O_{ij} c_j \quad (27)$$

Inserting the MPS for the coefficient c gives

$$c'_{\{s'_i\}} = \sum_{\{s_j\}} O_{\{s'_i\}\{s_j\}} \Gamma^{[0]s_0} \lambda^{[1]} \dots \Gamma^{[L]s_L} \quad (28)$$

If the operator only acts on the spin at site n , the operator becomes a δ at every site but n :

$$O_{\{s'_i\}\{s_j\}} = \delta_{s'_1, s_1} \delta_{s'_2, s_2} \dots \delta_{s'_{n-1}, s_{n-1}} O_{s'_n, s_n} \delta_{s'_{n+1}, s_{n+1}} \dots \delta_{s'_L, s_L} \quad (29)$$

Inserting the operator in Eq. 28 leads to:

$$c'_{\{s'_i\}} = \Gamma^{[0]} \dots \lambda^{[n-1]} \left(\sum_{s_n} O_{s'_n, s_n} \Gamma^{[n]s_n} \right) \lambda^{[n]} \dots \Gamma^{[L]s_L} \quad (30)$$

The only matrices that change during the application of the operator are therefore the matrices $\Gamma^{[n]}$:

$$\Gamma'^{[n]s'_n} = \sum_{s_n} O_{s'_n, s_n} \Gamma^{[n]s_n} \quad (31)$$

Note that the new matrix $\Gamma'^{[n]s'_n}$ will usually no longer satisfy the orthonormalization conditions Eq. 25 and 26.

8.2 Application of two-site operators

Similar to the application of a one-site-operator, also the application of a two-site-operator (an operator that only acts on two adjacent sites) requires only the matrices in its vicinity. If an operator acts on site n and $n+1$ only the matrices $\Gamma^{[n]}$, $\Gamma^{[n+1]}$ and the connecting matrix $\lambda^{[n]}$ are affected (which can be shown in the same way as for the one-site case), but the evaluation becomes easier if the unaffected matrices $\lambda^{[n-1]}$ and $\lambda^{[n+1]}$ are also included in the calculations.

Again it is of interest to calculate the modified coefficients c' :

$$c'_i = \sum_j O_{ij} c_j \quad (32)$$

Since not all matrices are affected by this transformation, not the whole matrix product must be substituted for c_j , but only the affected part, which will be called Θ :

$$\Theta^{s_n s_{n+1}} := \lambda^{[n-1]} \Gamma^{[n] s_n} \lambda^{[n]} \Gamma^{[n+1] s_{n+1}} \lambda^{[n+1]} \quad (33)$$

Note that Θ depends on 4 indices: The physical spin indices s_n , s_{n+1} and the auxiliary matrix indices which shall be called α and γ .

Application of O , according to Eq. 32 yields:

$$\Theta'^{s'_n s'_{n+1}} = \sum_{s_n s_{n+1}} O_{(s'_n s'_{n+1}), (s_n s_{n+1})} \Theta^{s_n s_{n+1}} \quad (34)$$

Now it is necessary to decompose Θ' into the form of a canonical MPS. A matrix $\tilde{\Theta}$ is defined according to Eq. 35, which combines auxiliary indices and spin indices to form a matrix which only depends on 2 indices.

$$\tilde{\Theta}_{(\alpha s_n), (\gamma s_{n+1})} := \Theta'^{s_n s_{n+1}}_{\alpha \gamma} \quad (35)$$

Performing the SV-decomposition of $\tilde{\Theta}$ gives

$$\tilde{\Theta} = \sum_{\beta} U_{(\alpha s_n), \beta} S_{\beta} V_{\beta (\gamma s_{n+1})}^{\dagger} \quad (36)$$

The diagonal matrix of singular values can be identified with the transformed matrix $\lambda'^{[n]}$. Since the matrices $\lambda^{[n-1]}$ and $\lambda^{[n+1]}$ are unchanged, a comparison of Eq. 36 with the desired form in Eq. 33 leads to Eq. 37

$$\begin{aligned} U &= \lambda^{[n-1]} \Gamma'^{[n]} \\ V^{\dagger} &= \Gamma'^{[n+1]} \lambda^{[n+1]} \end{aligned} \quad (37)$$

The matrices $\Gamma'^{[n]}$ and $\Gamma'^{[n+1]}$ can therefore be calculated by multiplying the pseudoinverse of λ onto U and V^{\dagger} , which can easily be done since the matrices λ are diagonal.

Note that in Eq. 35 the combining of the indices increases the size of the matrix $\tilde{\Theta}$ in comparison to Θ' . If each index s_n has 2 possibilities (as for example in a 1D spin $\frac{1}{2}$ chain), the matrix size doubles. If $\lambda^{[n]}$ contained χ Schmidt values, the new matrix $\lambda'^{[n]}$ will contain up to 2χ Schmidt values. Therefore it will in most cases be necessary to truncate the resulting Schmidt values (see Sec. 8.4).

8.3 Expectation values of one-site operators

To calculate the expectation value $\langle O \rangle$ of a state $|\psi\rangle$, represented by the coefficient c_i in the basis $|s_i\rangle$, the following sum must be evaluated.

$$\begin{aligned}\langle O \rangle &= \langle \psi | \hat{O} | \psi \rangle \\ \langle O \rangle &= \sum_{\{s'_i\}} \sum_{\{s_i\}} \langle \{s'_i\} | \hat{O} | \{s_i\} \rangle c_{\{s'_i\}}^* c_{\{s_i\}}\end{aligned}\quad (38)$$

The coefficients c are expressed as canonical MPS, according to Eq. 24. Since every coefficient c is a scalar, it holds that $c^* = c^\dagger$. Furthermore it is convenient to append the $[1 \times 1]$ matrices $\lambda^{[0]} = \lambda^{[L]} = 1$ to the ends of the matrix product.

$$\langle O \rangle = \sum_{\{s'_i\}} \sum_{\{s_i\}} \langle \{s'_i\} | \hat{O} | \{s_i\} \rangle \left(\lambda^{[L]} \Gamma^{[L]s_L \dagger} \dots \lambda^{[1]} \Gamma^{[1]s_1 \dagger} \lambda^{[0]} \right) \left(\lambda^{[0]} \Gamma^{[1]s_1} \lambda^{[1]} \dots \Gamma^{[L]s_L} \lambda^{[L]} \right)\quad (39)$$

If the operator only acts on site n , then all $|s_i\rangle$ with $i \neq n$ can jump over the operator and combine with $\langle s'_i |$ to $\delta_{s'_i s_i}$. The remaining sum over s_1 can be moved to the center of the matrix product and be eliminated, since it is 1 due to the orthonormalization Equation 25.

$$\langle O \rangle = \sum_{s'_n} \sum_{s_2 \dots s_L} \langle s'_n | \hat{O} | s_n \rangle \lambda^{[L]} \Gamma^{[L]s_L \dagger} \dots \lambda^{[1]} \sum_{s_1} \left(\Gamma^{[1]s_1 \dagger} \lambda^{[0]} \lambda^{[0]} \Gamma^{[1]s_1} \right) \lambda^{[1]} \dots \Gamma^{[L]s_L} \lambda^{[L]}\quad (40)$$

This procedure can be repeated for the product $\Gamma^{[2] \dagger} \lambda^{[1]} \lambda^{[1]} \Gamma^{[2]}$ and so forth, up to position $\Gamma^{[n]}$. Here Eq. 25 cannot be used, because not only the matrices, but also the term $\langle s'_n | \hat{O} | s_n \rangle$ depends on s_n . But since the product of matrices is a scalar (a $[1 \times 1]$ matrix), it is the same as the trace of the matrix product. The trace of a product of matrices is invariant under cyclic permutation of the matrices, which allows us to swap c^* with c , yielding Eq. 41.

$$\langle O \rangle = \sum_{s'_n} \sum_{s_n \dots s_L} \langle s'_n | \hat{O} | s_n \rangle \text{Tr} \left(\lambda^{[n-1]} \Gamma^{[n]s_n} \dots \Gamma^{[L]s_L} \lambda^{[L]} \lambda^{[L]} \Gamma^{[L]s_L \dagger} \dots \Gamma^{[n]s_n \dagger} \lambda^{[n-1]} \right)\quad (41)$$

Now Eq. 26 for right orthonormalization can be used to decimate the matrices in the

center of the product yielding the final result:

$$\langle O \rangle = \sum_{s'_n} \sum_{s_n} \langle s'_n | \hat{O} | s_n \rangle \text{Tr} \left(\lambda^{[n-1]} \Gamma^{[n]s_n} \lambda^{[n]} \lambda^{[n]} \Gamma^{[n]s_n \dagger} \lambda^{[n-1]} \right) \quad (42)$$

The same result can also be achieved by using the graphical notation as introduced in Fig. 2. Figure 4 shows a graphical representation of Equation 39, with $|\psi\rangle$ at the bottom, the operator O in between, acting only on site n , and $\langle\psi|$ on top.

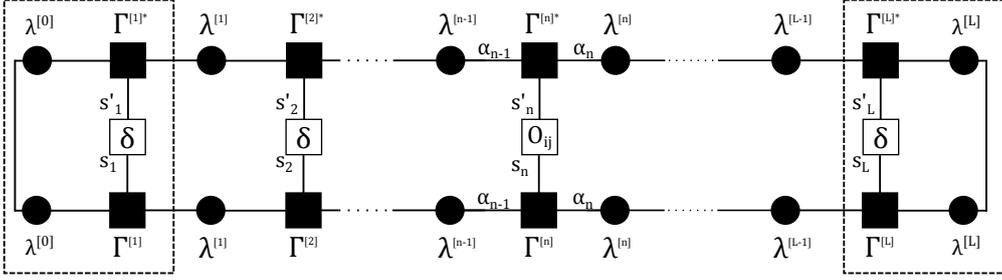


Figure 4: Graphical notation of Eq. 39. The parts enclosed by the dashed rectangles can be identified as the graphical representation of the orthonormalization Equations 25, 26, depicted in Fig. 3

Both ends of the chain in Fig. 4 correspond to the orthonormalization conditions and can therefore be replaced by a Kronecker delta, which corresponds to a simple brace in graphical notation (see 3). This leads to the orthonormalization condition for the next site which allows to iteratively decimate the chain up to the site n where the orthonormalization can no longer be used since $O_{ij}^{[n]} \neq \delta_{ij}$. The remaining expression, corresponding to Eq. 42 can thus be depicted by Fig. 5.

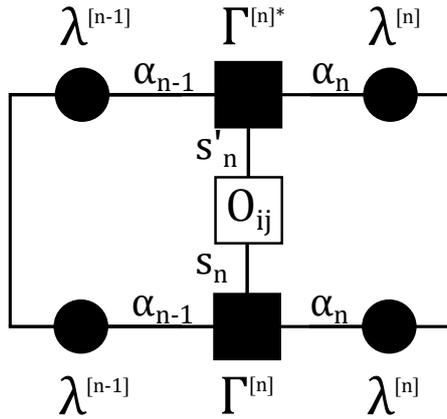


Figure 5: Graphical representation of the expectation value of an operator, which only acts on a single site n . For an analytical expression see Eq. 42

It is important to note that the evaluation of an operator that only acts on one site, does not require the evaluation of the whole matrix product, but only requires the matrices at that position.

8.4 Truncation of Schmidt Values

Up to this point all operations on MPS are, in principle, exact. But when applying 2-site operators several times the number of Schmidt values potentially grows exponentially, rendering practical implementation impossible. A way to still approximate the state is to only keep, at maximum, a number of χ Schmidt values for every bond n and truncate the matrices $\Gamma^{[n]}$, $\lambda^{[n]}$ and $\Gamma^{[n+1]}$ accordingly.

After truncating, the matrix $\lambda^{[n]}$ must be renormalized to satisfy Eq. 19.

$$\lambda = \frac{\lambda_{\text{truncated}}}{\sqrt{\sum_{k=1}^{\chi} \lambda_{kk}^2}} \quad (43)$$

8.5 Time development

MPS provide an efficient way to apply operators that only act locally. It is therefore desirable to decompose the time development operator U into parts that only act on two consecutive sites at most. This can be achieved using the Trotter-Suzuki decomposition [7]. When assuming $\hbar = 1$, the time development operator U is given by

$$\hat{U} = e^{-i\hat{H}t} \quad (44)$$

where \hat{H} is the total Hamiltonian of the system. \hat{H} can be written as a sum \hat{H}_e , for the even numbered bonds and \hat{H}_o for the odd numbered bonds, which only act on every second bond each (Fig. 6).

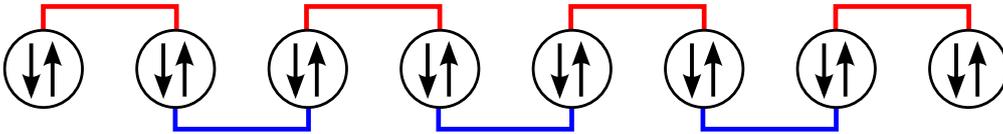


Figure 6: Schematic representation of the Trotter-Suzuki decomposition. Spin sites are depicted by circles, bonds by braces. The total Hamiltonian \hat{H} can be split into \hat{H}_o (red) and \hat{H}_e (blue), which do not commute with each other. The individual parts of each Hamiltonian, however, commute with each other.

\hat{H}_e and \hat{H}_o are both a sum of Hamiltonians that only act on 2 consecutive sites (=bonds). Since for \hat{H}_e and \hat{H}_o the bonds never act on the same site, the bonds do commute with

each other, which allows to easily express the time evolution using \hat{H}_e and \hat{H}_o .

$$\begin{aligned} e^{-it\hat{H}_e} &= e^{-it\sum_{j \text{ even}} \hat{H}_j} = \prod_{j \text{ even}} e^{-i\hat{H}_j t} \\ e^{-it\hat{H}_o} &= e^{-it\sum_{j \text{ odd}} \hat{H}_j} = \prod_{j \text{ odd}} e^{-i\hat{H}_j t} \end{aligned} \quad (45)$$

But since \hat{H}_e and \hat{H}_o do not commute, the total time development operator cannot be written as a product of these two operators in Eq. 46.

$$e^{-it\hat{H}} \neq e^{\hat{H}_e} e^{\hat{H}_o} \quad (46)$$

This problem can be solved using the Trotter-Suzuki decomposition. The time t is divided into smaller steps Δt . When the system is only developed a small step Δt , the time development operator U can be expanded using the Baker–Campbell–Hausdorff formula:

$$e^{At} e^{Bt} = e^{(A+B)t + \frac{1}{2}[A,B]t^2 + O(t^3)} \quad (47)$$

$$\begin{aligned} e^{-i\hat{H}_e \Delta t} e^{-i\hat{H}_o \Delta t} &= e^{-i(\hat{H}_e + \hat{H}_o)\Delta t - \frac{1}{2}[\hat{H}_e, \hat{H}_o]\Delta t^2 + O(\Delta t^3)} \\ e^{-i\hat{H}_e \Delta t} e^{-i\hat{H}_o \Delta t} &= e^{-i\hat{H}\Delta t + O(\Delta t^2)} \end{aligned} \quad (48)$$

Eq. 48 shows that when performing only a small step Δt , by applying $e^{-i\hat{H}_o \Delta t}$ and $e^{-i\hat{H}_e \Delta t}$ consecutively instead of the full operator $e^{-i\hat{H}\Delta t}$, the error is of the order $O(\Delta t^2)$. When performing $\frac{t}{\Delta t}$ steps the accumulated error is of $O(\Delta t)$ and can therefore be reduced by choosing a smaller step size Δt during simulation. An even better decomposition is given by Eq. 49 [7].

$$\hat{U}_{TZ} = e^{-i\hat{H}\Delta t} \approx e^{-i\hat{H}_e \frac{\Delta t}{2}} e^{-i\hat{H}_o \Delta t} e^{-i\hat{H}_e \frac{\Delta t}{2}} \quad (49)$$

This can again be proven by using the Baker–Campbell–Hausdorff formula:

$$\begin{aligned} e^{-i\hat{H}_e \frac{\Delta t}{2}} e^{-i\hat{H}_o \Delta t} e^{-i\hat{H}_e \frac{\Delta t}{2}} &= e^{-i\Delta t(\frac{\hat{H}_e}{2} + \hat{H}_o) - \frac{1}{2}[\frac{\hat{H}_e}{2}, \hat{H}_o]\Delta t^2 + O(\Delta t^3)} e^{-i\hat{H}_e \frac{\Delta t}{2}} \\ &= e^{-i\Delta t(\hat{H}_e + \hat{H}_o) + (\frac{1}{4}[\hat{H}_e, \hat{H}_o] + \frac{1}{8}[\hat{H}_e, \hat{H}_e] + \frac{1}{4}[\hat{H}_o, \hat{H}_e])\Delta t^2 + O(\Delta t^3)} \\ &= e^{-i\hat{H}\Delta t + O(\Delta t^3)} \end{aligned} \quad (50)$$

Time development to a time t is done by applying the operator U_{TZ} from Eq. 49 $\frac{t}{\Delta t}$ times, causing a total error of $O(\Delta t^2)$. In practice this 2nd-order TZ decomposition requires almost no additional computational effort, compared to the 1st-order decomposition, because consecutive time developments with $\frac{\hat{H}_e}{2}$ can be combined to a single

time development with \hat{H}_e , as long as no measurements are performed in between these steps.

$$\begin{aligned} (\hat{U}_{TZ})^n &= \left(e^{-i\hat{H}_e \frac{\Delta t}{2}} e^{-i\hat{H}_o \Delta t} e^{-i\hat{H}_e \frac{\Delta t}{2}} \right)^n \\ &= e^{-i\hat{H}_e \frac{\Delta t}{2}} \left(e^{-i\hat{H}_o \Delta t} e^{-i\hat{H}_e \Delta t} \right)^{n-1} e^{-i\hat{H}_o \Delta t} e^{-i\hat{H}_e \frac{\Delta t}{2}} \end{aligned} \quad (51)$$

9 Implementation and validation of 1D-Simulation

The following algorithm for simulating systems with MPS and TEBD was used [9]:

1. Choose a starting configuration (in general a linear combination) and bring it into MPS form (Sec. 4.2).
2. Transform it into canonical form (Sec. 7)
3. Time evolve using TZ decomposition with stepsize dt (Sec. 8.5). During time development truncate the number of Schmidt values to a maximum of χ .
4. Measure $\langle \hat{S}^z \rangle$ for every site of the system. (Sec. 8.3)
5. Iterate from 3 until desired time T_{end} .

To validate the obtained results the following procedure was chosen:

1. Simulate small 1D systems that can be compared to full diagonalization: Sec. 9
2. Simulate larger 1D system to observe "Quantum Bowling" in 1D: Sec. 10
3. Simulate small 2D system and compare to full diagonalization: Sec. 12
4. Simulate "Quantum Bowling" in 2D: Sec. 13

To validate the simulation results of the MPS, exact diagonalization of the Hamiltonian was used. Because the Hamiltonian conserves the total spin of the system in z direction S^z , it is possible to only choose basis states that share a given total S^z . To simulate the time evolution of k flipped spins on a 1D-lattice of length L one needs to consider all basis states where k spins are \uparrow and $L - k$ are \downarrow .

The number of possibilities to distribute k particles on a lattice of size L , where only at maximum one particle may exist per lattice site, is given by the binomial coefficient.

$$n_{states} = \binom{n}{k} \quad (52)$$

For the case of $L = 20$ and $k = 2$ this gives a total of 190 basis states, which can easily be treated on any modern computer.

Fig. 7 shows the time development of an excitation of two neighboring particles. Two distinct propagation speeds can be observed: A small signal that reaches the domain

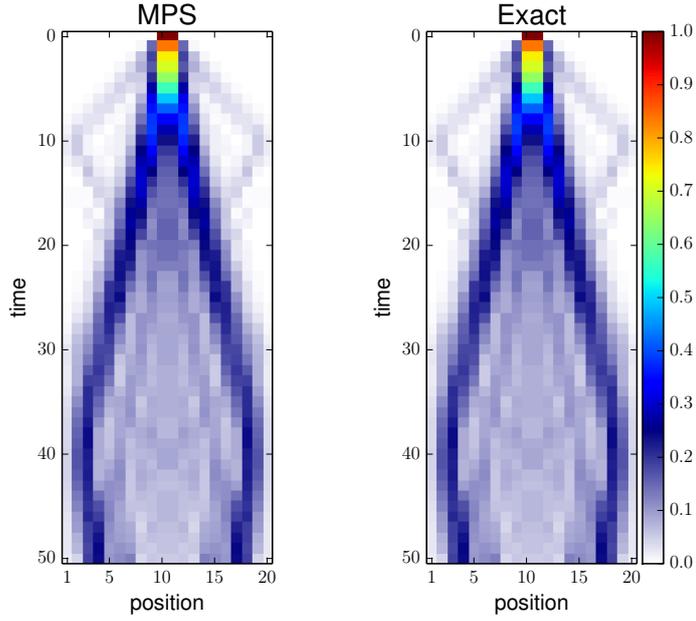


Figure 7: Expectation value of particle density $n^{[j]}$ for every position in the chain for times between 0 and 50 for a 2 spin excitation with $J^{xy} = 1.0$, $J^z = 2.0$, $dt = 0.01$, $\chi = 40$.

Left: Obtained by exact diagonalization, Right: Obtained by MPS simulation. The results are qualitatively indistinguishable, the quantitative difference is in the order of 1 %.

boundaries at $t = 10$ and a strong signal that reaches the boundaries at $t \approx 35$. The fast propagating branch is the signal of a free particle as can be seen by comparison of a single spin excitation. The slowly propagating, stronger signal is the signal caused by the 2 particles moving together as a bound couple [5]. Increasing the bond energy J^z leads to a strong bound signal, which propagates slower while the signal of the free particle becomes weaker. This concept is further noted in Sec. 10.

The simulation error was estimated as the maximum difference between the results obtained via diagonalization and MPS.

$$\delta = \max_j \left| \langle S_j^z \rangle_{diag} - \langle S_j^z \rangle_{MPS} \right| \quad (53)$$

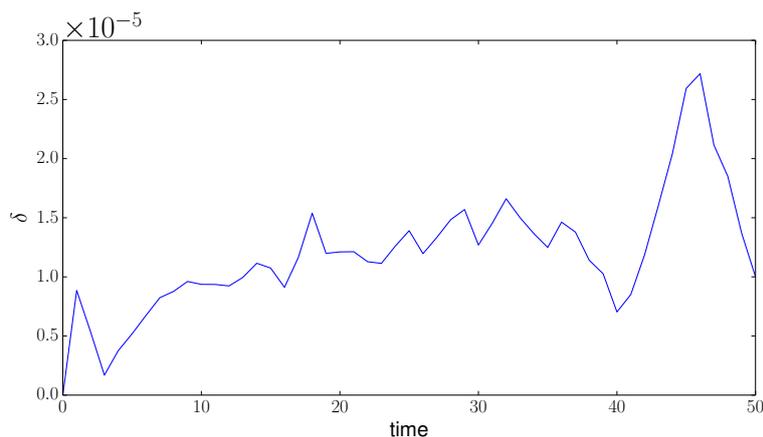


Figure 8: MPS simulation error according to Eq. 53 over time for the system of Fig. 7. Note that the maximum error is smaller than $3 \cdot 10^{-5}$ and therefore about 4 magnitudes smaller than the expectation values.

10 Quantum Bowling in 1D

At large bond strengths $\left| \frac{J^z}{J^{xy}} \right|$, chains of flipped spins on a background of oppositely oriented spins will decay only slowly over time. This can be seen both by simulation results (Fig. 9), and analytical treatment via the Bethe ansatz [1, 5].

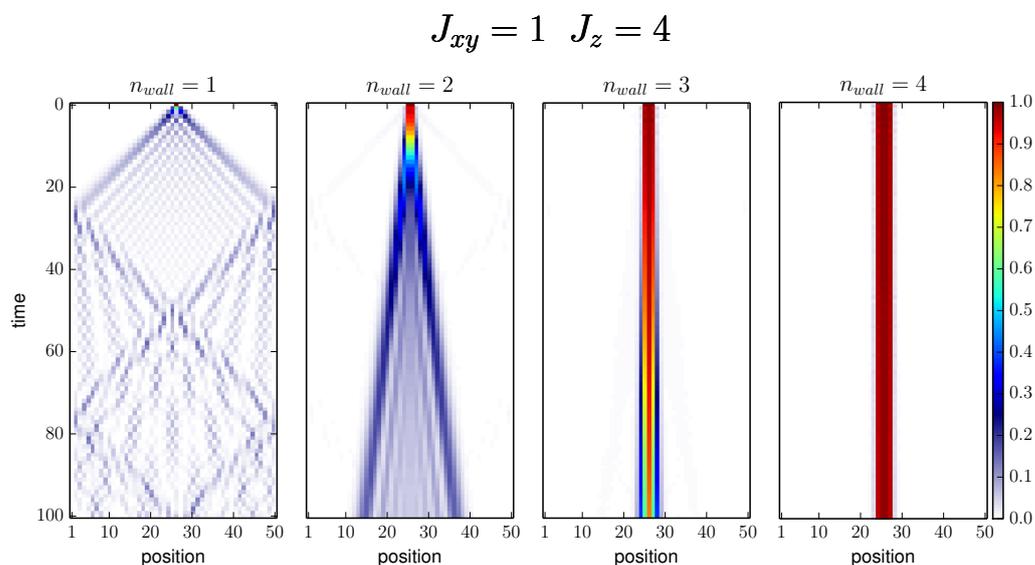


Figure 9: Expectation value of particle density $n^{[j]}$ for every position j in the chain for times between 0 and 100 for an excitation, where n_{wall} particles are created at $t = 0$. $J^{xy} = 1.0$, $J^z = 4.0$, $dt = 0.05$, $\chi = 40$. Walls of several particles are almost stationary compared to a single particle (leftmost image).

At sufficiently high J^z these walls are almost stationary compared to freely moving particles.

[1] and [6] have studied the behavior of free particles hitting these walls under various conditions. Some of their results have been reproduced in this work.

Figure 10 shows a particle that is moving towards a wall of 5 particles, which are bound together by the high bond energy J^z . The particle hits the wall and another particle is emitted off the wall at the other end, which resembles the well known Newton Cradle. But a careful examination shows that the wall is shifted by two sites during the process, as opposed to the newton cradle, where the balls at rest shift by one site (one ball is added on the left, one ball leaves at the right). The phenomenon becomes clearer when examining conservation of energy, in particular the bond energy [6].

The easiest way to create an incoming particle is by flipping a single spin (Fig. 10 left), which will move towards the wall due to the hopping process. But due to the strong spread of the particle, results are hard to interpret. A linear combination of single flipped spins, with their coefficients distributed according to a Gaussian distribution, as used in [6], lowers the spread and yields easier to interpret results (Fig. 10 right).

$$|\psi\rangle = \sum_x c_x |\uparrow_x \uparrow_{w_s} \uparrow_{w_{s+1}} \dots \uparrow_{w_e}\rangle$$

$$c_x \propto e^{-\frac{(x-x_0)^2}{2\sigma^2}} e^{i(x-x_0)k}$$
(54)

where w_s and w_e are the start and end position of the target wall.

Fig. 11 shows in a simplified schematic what happens during the collision. Before the incoming particle hits the wall, the leftmost particle of the wall hops towards the incoming particle, forming a bond. This process is repeated until only one particle of the wall remains. This single particle then propagates further to the right. It is impossible for the incoming particle to directly reach the wall, since this would form an additional bond (shown in red), violating the conservation of energy.

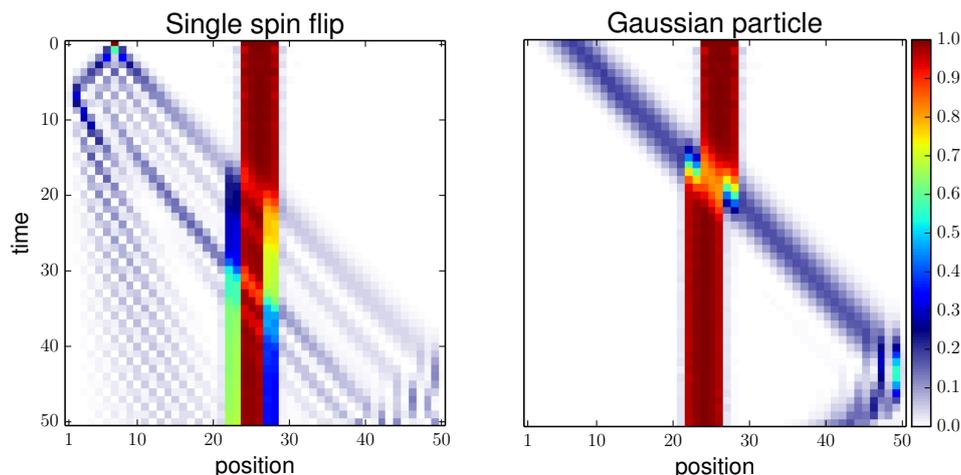


Figure 10: Plot of particle density $\langle n_j \rangle$ over time, for a particle moving through a wall of 5 particles at $J^{xy} = 1$, $J^z = 4$. Left: Particle is a single flipped spin. Right: Particle is a Gaussian packet according to Eq. 54 with $k = -\frac{\pi}{2}$ and $\sigma = 3$

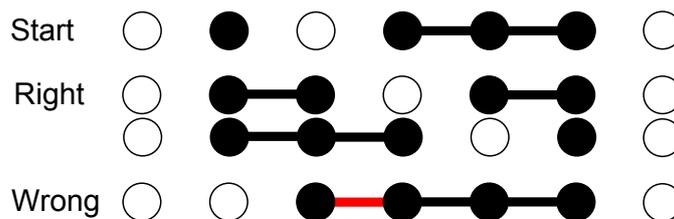


Figure 11: Schematic explanation of hole formation. Empty circles represent empty sites, filled circles occupied sites. When a particle approaches a wall (Start), the leftmost wall particle hops to the left creating a hole (Right). If the incoming particle hopped towards the wall, an additional bond would be created, violating energy conservation (Wrong).

11 Extension to 2 dimensions

If one of the two dimensions of the system is small (in the implementation used ≤ 5 sites), the methods used for 1D simulation can easily be adapted to simulate a 2D system.

To simulate spins on a rectangular 2D-grid of size $L_x \times L_y$ we assume without loss of generality that $L_y \leq L_x$. All spins within one column are combined to form a subsystem. As opposed to a single spin, which can be described by 2 basis states, each subsystem is now described by 2^{L_y} basis states. The total system is then merely a one-dimensional chain of subsystems and can therefore efficiently be simulated using the techniques described in Sec. 4.

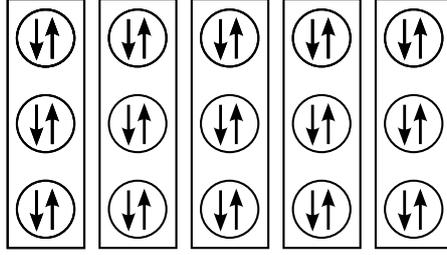


Figure 12: Schematic of 2D MPS-implementation. All spins (circles) along a column are combined to form a subsystem (rectangle). The total system becomes a 1D chain of subsystems.

A major problem of this approach is the exponential growth of complexity with the increase of L_y . While the algorithm scales well with an increase of L_x , only small system widths L_y can be treated. One of the limiting factors of this approach is the Hamilton operator H_{bond} , which describes the interaction of 2 adjacent subsystems and the corresponding time development operator U , which are both of size $[2^{2L_y} \times 2^{2L_y}]$. The total number of matrix elements of U is thereby proportional to 16^{L_y} . On a personal computer, systems up to width $L_y = 5$ could be simulated. To simulate a system of width $L_y = 10$, a $\approx 10^6$ fold increase in computational power would be needed.

When calculating the time evolution of the 2D system a Trotter-Suzuki decomposition can be used just as in the 1D case.

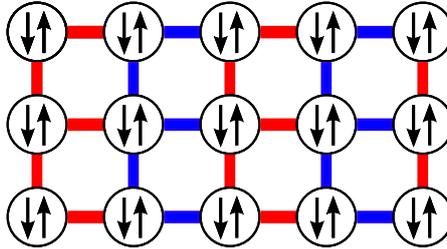


Figure 13: Schematic of the Trotter-Suzuki-Decomposition in 2D. Red and blue represent the time evolutions due to H_{even} and H_{odd} . Note that at the end of the chain a time evolution on a single subsystem remains.

The time evolution can be achieved using two types of operators: A 2-site operator and a 1-site operator. As introduced in Eq. 4, for the 2D case, the Hamiltonian does in general depend on 4 parameters: J_{\parallel}^{xy} , J_{\perp}^{xy} which describe hopping along the horizontal and vertical direction respectively, and J_{\parallel}^z , J_{\perp}^z which describe the bond strength along the corresponding axes.

12 Validation of 2D implementation

As in Sec. 9 the output of the MPS algorithm was validated by comparing it to exact diagonalization for initial conditions of 2 particles on a system of size $[4 \times 6]$. The resulting time evolution is partially displayed in Fig. 14. Figure 15 shows the simulation error over time. The error estimate was calculated by Eq. 55 in correspondence to Eq. 53 by taking the maximum difference between both simulation results.

$$\delta = \max_{x,y} \left| \langle S_{z,diag}^{[x,y]} \rangle - \langle S_{z,MPS}^{[x,y]} \rangle \right| \quad (55)$$

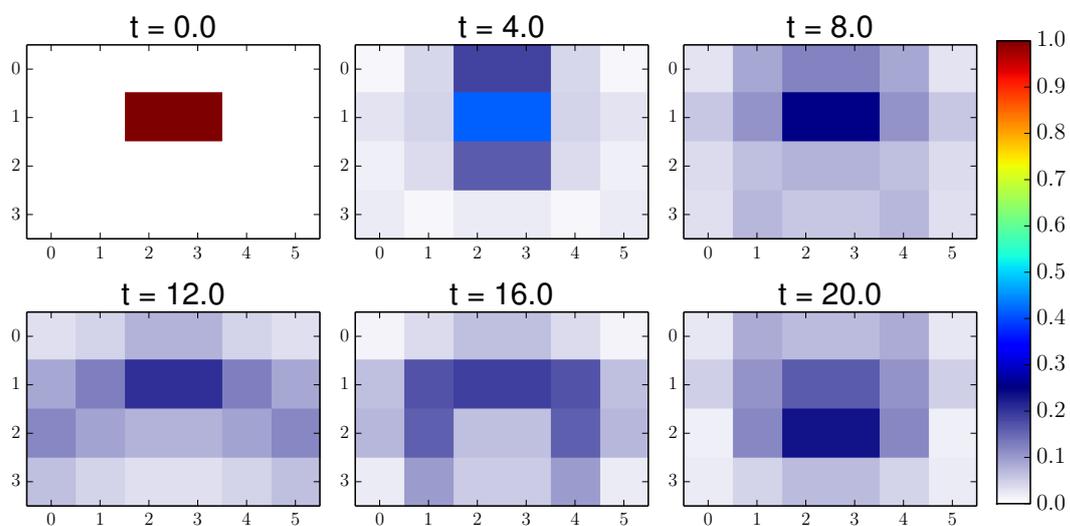


Figure 14: Expectation value of particle density $n_{x,y}$ for every position at selected times after a 2 spin excitation at $t = 0$ with $J_{\parallel}^{xy} = J_{\perp}^{xy} = 1.0$, $J_{\parallel}^z = J_{\perp}^z = 4.0$, $dt = 0.02$, $\chi = 40$.

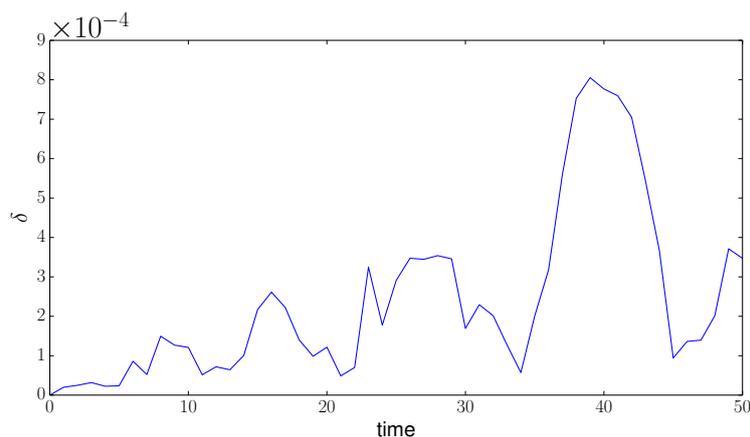


Figure 15: Simulation error of the simulation displayed in Fig. 14, estimated by Eq. 55. The maximum error is smaller than $9 \cdot 10^{-4}$

13 Quantum Bowling in 2D

13.1 2D Quantum-Bowling at high J_z

Just as in the 1D case (Sec. 10) one can prepare a "wall" of particles bound together by high J_z and a single particle that hops towards that wall. A simulation of one possible initial condition is shown in Fig. 16. Unlike in the 1D case the single particle does not penetrate the wall, but bounces off it. This can be explained by conservation of energy as displayed in Fig. 17.

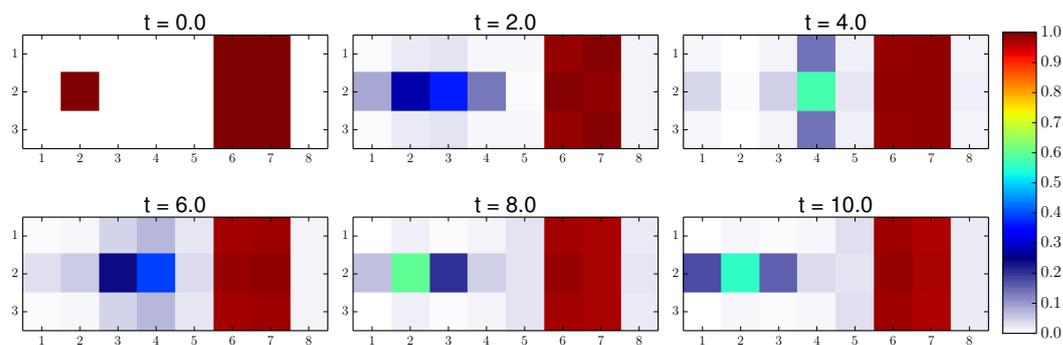


Figure 16: Expectation value of $n_{x,y}$ for every position at selected times, for a single flipped spin hopping towards a wall of depth 2. $J_{\parallel}^{xy} = J_{\perp}^{xy} = 1.0$, $J_{\parallel}^z = J_{\perp}^z = 4.0$, $dt = 0.01$, $\chi = 40$. The single particle bounces off the wall.

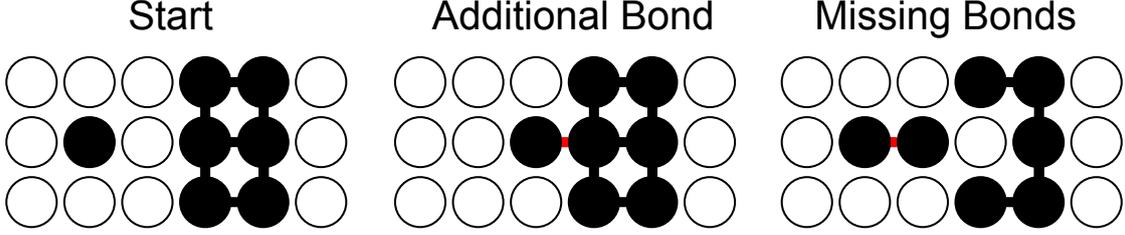


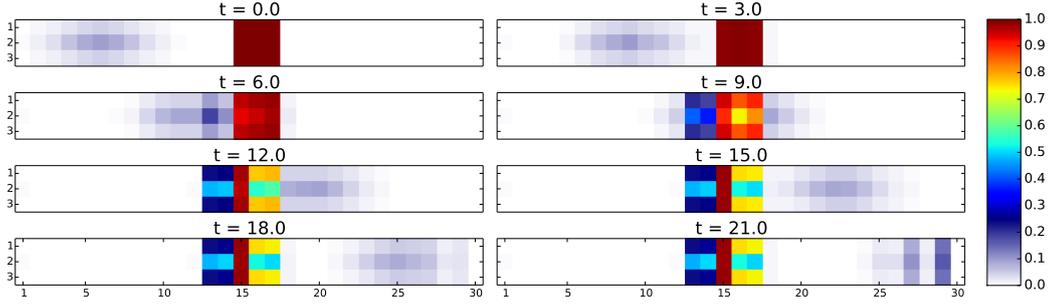
Figure 17: Schematic representation of J^z -bonds. Empty circles represent empty sites, filled circles occupied sites, connecting lines represent bonds by S^z -interaction. When a single particle approaches a wall (Start) it cannot hop towards the wall, because this would create an additional bond, similar to the 1D-case. But in contrast to the 1D case a wall particle cannot leave the wall and hop towards the incoming particle, since this would break two vertical bonds which are not replaced.

13.2 Transmission of a single particle through a 2D wall

Sec. 13.1 demonstrates that due to energy conservation, at high vertical bond energies J_{\perp}^z incoming particles do not penetrate the wall, but are reflected. To allow the incoming particle to be (at least partially) transmitted, anisotropic coupling was introduced. Instead of using the same hopping term J^{xy} for both horizontal and vertical hopping, a separate hopping J_{\parallel}^{xy} for horizontal hopping and J_{\perp}^{xy} for vertical hopping were introduced. In similar fashion different values J_{\parallel}^z and J_{\perp}^z for vertical and horizontal bond energies were allowed.

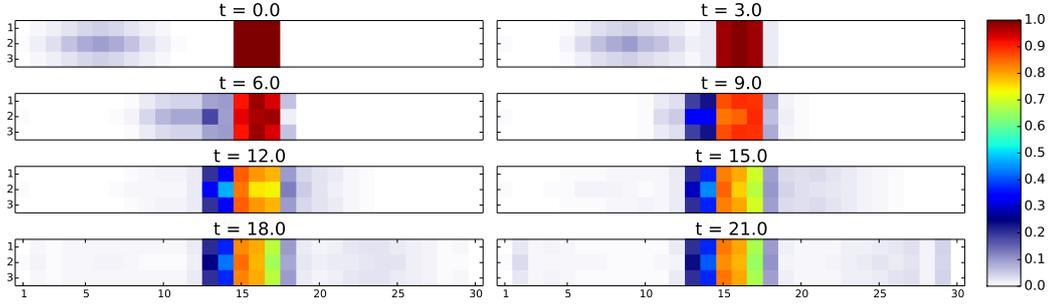
To simulate the effect of the coupling energies on the transmission, the impact of a single particle onto a wall of depth 3 and height 3 from position $x_{start} = 14$ to $x_{end} = 17$ was simulated on a grid of size 30×3 . As a first attempt, the incoming particle was a single flipped spin, as seen in Fig. 16, but this proved to be inadequate due to its spread in both horizontal and vertical direction, complicating the interpretation of obtained simulation results. Therefore a linear combination of states with a Gaussian distribution along the x-axis was used in later simulations, similar to the 1D-case (Eq. 54, Fig. 10r). While this solves the issue of the unwanted spread in the x-direction, the particle still spreads vertically. To prevent this, the incoming particle is chosen to be an eigenstate along the y-axis and normally distributed along the x-axis. Since the problem is symmetric, all eigenstates are symmetric or antisymmetric, and the corresponding spin expectation value of the eigenstates is therefore always symmetric. For the simulations the eigenstate with the lowest energy was chosen, since it has a single particle density peak in the center and drops towards the edges.

Fig. 18 shows the impact of a particle at $J_{\perp}^{xy} = 0.1$, $J_{\perp}^z = 0$. Since there is almost no coupling between the rows of the system, the particle is almost fully transmitted and the wall partially shifts by 2 sites as observed in the 1D case. When the coupling between the rows rises, energy conservation starts to obstruct the movement through the wall (Fig. 19).

Figure 18: $\langle n_{x,y} \rangle$ for every position at selected times.

$$J_{\parallel}^{xy} = 1.0, \mathbf{J}_{\perp}^{xy} = \mathbf{0.1}, J_{\parallel}^z = 8.0, \mathbf{J}_{\perp}^z = \mathbf{0.0}, dt = 0.05, \chi = 30.$$

The rows are almost independent, the particle is nearly fully transmitted.

Figure 19: $\langle n_{x,y} \rangle$ for every position at selected times.

$$J_{\parallel}^{xy} = 1.0, \mathbf{J}_{\perp}^{xy} = \mathbf{0.6}, J_{\parallel}^z = 8.0, \mathbf{J}_{\perp}^z = \mathbf{0.0}, dt = 0.05, \chi = 30.$$

Only a fraction of the particle is transmitted.

13.3 Transmission rate of a single particle through a 2D wall

Fig. 16 shows that in the limit of high vertical bond energies an incoming particle is fully reflected. On the other hand, in the special case that $J_{\perp}^z = 0$ and $J_{\perp}^{xy} = 0$ (i.e. there is no vertical coupling), the 2D system becomes a stack of independent 1D systems, all of which will show the transmissive behavior seen in Fig. 10. It is therefore of interest to observe the transmission rate of particles at intermediate values of vertical bond energies.

For each configuration of energy parameters, two simulations were performed: First one simulation, featuring only the wall as a starting configuration to get a background level of spin density n_B . Then a second simulation with the wall and a particle as described above to obtain the transmitted particle density. After $t = 24$ (the time required for the incoming particle to pass through the wall and reach the end of the simulation domain) the particle density n_P on the right of the wall was summed up and the sum over the

background particle density n_B subtracted (Eq. 56).

$$T = \sum_{x > x_{end}+1} \sum_y \langle n_P^{x,y} \rangle - \langle n_B^{x,y} \rangle \quad (56)$$

Fig. 20 shows the dependence of the transmission rate on the vertical hopping and bond parameters J_{\perp}^{xy} and J_{\perp}^z , for a fixed set of horizontal parameters.

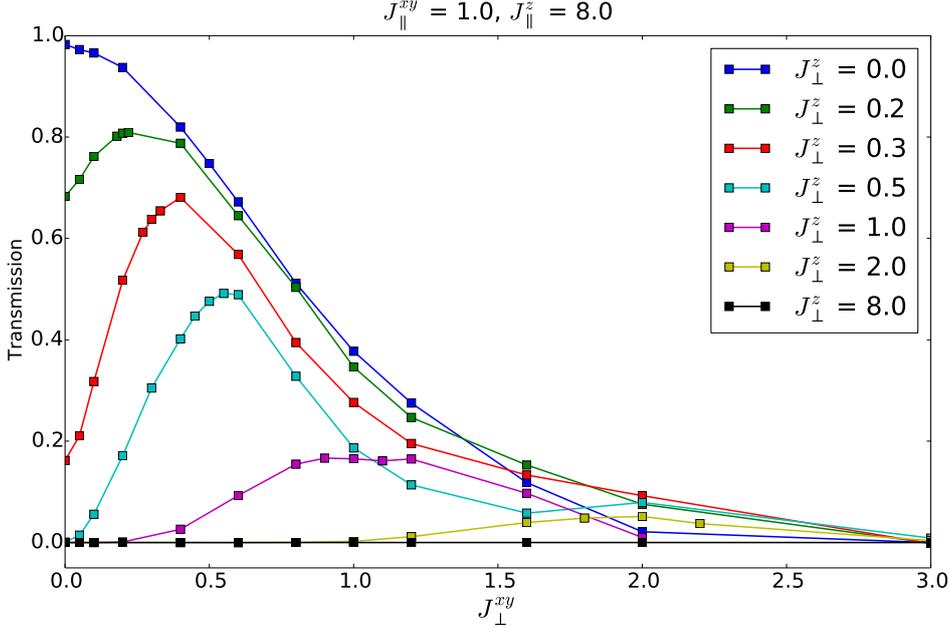


Figure 20: Graph of integrated, transmitted particle density (Eq. 56) of a single particle hitting a wall of thickness 3, simulated on a 30×3 grid with $\chi_{max} = 30$, with $J_{\parallel}^{xy} = 1$ and strong horizontal bond strength $J_{\parallel}^z = 8$. For an error estimation see Sec. 13.4.

At $J_{\perp}^{xy} = 0$, $J_{\perp}^z = 0$ the rows are fully decoupled and transmission is almost 100 %. Increasing vertical bond strength J_{\perp}^z leads to a decrease in transmission due to the conservation of energy (see Sec. 13.1). As J_{\perp}^z rises, the transmission peak shifts to higher values of J_{\perp}^{xy} . To further refine the configuration of the transmission maxima, for each J_{\perp}^z , additional simulations were performed in the range of $J_{\perp}^{xy} = J_{\perp}^z \pm 10\%$. Fig. 21 shows the hopping parameter J_{\perp}^{xy} of maximum transmission for different vertical interaction strength J_{\perp}^z .

The data from Fig. 21 suggest that the maxima lie on the line $J_{\perp}^{xy} = J_{\perp}^z$, which corresponds to the isotropic Heisenberg model. The deviations from the line might be caused by simulation error, since at the number of Schmidt values $\chi_{max} = 30$ used, the simulation has not yet fully converged to its solution.

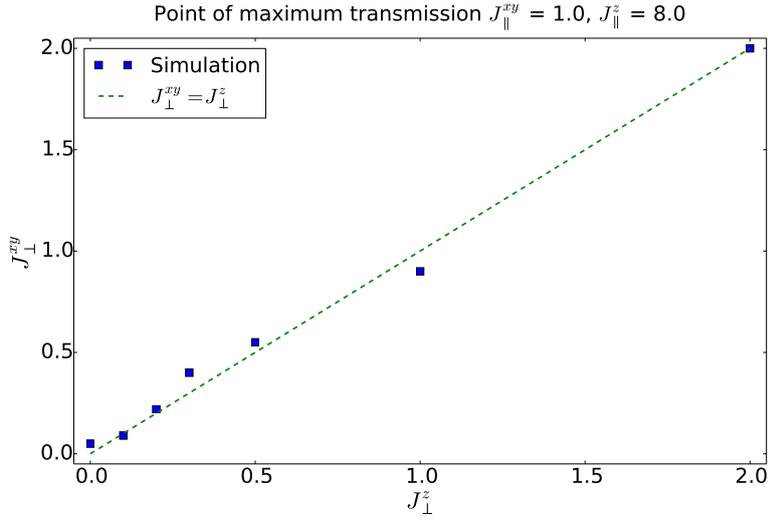


Figure 21: Vertical coupling at maximum transmission for fixed values of horizontal coupling $J_{\parallel}^{xy} = 1$, $J_{\parallel}^z = 8$. Markers indicate the transmission maxima, the line is the model $J_{\perp}^{xy} = J_{\perp}^z$.

13.4 Error estimation of transmission rate

When describing a general quantum state exactly, the maximum number of Schmidt values grows exponentially with the distance from the chain's ends. For the simulated grid of size 30×3 , the maximum number of Schmidt values is

$$\chi_{max} = 8^{15} \approx 4 \cdot 10^{13} \quad (57)$$

To be able to simulate this system on a contemporary PC, the Schmidt values need to be heavily truncated. In the case of Fig. 20, $\chi_{max} = 30$ was chosen. It is rather remarkable that despite this extreme truncation (only 10^{-12} of the Schmidt values are considered), the maximum discarded weight per timestep is of the order of 10^{-4} . To examine the choice of $\chi_{max} = 30$, a few selected configurations were simulated with different values of χ , and their results compared.

Figure 22 shows that the systems have not yet fully converged at $\chi = 30$, which was used for the simulations. In particular for the cases $J_{\perp}^z = 0.5$ and $J_{\perp}^z = 1.0$ the remaining slope at $\chi = 100$ indicates that a significantly higher number of Schmidt values might be necessary to achieve quantitative results. However, considering the absolute variation of the transmission over the the tested range of χ , the results will most likely not change qualitatively at higher simulation accuracy.

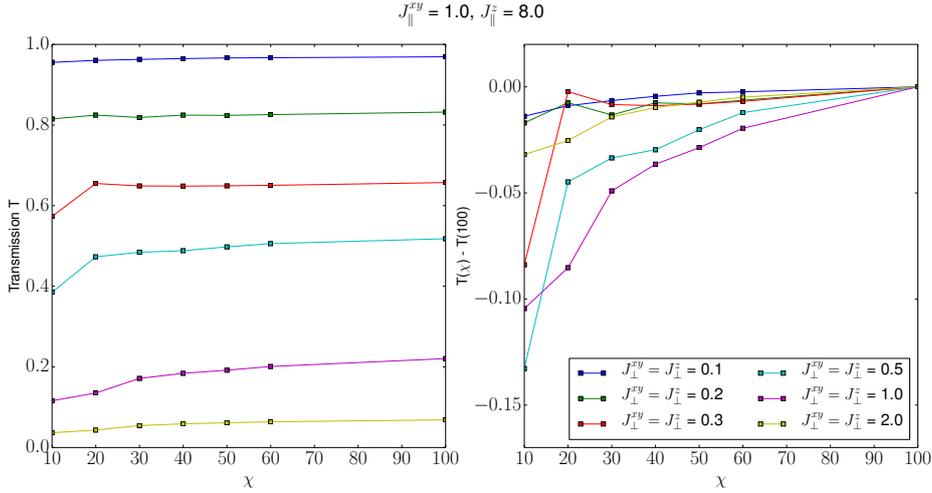


Figure 22: Convergence of particle transmission for $J_{\parallel}^{xy} = 1$, $J_{\parallel}^z = 8$ in the case of vertical isotropic coupling, for different maximum number χ of Schmidt values used. The left graph shows the absolute value of transmission, the right graph the difference to the result simulated with the highest $\chi = 100$.

14 Summary

A simulation of the spin $\frac{1}{2}$ Heisenberg model was implemented using TEBD and validated in the 1D case by comparison with full diagonalization (Sec. 9) and by simulating "Quantum Bowling" in 1D. In agreement with [6] nearly full transmission of an incident particle was observed in the 1D case.

The simulation was extended to 2D by using full diagonalization along the second axis which proved to be not very scaleable for systems with a large second dimension, but sufficient to observe the behavior of incident particles on a 2D wall. Verification of the obtained simulation results was again performed using full diagonalization (Sec. 12).

The dependence of the transmission rates on the coupling energies was investigated in Sec. 13.2. For high bond strengths $J^z \gg J^{xy}$, the incoming particle was found to be fully reflected, which can be explained by conservation of energy (Sec. 13.1). For lower bond strengths the incoming particle is partly transmitted and its maximum transmission was found to occur when the couplings along the axis perpendicular to the incident direction are isotropic: $J_{\perp}^{xy} = J_{\perp}^z$ (Fig. 21).

The impact of the number χ of Schmidt values used was examined in Sec. 13.4. For the simulated systems a maximum number of $\chi = O(100)$ was found to be sufficient for qualitatively correct results.

References

- [1] Ganahl M. (2014). "Dynamics of Strongly Correlated One-Dimensional Quantum Systems using Matrix Product States", Doctoral Thesis at Graz University of Technology
- [2] M. Endres, M. Cheneau, T. Fukuhara, C. Weitenberg, P. Schauss, C. Gross, L. Mazza, M. C. Bañuls, L. Pollet, I. Bloch, and S. Kuhr, *Science* 334, 200 (2011) <http://www.sciencemag.org/content/334/6053/200>
- [3] T. Fukuhara, A. Kantian, M. Endres, M. Cheneau, P. Schauß, S. Hild, D. Bellem, U. Schollwöck, T. Giamarchi, C. Gross, I. Bloch, S. Kuhr (2012). "Quantum dynamics of a single, mobile spin impurity" <http://arxiv.org/abs/1209.6468>
- [4] G. Vidal, (2008) "Efficient classical simulation of slightly entangled quantum computations" <http://arxiv.org/abs/quant-ph/0301063>
- [5] M. Ganahl, E. Rabel, F. Essler, and H. G. Evertz, (2012). "Observation of Complex Bound States in the Spin-1/2 Heisenberg XXZ Chain Using Local Quantum Quenches" *Phys. Rev. Lett.* 108, 077206 <http://journals.aps.org/prl/abstract/10.1103/PhysRevLett.108.077206>
- [6] M., Ganahl, M. Haque, H.G. Evertz (2013). "Quantum Bowling: Particle-hole transmutation in one-dimensional strongly interacting lattice models". <http://arxiv.org/abs/1302.2667>
- [7] U. Schollwöck (2011). "The density-matrix renormalization group in the age of matrix product states". *Annals of Physics* 326 (2011), pp. 96–192. <http://www.sciencedirect.com/science/article/pii/S0003491610001752>
- [8] P. Jordan, E. Wigner. "Über das Paulische Äquivalenzverbot". *Zeitschrift für Physik* 47, No. 9. (1928), pp. 631-651
- [9] H.G. Evertz, (2014). "Time Evolution of one-dimensional spin systems, using Matrix Product States", Lecture notes for the course "Advanced Computational Physics" at the Graz University of Technology https://itp.tugraz.at/~evertz/CP/MPS_2014.pdf (09.10.2014)
- [10] Hogben, L. 2013, *Handbook of Linear Algebra*, Second Edition, Chapman and Hall/CRC, Hoboken.