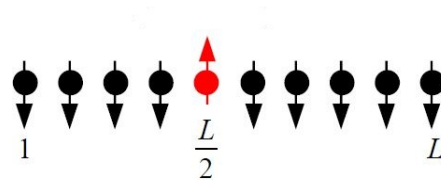


# Quantum Dynamics: Time evolution with Matrix Product States

Write a program to simulate the time evolution of spin excitations in the one-dimensional  $S = \frac{1}{2}$  Heisenberg chain (with  $J_{xy} = 1$  and parameter  $J_z$ ) using Matrix Product States. Employ the Time Evolving Block Decimation (TEBD) algorithm, as explained in the lecture notes.

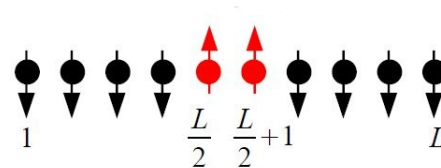
Consider the following initial states:

1. A single-spin excitation of the ferromagnetic ground state



Single spin up: need only  $\chi=2$  at any time t:  
 Schmidt decomp for any subdivision Left Right:  
 $|\psi\rangle = \sum_x c_x |x\rangle$   
 $= |0\rangle_L \text{ times } (\sum_{(x \text{ in R})} c_x |x\rangle)$   
 $+ (\sum_{(x \text{ in L})} c_x |x\rangle) \text{ times } |0\rangle_R$

2. A two-spin-excitation of the ferromagnetic ground state, with two neighboring spins pointing opposite to all others:



- The program should output the time evolution of the expectation value of  $S_i^z$  for every site of the chain. For easy interpretation it is advantageous to show the results color-coded in space-time (e.g. with the Matlab command `imagesc`)
- Determine how the intensity and the velocity of propagation of the two kinds of states depend on  $J_z$ . Try to identify the propagation of any bound states. Use, e.g.,  $J_z = 0, 0.5, 1, 1.5$ . What happens for  $J_z \gg 1$ ?

+ entanglement entropy (see page 3)  
 + discarded weight

Also try negative  $J_z$

## Hints:

### Initial state:

The canonical MPS representation of a product state needs only a  $\chi = 1$  matrix of dimension  $\chi = 1$ . All  $\lambda_i = 1$ , and the  $\Gamma$ -matrices are the same as the  $A^{\uparrow/\downarrow}$ -matrices introduced at the beginning of the lecture notes. For the case of a two-spin-excitation mentioned above, this means, in case of a chain of  $L = 10$  sites:

$$\begin{array}{rcccccccccccc}
 |\Psi\rangle & = & | & \downarrow & \downarrow & \downarrow & \downarrow & \uparrow & \uparrow & \downarrow & \downarrow & \downarrow & \downarrow & \rangle \\
 \lambda & = & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 \Gamma^\uparrow & = & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 \Gamma^\downarrow & = & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1
 \end{array}$$

In order to keep the program simple, you can use matrices of constant dimension  $\chi > 1$ , e.g.  $\chi = 20$  or  $30$ . In this case the values above are the  $(1, 1)$  matrix elements, and all others are zero.

You should use a system size of  $L = 50$  or more, and simulate up to  $t = L$ .

### Sketch of the program:

- Initialization: define external parameters ( $J_z, L, \Delta t, \dots$ ) e.g. Delta t = 0.1
- Initialize matrices  $\lambda^{[n]}$ - and  $\Gamma^{[n]s_n}$  for the initial physical spin state (pad with zeroes)
- Loop over time steps (Left/right: end can use matrices instead of row/column; then use only the (1,1) element of result)
  - Do time evolution, by applying the unitary 2-site operators  $\hat{u}_{n,n+1} = e^{-i\tau\hat{h}_{n,n+1}}$  (see below)
  - Calculate the expectation value of the 1-site operators  $\hat{S}_z^{[n]}$  (see lecture notes)
- Output and analysis of results

### Time evolution:

- In order to apply the unitary 2-site operators  $\hat{u}_{n,n+1} = e^{-i\tau\hat{h}_{n,n+1}}$ , you first need the 4 x 4-matrix  $\hat{u}_{kl}^{ij} = \langle ij | e^{-i\tau\hat{h}_{n,n+1}} | kl \rangle$ . The nonzero elements are:

$$\langle \downarrow\downarrow | e^{-i\tau\hat{h}_{n,n+1}} | \downarrow\downarrow \rangle = \langle \uparrow\uparrow | e^{-i\tau\hat{h}_{n,n+1}} | \uparrow\uparrow \rangle = e^{-i\tau\frac{J_z}{4}}$$

$$\langle \downarrow\uparrow | e^{-i\tau\hat{h}_{n,n+1}} | \downarrow\uparrow \rangle = \langle \uparrow\downarrow | e^{-i\tau\hat{h}_{n,n+1}} | \uparrow\downarrow \rangle = e^{i\tau\frac{J_z}{4}} \cos\left(\frac{\tau}{2}|J_{xy}|\right)$$

$$\langle \uparrow\uparrow | e^{-i\tau\hat{h}_{n,n+1}} | \uparrow\downarrow \rangle = \langle \uparrow\downarrow | e^{-i\tau\hat{h}_{n,n+1}} | \uparrow\uparrow \rangle = e^{i\tau\frac{J_z}{4}} (-i) \sin\left(\frac{\tau}{2}|J_{xy}|\right)$$

### Theta

When updating the Gamma-matrices, you can save some computational time by only treating those combinations of  $\Gamma^{[n]k}$  and  $\Gamma^{[n+1]l}$ , for which  $u_{kl}^{ij}$  is nonzero. The actual singular value decomposition can be done by using the Matlab-routine *svd*, for which the specification of *'econ'* avoids unnecessarily large matrices.

- The SVD provides  $2\chi$  values for  $\lambda^{[n]}$ . Keep only those larger than a threshold  $\varepsilon \simeq 10^{-10}$ , but maximally  $\chi_{max}$  of them, where  $\chi_{max}$  is the maximum matrix dimension you want to employ. You can either work with a fixed matrix dimension, or you can let your matrices grow up to some maximum. The values of  $\varepsilon$  and  $\chi_{max}$  are parameters of your program.

### mitmessen, pro Zeitschritt

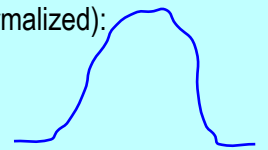
Die Summe der Quadrate der weggelassenen  $\lambda^{[n]}$  Werte kann hierbei zur Kontrolle der Genauigkeit der Simulation verwendet werden. Wird eine vorgeschriebene Schranke überschritten (z.B.  $10^{-8}$ ), wiederholen Sie die Simulation mit einem größeren Wert für  $\chi$ . Um den Fehler aufgrund der Trotter-Suzuki Zerlegung klein zu halten, sollte für  $\tau$  ein Wert von etwa 0.01 oder kleiner verwendet werden (0.05 oder kleiner bei Trotter-Suzuki-Zerlegung 2. Ordnung). Die Signale für gebundene Zustände werden bei genügend kleinen  $\tau$  deutlicher. Eine genaue Beschreibung des TEBD-Algorithmus kann auch in der DA Ganahl S.55, 57, 61-63 und in der DA Zauner S.45-50 gefunden werden.

- Da  $\tau$  in der Regel wesentlich kleiner als die Simulationszeit sein wird, ist es günstig, nicht nach jedem Zeitschritt eine Messung durchzuführen. Details zur Berechnung von Erwartungswerten können auch in der DA Zauner S.30-32 nachgelesen werden.

### Optional: single particle with Gaussian envelope around site $x_0$ , and around momentum $k_0$

We want a linear superposition of states  $|x\rangle$  with a single particle at position  $x$ , with coefficients which have a Gaussian shape around  $x_0$  in real space, and Gaussian shape around  $k_0$  (with width  $1/\sigma$ ) in  $k$ -space, i.e. (not normalized):

$$|psi\rangle = \sum_x \exp(-(\mathbf{x}-\mathbf{x}_0)^2 / (2 \sigma^2)) \exp(i(\mathbf{x}-\mathbf{x}_0) \mathbf{k}_0) |x\rangle$$



#### Parameters:

Take (e.g.)  $\sigma=4$ , so that the Gaussian peak is narrow in  $x$ -space as well as in  $k$ -space.

You can restrict the  $x$ -summation to a range  $x_0$  plusminus  $2 \sigma$ .

Take  $k_0 = \pi/2$  (or  $-\pi/2$ ), then the particle will propagate approximately with a velocity of  $v(\pi/2) = J_{xy}$

#### Expression as a normalized MPS:

For MPS, the easiest way to code this is to use a separate value of the matrix index for each position  $x$ , see example on page 3a (new) of the lecture notes.

Unless the normalization coefficients are included, the matrices then have to be *normalized*:

see Schollwöck review 2011, p.129, commented version).

During the right-left sweep of that procedure, the lambda-matrices are obtained and the matrices can be truncated. Then the Gamma-matrices can be calculated from  $A = \lambda \Gamma$  (lecture notes eq(4.6)).

## Optional (Anregungen):

Wenn Sie möchten, können Sie auch noch einen oder mehrere der folgenden Punkte behandeln:

- Berechnung und graphische Darstellung der **Entanglement Entropy**  $S_{i,i+1}(t)$ .

- Implementierung des verschränkten **1-Spin** Anfangszustands  $|\psi\rangle = \frac{1}{\sqrt{2}} (c_{j-1}^\dagger + \eta c_j^\dagger) |0\rangle$  als Startkonfiguration, mit  $\eta = e^{i\phi}$ . Dieser Zustand ist in der Teilchen-Sprache geschrieben, um eine weiter unten beschriebene analytische Rechnung einfacher zu machen. Es ist  $|0\rangle$  der leere Zustand ohne Teilchen (entsprechend  $s_i = \downarrow$  für alle Spins), und  $c_i^\dagger$  erzeugt ein Teilchen (bzw. einen up-Spin) am Platz  $i$ . Auch für diese Linearkombination lässt sich eine kompakte MPS-Entwicklung angeben. In kanonischer Form lauten die Matrizen

(j fest, Mitte des Systems)

$$i=j \quad \Gamma^{[i-1]\uparrow} = \begin{pmatrix} 1 & 0 \end{pmatrix}, \quad \Gamma^{[i-1]\downarrow} = \begin{pmatrix} 0 & 1 \end{pmatrix}, \quad \Gamma^{[i]\uparrow} = \begin{pmatrix} 0 \\ \eta \end{pmatrix}, \quad \Gamma^{[i]\downarrow} = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

Angegeben wieder: linke ober Ecke einer größeren chi x chi Matrix

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

$$\lambda^{[i-1]} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Die Matrizen für die restlichen Plätze haben wieder die gleiche Form wie schon bei den reinen Produktzuständen. Wie man sich leicht vergewissern kann, erhält man bei  $\eta = -1$  und durch Zusammenfassen von  $\lambda^{[i-1]}\Gamma^{[i]k}$  gerade die am Beginn der Vorlesung vorgestellten  $A$ -Matrizen für den Fall des Singulett-Zustandes.

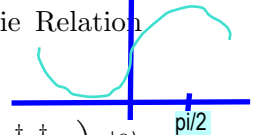
- Simulieren Sie auch hier die Zeitentwicklung, mit  $\eta = 1, -1, i$  und  $-i$ . Was geschieht, wenn der Imaginärteil von  $\eta$  ungleich Null ist?

- Interessierte können für den Anfangszustand analytisch den Erwartungswert von  $\langle \hat{n}_k \rangle = \langle \psi | c_k^\dagger c_k | \psi \rangle$  berechnen und damit Rückschlüsse auf die Impulsverteilung sowie mit Kenntnis der Dispersionsrelation  $\epsilon_k = -J_{xy} \cos(k)$  Rückschlüsse auf die auftretende Geschwindigkeitsverteilung ziehen. Hilfreich hierfür ist die Relation  $c_j^\dagger = \frac{1}{\sqrt{L}} \sum_k c_k^\dagger e^{-ikj}$ .

$$\langle 0 | c_{j-1}^\dagger | 0 \rangle = 0$$

$$\langle 0 | | 0 \rangle = 1$$

Ergebnis: prop. zu  $1 + \text{Re}(\eta \exp(ik))$ . Wenn  $\eta = -i \rightarrow 1 + \sin k$



$$\langle 0 | c_i^\dagger c_{j-1}^\dagger | 0 \rangle = 1$$

- Implementierung der verschränkten **2-Spin** Anregung  $|\psi\rangle = \frac{1}{\sqrt{2}} (c_{i-1}^\dagger c_i^\dagger + \eta c_i^\dagger c_{i+1}^\dagger) |0\rangle$  wiederum mit  $\eta = 1, -1, i, -i$ . Die Matrizen hierfür sind

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}$$

$$\Gamma^{[i-1]\uparrow} = \begin{pmatrix} 1 & 0 \end{pmatrix}, \quad \Gamma^{[i-1]\downarrow} = \begin{pmatrix} 0 & 1 \end{pmatrix}, \quad \Gamma^{[i]\uparrow} = \sqrt{2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \Gamma^{[i]\downarrow} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix},$$

Diese Matrizen sind schon richtig normiert vgl. Skript S.12, 13.

$$\Gamma^{[i+1]\uparrow} = \begin{pmatrix} 0 \\ \eta \end{pmatrix}, \quad \Gamma^{[i+1]\downarrow} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \lambda^{[i]} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \text{lambda bei } i-1$$

Auch in diesem Fall kann man sich leicht davon überzeugen, dass man analoge  $A$ -Matrizen wie für das „nichtlokale Singulett“ am Anfang der Vorlesung erhält, wenn man  $\lambda^{[i-1]}\Gamma^{[i]k}$  und  $\lambda^{[i]}\Gamma^{[i+1]k}$  bildet.

- Sie können im Fall von wenigen Up-Spins (=wenigen "Teilchen") die Zeitentwicklung auch *exakt* berechnen, indem Sie die Hamiltonmatrix im gesamten Vielteilchen-Zustandsraum aufstellen, welcher z.B. bei 2 Up-Spins und  $L$  Gitterplätzen nur die Dimension  $L(L-1)/2$  hat, und danach  $\text{expm}(-iHt)$  auf den Anfangszustand anwenden.