UNIVERSITEIT GENT

Master Engineering Physics

Year 2020-2021

Master desertation

Thesis Tensor networks

October 2020

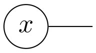
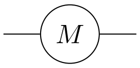
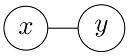
David Devoogdt

Academic supervisor: prof.

Contents

Todo list

Table 1: Caption

conventional	Einstein	tensor notation
$\vec{x}$	x_α	
M	$M_{\alpha\beta}$	
$\vec{x}\cdot\vec{y}$	$x_\alpha y_\alpha$	

1 Introduction

write this

To deal with these strongly correlated systems, a class of methods called tensor networs were introced.

The Hilbert space grows exponentially with system size.

Want to be able to model various Hamiltonians (e.g. Bosons and Fermions). Want to avoid the sign problem of quantum Monte Carlo. Want to measure physical properties and observables

1.1 Quantum monte carlo

2 tensor networks

2.1 introduction

2.1.1 graphical notation

Tensor networks can be written in a graphical notation. The legs f a tensor denote the number of external indices. The upper Connected legs are summed. Some examples are shown in table 1

2.2 Classification

2.2.0.1 MPS A general quantum state with n sites can be described in a given basis $|i\rangle$ as

$$|\Psi\rangle = \sum_{i_1 i_2 \cdots i_n} C^{i_1 i_2 \cdots i_n} |i_1\rangle \otimes |i_2\rangle \otimes \cdots \otimes |i_n\rangle \tag{1}$$

This requires an exponential number d^n of coefficients C where d is the dimensions of basis $|i\rangle$.

In order to make the problem tractable, the following form is proposed as wavefunction:

$$C^{i_1 i_2 \cdots i_n} = C^{1^{i_1}}_{\alpha_1} C^{2^{i_2}}_{\alpha_1 \alpha_2} \cdots C^{n^{i_n}}_{\alpha_{n-1}} \tag{2}$$

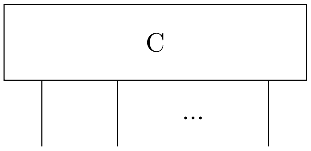
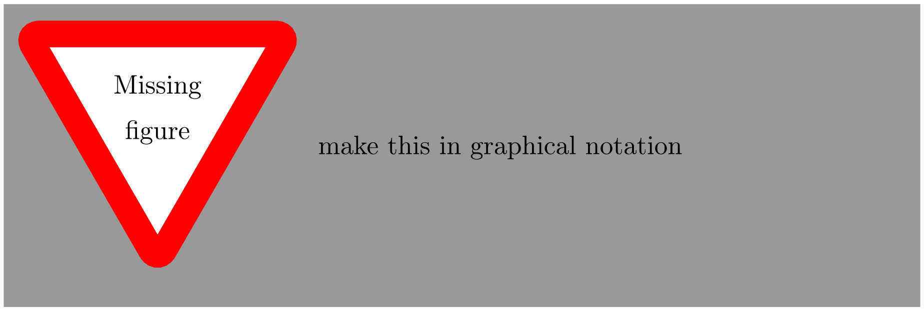


Figure 1: Caption

Where summation over shared indices is implied. It is always possible to find such an representation by means of matrix decomposition. The summation over α_i are called virtual bond and their dimension is denoted by χ.

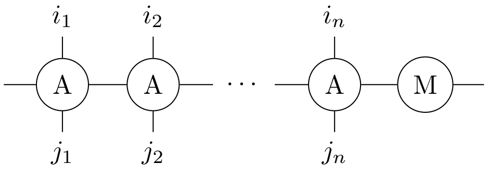
Explicit translational invariance is given by tensor $C^i_{\alpha\beta}$ that don't depend on the location. The chain is closed by setting $\alpha_n = \alpha_0$. We can now write this as a Trace over matrix products:

$$|\Psi\rangle = \mathrm{Tr}\big(C^{i_1}C^{i_2}\cdots C^{i_n}\big)\,|i_1\rangle \otimes |i_2\rangle \otimes \cdots \otimes |i_n\rangle \tag{3}$$



2.2.0.2 MPO In a similar fashion, a Matrix Product Operator (MPO) is of the following form:

$$\begin{aligned}\hat{O} = \sum Tr(A^{i_1j_1}A^{i_2j_2}\cdots A^{i_nj_n}M)\\ \times |i_1\rangle\langle j_1| \otimes |i_2\rangle\langle j_2| \otimes \cdots \otimes |i_n\rangle\langle j_n|\end{aligned} \tag{4}$$



connect trace and hide legs at M

The matrix M contains the boundary conditions of the operator. Many Hamiltonians can be represented by an MPO. For ins

2.2.0.3 PEPS Exact contraction is hashtag P-Hard:

No exact canonical form

2.2.0.4 PEPO

2.2.0.5 Others MERA,TTN,

3 Operator exponentials

While it is often possible to find exact MPO representation to represent a wide class of hamiltonians , it is much harder to do the same for exponentiated operators. These operators play an important role: they act as time evolution operators for quantum systems $|\Psi(t)\rangle = \exp\left(-i\hat{H}t\right)|\Psi(0)\rangle$. A very similar operator governs the partition function in statistical mechanics: the probability of finding a system at inverse temperature $\beta = \frac{1}{T}$ in a microstate i is given by $p_i \exp\left\{-\beta \hat{H}_i\right\}$. This is often called "imaginary" time, due to the substitution $\beta = it$. . The ability to calculate these operators is essential for understanding the dynamics of a given quantum model, and making contact with real world obsevations of these systems at finite temperature.

find citation and examples

explain link CFT

3.1 Statistical mechanics

The physics of a system in thermodynamical equilibrium can be derived from it's partition function Z. The classical formula generalises to a density matrix ρ as follows:

$$\begin{aligned} Z &= \sum e^{-\beta E_n} \\ &= \sum_n \left\langle n \middle| e^{-\beta \hat{H}} \middle| n \right\rangle \\ &= \mathrm{Tr}\left(e^{-\beta \hat{H}}\right) \end{aligned} \tag{5}$$

The first line is the partition function for clasical discrete systems. The index n runs of all possible microstates. It is known that the propability to find the system in a given microstates is given by:

$$p_i = \frac{\sum e^{-\beta E_i}}{Z} \tag{6}$$

An useful quantity is the density matrix ρ.

$$\begin{aligned} \rho &= \sum_j p_i \left|\Psi_j\right\rangle \left\langle \Psi_j\right| \\ &= \sum_j \frac{e^{-\beta \hat{H}}}{Z} \left|\Psi_j\right\rangle \left\langle \Psi_j\right| \end{aligned} \tag{7}$$

Whith this notation, the partition function Z and ensemble average of an operator $\hat{X}$ are given by:

$$\begin{aligned} Z &= \mathrm{Tr}(\rho) \\ \langle X \rangle &= \mathrm{Tr}\left(\rho \hat{X}\right) \end{aligned} \tag{8}$$

3.2 Time evolutions

In quantum field theory, , calculation of n-point correlation functions is extremely important to understand a given field theory.

LSZ theorema herbekijken

3.2.1 ground state

One practical way of finding the ground state is cooling an intitial stata down very small T.

3.3 Tensor network methods

In the following section I will give a very short review of the current tensor network methods to simulate real or imaginary time evolution. This overview is mainly based on the review paper [1].

[1]

3.3.1 Approximations to $\hat{U}(\delta)$

TEBD, MPO $W^{I,II}$

3.3.2 global Krylov method

3.3.3 MPS-local methods

local Krylov TDVP

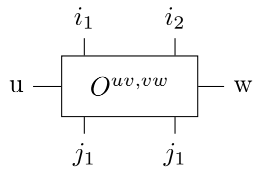
4 Tensor network manipulations

This section severs as an introduction of tensor network manipulations. The overview mainly focusses on MPS/MPO networks, but most of the oprations translate to the 2D case.

The manipulations of MPO's is done by manipulating the tensor into a matrix, performing some matrix calculations and casting it back into it's original form. This section gives some examples how these manipulations are done in practice:

4.1 Basics

4.1.1 decomposition

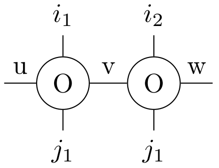


$$= O^{i_1 i_2 j_1 j_2}_{\alpha_u \gamma_w}$$

$$\cong O^{uw}_{(\alpha_u i_1 j_1)(\gamma_w i_2 j_2)}$$

$$= O^{uv}_{(\alpha_u i_1 j_1)\beta_v} O^{vw}_{\beta_v (\gamma_w i_2 j_2)} \tag{9}$$

$$\cong$$



Step 2 reshapes and groups the indices to one index. The dimension of this index is the sum of the seperate dimensions. Step 3 decomposes the matrix into a product of 2 matrices. The last step transforms the indices back to separate legs.

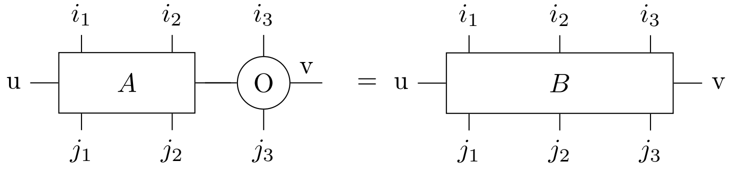
For an exact representation, the bond dimension of virtual level v is:

$$\dim v = \min(\dim u, \dim v) + 2 \dim i \tag{10}$$

Many different matrix decompositions exist. Some usefull examples here are svd decomposition and eigenvalue decomposition. The split in left and right matrix is not unique and differs per algoritm.

4.1.2 inverse

Suppose we want to find a MPO O for given tensors A and B such that the following holds:



(11)

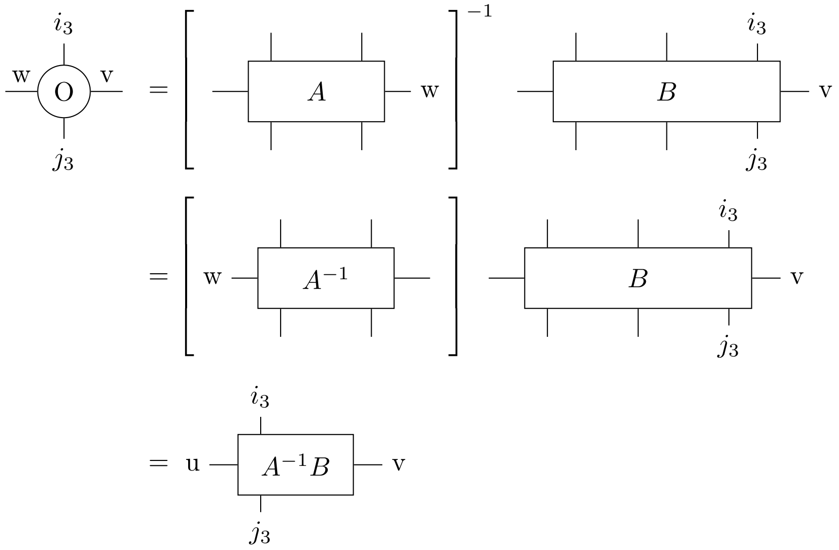
Again, the indices can be taken toghether in the following way: $\alpha = (u i_1 j_1 i_2 j_2)$ and $\beta = (i_3 j_3 v)$:

$$A_{\alpha\gamma} O_{\gamma\beta} = B_{\alpha\beta} \tag{12}$$

This a a standard matrix equation and can hence be solved with linear algebra packages. Note that it is not necessary to calculate A^{-1} to obtain the solution.

In matlab the procedures mldivide and lsqminnorm can be used, which give more robust solutions.

The solution will be denoted by



(13)

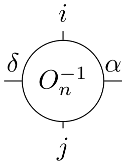
For the first equation the unmarked legs on the same positions need to be contracted with each other. The second line the mirrored positions are contracted.

If the tensor A is an MPO, the inverse can also be constructed as an mpo. This is espacially usefull if the MPO is created with decomposition for which the inverse can be computed easily, suchs as an svd decomposition.

Take has to be taken with the indices to apply the inverse.

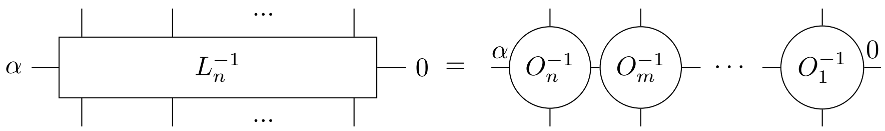
$$\begin{aligned} U^n_{(\alpha ij)\beta} A_{\beta\gamma} &= B_{\alpha ij\gamma} \\ A_{\delta\gamma} &= U^{n\dagger}_{\delta(\alpha ij)} B_{\alpha ij\gamma} \end{aligned} \tag{14}$$

If we now define the MPO O_n^{-1} equal to $U^{n\dagger}$ with the second index split and permuted:



$$\cong U^{n\dagger}_{\delta ij\alpha} \tag{15}$$

With the notation from eq. (13) we have:



(16)

The inverse can be applied sequentially.

PEPS inverse, fast version, bookkeeping

4.1.3 virtual levels and matrisation

4.1.4 truncation

explain

From the construction with svd we can see that the dimension of virtual bond dim $n = d^{2n}$ with d the dimension of $|i\rangle$. The virtual levels can be joined into a $\chi \times d \times d \times \chi$ dimensional tensor O. This tensor is given by a tridiagonal block matrix :

$$O^{ij}_{\alpha\beta} = \begin{bmatrix} O^{00,ij} & O^{01,ij} & & \\ O^{10,ij} & O^{11,ij} & O^{12,ij} & \\ & O^{21,ij} & O^{22,ij} & \ddots \\ & & \ddots & \ddots \end{bmatrix} \tag{17}$$

The boundary conditions (leftmost and rightmost virtual level are zero) correspond to vectors:

$$\begin{aligned} l &= \begin{bmatrix} 1 & 0 & \cdots \end{bmatrix} \\ r &= l^T \end{aligned} \tag{18}$$

The total dimension is the sum of dimensions of the virtual level. In this case the

berken exact en maak tabletje voor de verschillende types

4.1.5 contraction order

4.1.6 Gauge freedom

4.2 MPS algoritms

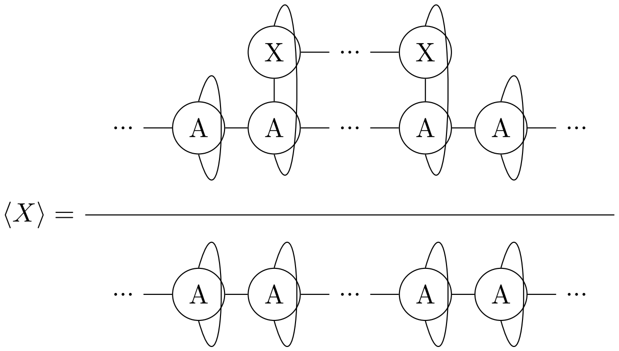
4.2.1 cononical form

schmidt decomp,

4.2.2 DMRG

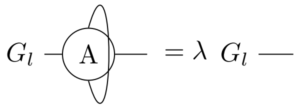
4.2.3 Expectation values

Suppose that the there is an MPO representation of $e^{-\beta \hat{H}}$ A and that the mpo represenation for X Y is localised over n sites, then the expactation value is given by:

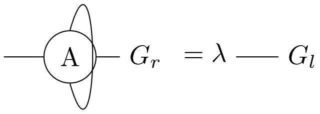


(19)

In the thermodynamic limit there are an infinity number of A to the left and the right. This can be simulated by taking the left and right fixed points of the traced MPO A corresponding to the largest eigenvector λ.

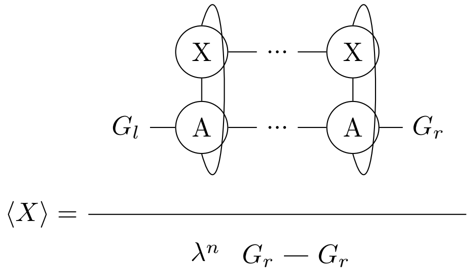


(20)



(21)

Equation eq. (19) can now be easily caclulated:



(22)

4.3 PEPS algorimts

4.3.1 compression

4.3.1.1 CTM

4.3.1.2 Vumps

5 Contruction MPO

5.1 Time evolution methods

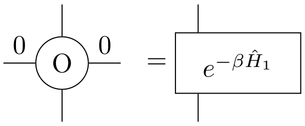
https://tensornetwork.org/

5.2 Cluster expansion

This thesis builds on the cluster expansions introduced in [2]. The idea is to create tensor network with a number of virtual levels. The representation is exact up to M connected sites, where M is the order. Different variations are possible. A Hamiltonian of the folowing form is assumed

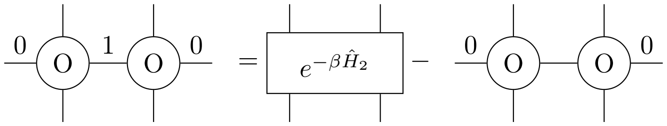
$$\hat{H}_n = \sum_{i=1}^{n-1} \hat{h}_{i,i+1} + \sum_{i=1}^{n} \hat{h}'_i \tag{23}$$

Virtual level zero is defined as follows:



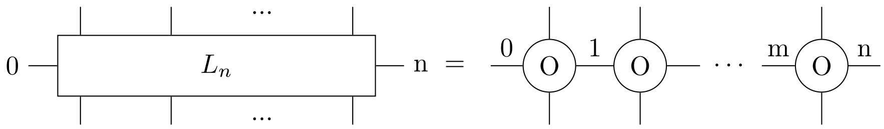
(24)

Similarly, the contraction of elements O_{01} and O_{10} are defined as follows:



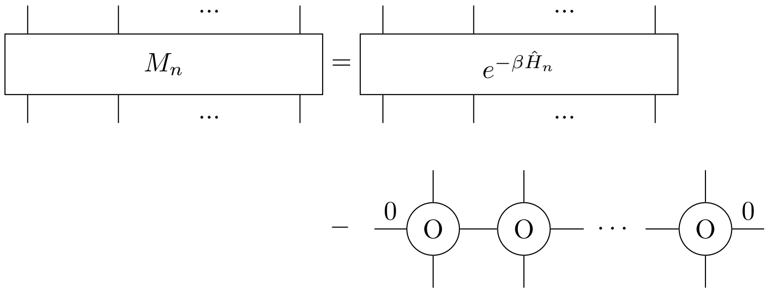
(25)

Some notation will be introduced that will be used later on. The rensor L_n is the contraction of n MPO's where the virtual index increases between each bond. R_n is similar but the virtual bond starts from n and decreases. The tensor is defined in gra[hical notation by:



(26)

M_n is the difference between the exponentiated hamiltonian for n sites and the contraction of the MPO over all the currently assigned combinations of virtual levels.

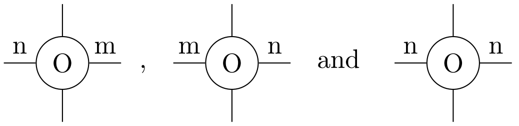


(27)

In what follows different constructions will detailed. Some notes on the practical implementation is given later in this section.

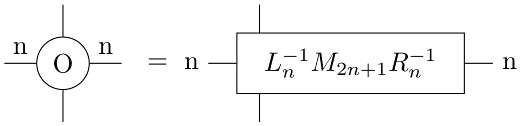
5.2.1 Type A

This type was originally proposed in [2]. The following types of blocks appear in the cluster expansion:



with $n \in \mathbb{N}_0$ and $m = n - 1$.

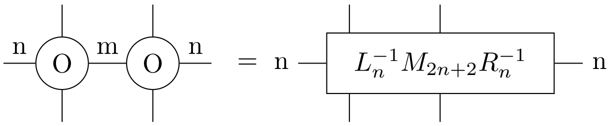
5.2.1.1 O^{nn} The O^{nn} block is defined by eq. (28)



(28)

The residual error M is calculated for a chain of size $2n + 1$. The left and right inverses L_n and R_n are applied to M to find the block O^{nn}.

5.2.1.2 O^{mn} **and** O^{nm} The contraction of O^{nm} and O^{mn} is defined by:



(29)

The individual elements O^{mn} and O^{nm} are obtained by doing an svd decoposition as explained in section 4.1.1. The square root of the singuar values is multiplied to the left and right of the decomposition. The operator L_{n+1} and R_{n+1} can be construdted directly.

implementeren en testen.

During the svd the bond dimension can be lowered by only keeping the rows and colums belonging to $\sigma_i > \sigma_0$. This also helps the invertibility. Increasing σ_0 reduces the precision of the MPO.

5.2.1.3 Truncation Intoduction of a new block can result in large fluctuating errors. This happens because the inverses are possible ill conditioned. Therefore the construction of the MPO should be stopped at a certain optimal order. Many different criteria can be tought of (and have been tried), but the most reliable method is as follows:

5.2.1.4 O^{nn} The O^{nn} blocks can form long chains. To test whether these chains improve accuracy, the norm (see section 9.1.1) of the residul error is calculated before and after the insertion of the block for a cyclic chain. A closed chain is used with the same number of sites. The closed chain resembles much better an infite chain than the open counterpart.

Only strict improvements are accepted, otherwise the MPO without the newly calculated blocks is returned.

5.2.1.5 O^{mn} and O^{nm} For these the same procedure holds.

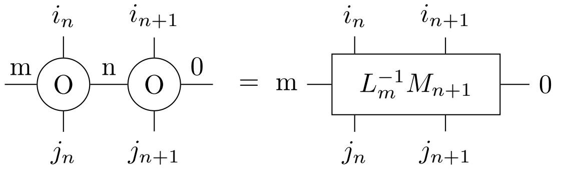
5.2.1.6 dimension From the construction is can be seen that the dimension of the new virtual level is at most d^2 times the dimension of the previous level. Depending on σ_0, the bond dimension is even lower.

5.2.1.7 discussion The only parameter in the construction is σ_0 As can be seen in fig. 2, mainly the construction for small values of β get affected by the choice of σ_0.This can be seen in fig. 2. A good tradeoff seems to be $\sigma_0 = 10^{-12}$. There is almost no precision loss vor small β, while for intermediate it performs optimal for intermediate β.

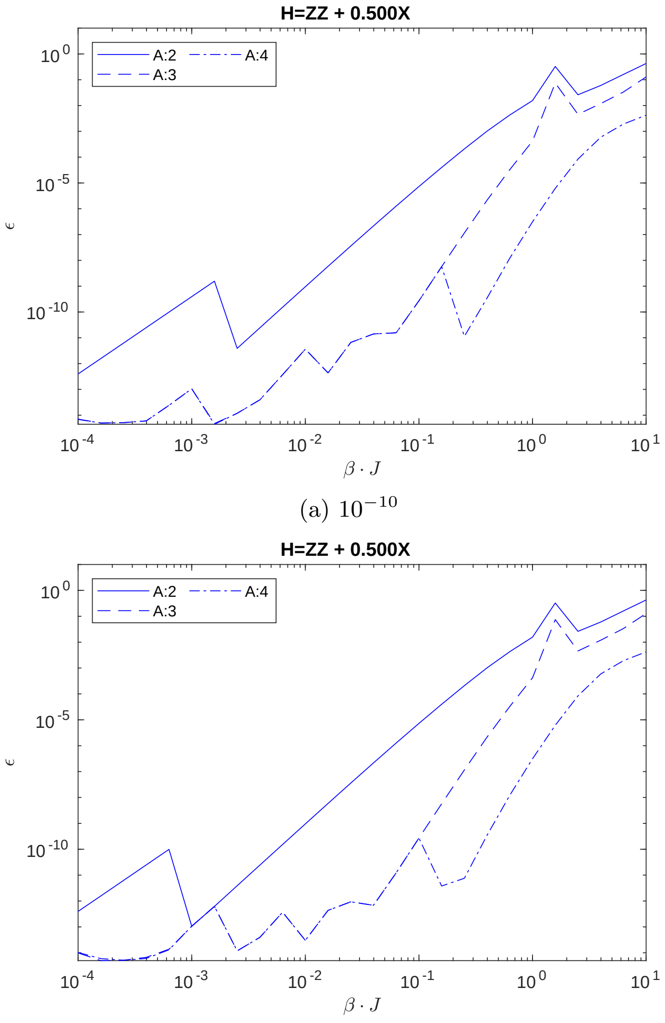
In practice, the truncation criterium is not completely optimal in some specific cases.

5.2.2 Type B

Type B only contains blocks of the following form; O^{mn} and O^{n0}



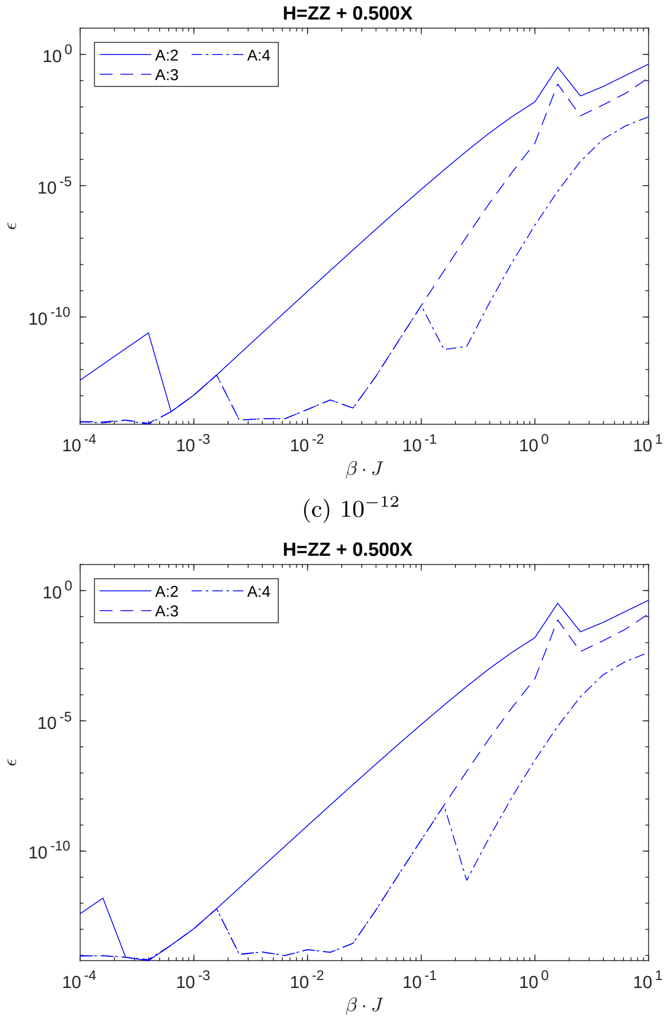
$$\begin{aligned} &\cong X_{(\alpha_m i_n j_n)(i_{n+1} j_{n+1})} \\ &= U^n \Sigma V^\dagger \end{aligned} \tag{30}$$



(a) 10^{-10}

(b) 10^{-11}

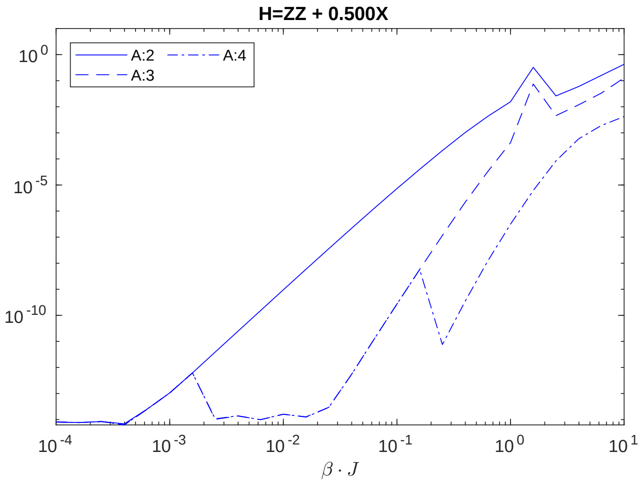
Figure 2: A figure with two subfigures



(c) 10^{-12}

(d) 10^{-13}

Figure 2: A figure with two subfigures



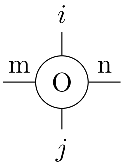
(e) 10^{-14}

Figure 2: A figure with two subfigures

The following split is made: $O^{mn} \cong U^n$ and $O^{n0} \cong \Sigma V^\dagger$. In this way the inverse exists and doesn't need any calculation: $O^{mn} = U^\dagger$.

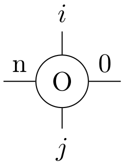
implement this!

5.2.2.1 dimension From the construction the bond dimension grows from the left to the right. For the last step, there are only d^2 non zero singular values. Each steps adds d^2 to the dimension. For the last step, only d^2 non zero singular values need to be keeped. With the following natation:



$$= A^m_{(\alpha i j)\beta}$$

(31)



$$= B^n_{(\alpha i j)\beta}$$

The bond dimension of lower virtual levels can be reduced if we can solve the following equations simultaneously:

Then the MPO doesn't change if there are matrices A'^n, A'^{n+1} and B'^n such that

$$\begin{aligned} S = A^m A^n = A'^m A'^n \\ T = A^m B^n = A'^m B'^n \end{aligned} \tag{32}$$

Such matrices with optimal bond dimension can be found with generalised SVD. Generalised SVD decomposes 2 matrices as follows:

$$\begin{aligned} S^\dagger &= (U\Sigma_1)Q^\dagger \\ T^\dagger &= (V\Sigma_2)Q^\dagger \end{aligned} \tag{33}$$

The new bond dimension is the $\dim n' = d^2 \cdot \min(\dim n - 1, \dim(n+1))$. The dimension is higher than type A.

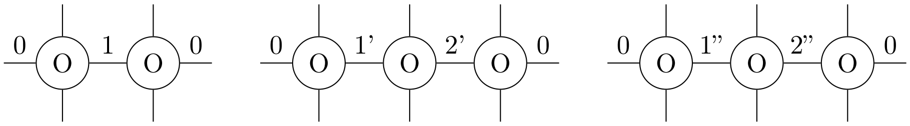
meer
uitleg gsvd
https://nl.mathworks.com/

5.2.2.2 discussion This type has no parameters to fine tune. The only blocks appearing in this series expansion are explicitly calculated to lower the remaining error. The inverse is well defined. It is expected that the error increases strictly with increasing temperature and keeps decreasing for higher orders.

5.2.3 Type C

primed virtual levels

This type implements the same strict type as Type B, blubut in a different way. No calculation is involved, except the calculation of the the exponentiated hamiltonian to certain order. The following kind of MPO strings are allowed:



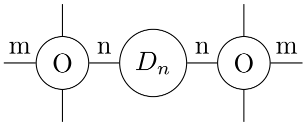
and so forth. All but one MPO elements are chosen to be the identity matrix. The middle one is the exponentiated hamiltonian with reshaped legs.

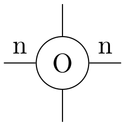
Primed levels are in fact just non overlapping virtual levels. They can be mapped to normal virtual levels.

5.2.3.1 discussion As can be expected from the construction, the bond dimension grows very fast. This type is just as precise as Type B.

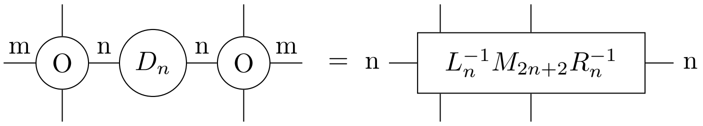
5.2.4 Type D

This type uses a differemt setup which tries to capture the best of both Type A and B. Type could handle long range correlation better because of the introuction of O^{nn}, but the inverse was not necesarily well defined. Type B had well conditioned inverses, but performed in most of the cases worse. The block appearing in type D are as follows:



and


Similar to type A,



(34)

$$= U\Sigma V^\dagger$$

Matrix D_n is the singular value diagonal matrix devided by a normalisation factor ϕ. Both U and V are multiplied by $\sqrt{\phi}$.

5.2.4.1 discussion It's not completely clear what the values of ϕ should be. If ϕ is to large, large chains are not surpressed. If phi is to small, the O^{nn} blocks will become large and hence the chain will diverge again. A reasonable value is the sum of the singular values.

other things could be tried here, WIP

5.2.4.2 matrisation The cost of this type lies in the fact that it has no compact way of casting it to a matrix. The following works, but has quite a large dimension:

$$\begin{array}{ccc|cc|cc|cc}
O_{00} & O_{01} & & -2O_{01} & & O_{01} & & O_{01}D_1^{1/2} & \\
O_{10} & & O_{12} & & -2O_{12} & & O_{12} & & \\
 & O_{21} & & & & & & & \\
\hline
O_{10} & & & & & & & & \\
 & O_{21} & & & & & & & \\
\hline
O_{10} & & & & & O_{11} & & & \\
 & O_{21} & & & & & O_{22} & & \\
\hline
D_1^{1/2}O_{10} & & & & & & & & D_1^{-1/2}O_{12}D_2^{1/2} \\
 & & & & & & & D_2^{1/2}O_{21}D_1^{-1/2} &
\end{array}$$

fix this

5.3 Implementation details

All of the above types are implemented for arbitrary order in Matlab.

5.3.1 Source code structure

The source code for this project can be found on github. The implementation of these types can be found under `src/generateMPO.m`. In this class the different types of MPO can be constructed. It bundles some helper functions such as contracting a chain or cycle of MPO's or construction of an exponentiated hamiltonian for the given input hamiltonian. Other examples are making L_n^{-1} by sequential invers MPO contractions,...

`src/test.m` contains the code to create the plots to compare different types and orders. The other files in the folder are self-explanatory.

5.3.2 internal representation

Two main inernal representations are used to construct the given MPO. Either, the MPO is stored as a cell of matrices, or as one big matrix where the blocks are added to during the construction. The output type can be chosen. For some types, sparse matrices are used during the construction. Given that matlab does't support multidimensional matrices by default, this library is used.

5.3.3 normalisation factor

An overal normalisation factor κ per site is used to keep the numbers in matric within a reasonable range. In practice, $e^{\beta H_n}$ is divided by by κ^n. Otherwise nothing has to be changed to the algorithms. κ can be changed during the construction by absorbing the factor in the matrices.

5.3.4 buffering results

The matrix exponential for different number of sites is called on many different places. The results for chain and cycle are stored in the class to save computing time the next time.

5.3.5 Optimalisations

The construction is not yet fully optimised for speed. The arrays can be cast to Matlab gpuArray's to offload the calculations.

6 Contruction PEPO

The MPO construction outlined in previous section can be generalized to

6.1 Construction block

The construction

6.2 Linear block

6.3 Blocks with loops

6.3.1 hypothesis

6.4 Framework Implementation

A large part of the code written consist of a general framework in Matlab to perform many operations in an automated manner.

6.4.1 Linear solver

The linear solver is a general purpose block solver which reduces the problem to a set of linear matrix equations. Linear block consist of a tree structure, where the new block is the root of the tree, and all the branches need to be inverted. Let $I^m = (i^1_1 i^1_2 \cdots i^1_{n_1})$, then the problem can in general, after some tedious tensor leg bookkeeping, be rewritten in the following form:

$$\begin{aligned} &A_{I_1 I_2 \cdots I_n \alpha^1 \alpha^2 \cdots \alpha^m} X_{\alpha^1 \alpha^2 \cdots \alpha^m j} \\ =&B_{I_1 I_2 \cdots I_n j} \end{aligned} \tag{35}$$

Here i^M_N has the following meaning: M numbers the differrent legs or branches of the tree, N number of sites of the leg and i numbers the bra and ket states and has dimension d^2. Hence the bond dimension of $I_n = d^{2n_m}$. The most obvious way to solve this system is by using a linear solver. The problem is that the bond dimension increases very fast: matrix A has dimension $d^{2\sum_m n_m} \times d^{2\sum_m n_m}$. Although using a linear solver instead of full inversion is considerably faster, this becomes infeasable for very quickly. A second method consist of solving the following sequence of linear problems one leg at a time:

$$\begin{aligned} A^1_{I^1\alpha^1} X_{\alpha^1 I^2 \cdots I^m j} &= B_{I_1 I_2 \cdots I_n j} \\ A^2_{I^2\alpha^2} X_{\alpha^1 \alpha^2 I^3 \cdots I^m j} &= B_{\alpha^1 I_2 \cdots I_n j} \\ &\vdots \\ A^m_{I^m\alpha^m} X_{\alpha^1 \alpha^2 \cdots \alpha^m j} &= B_{\alpha^1 \alpha^2 \cdots \alpha^{m-1} I_m j} \end{aligned} \tag{36}$$

While this method is very quick and scales well, in practice it results in unstable result. This is a result of the potentially ill conditioned inverses inherent to the construction. A pseudo-inverse of the full matrix can be easily obtained and resolves this issue . Solving in a sequetial way, the errors of the pseudo-inverses accumulate. Luckily the problem can be resolved by first performming an SVD decomposition of $A^m = U^m S^m V^{m\dagger}$ matrices, with S diagonal and U and V unitary. All the U^m matrices can be inverted by applying the hermitian transpose to B. The Tensor $S^1 \otimes S^2 \cdots \otimes S^m$ is very sparse and can be inverted at once. The last step consist of inverting all unitary V.

link to right section

The linear solver also works

6.4.2 Nonlinear solver

6.4.2.1 symmetry

6.5 Bookkeeping

6.6 Fast contraction

6.7 Normalisation

7 Models

7.1 Ising model

7.1.1 Classical Ising

The classical ising model is given by the following hamiltonian:

$$H = -J \left(\sum_{<ij>} \sigma_i \sigma_j + g \sum_i \sigma_i \right) \tag{37}$$

where $< ij >$ runs over all neighbouring lattice sites. Classical refers to the fact that all the operators in the hamiltonian commute with each other. The values of σ depends on the spin dimension. For spin 1/2 lattices $\sigma \in -1, +1$.

7.1.1.1 1D Phase Diagram

7.1.1.2 2D Phase Diagram

7.1.2 Quantum Ising

In the quantum Ising model, the operators no longer commute with each other. An example is the transversal ising model given by:

$$\hat{H} = -J \left(\sum_{<ij>} \sigma_i^x \sigma_j^x + g \sum_i \sigma_i^z \right) \tag{38}$$

7.1.2.1 1D Phase Diagram

7.1.2.2 2D Phase Diagram

7.2 Heisenberg

The heisenberg model is given by:

$$\hat{H} = - \left(\sum_{<ij>} J_x \sigma_i^z \sigma_j^z + J_y \sigma_i^y \sigma_j^y + J_z \sigma_i^z \sigma_j^z + h \sum_i \sigma_i^z \right) \tag{39}$$

These models have different names depending on the values of J_α with $\alpha = x, y, z$. $J_x = J_y \neq J_z = \Delta$ is called the XXZ model.

7.3 Random

It's also possible to construct random hamiltonians.

in basis: hermitian H

8 Phases and Criticality

8.1 Phases of matter

An important area of research is the study of the different phases of (quantum) matter. A phase is a state of matter in which the macroscopic physical properties of the substance are uniform on a macroscopic length scale. These phase can be measured by thermodynamic function, i.e. by function of a few macroscopic parameters. [3]. More precisely, for a given phase the properties vary as an analytic function of the macroscopic variables.

Interesting physics happens at the boundary between 2 or more distinc phases. The phase transitions were classified by Ehrenfest [4], who looked at the free energy across the phase boundary. If the free energy shows a discontinuity, it is called first order (or discontinuous) phase transition. Similarly, if the derivitive shows a discontinuity, it is called second order (or continuous). Higher order phase transitions are possible, and there are even examples of infinite order transitions, such as the BKT transition.

8.2 symmetry breaking

Sometimes, but not always, a phase transition is related to spontaneous symmetry breaking. A state $|\Psi\rangle$ is said to be symmetric under a unitary transformation U if the state only changes by a phase factor: $\hat{U}|\Psi\rangle = e^{i\phi}|\Psi\rangle$. A hamiltonian posesses a symmetry if it commutes with U: $[H, U] = 0$ [5]. A remarkable fact is that many ground states are not invariant under a symmetry U of the hamiltonian.

For phase transitions associated with a broken symmetry, one can define an order parameter. This parameter evalutes to 0 for the symmetric phase, but not for the sponteous broken phase.

In continuous or second-order phase transitions the order parameter increases continuously from zero as the critical temperature is traversed. The entropy also changes continuously. On the other hand, the correlation length and related energy scales diverge at the critical temperature. In fact, at the critical temperature of a second-order phase transition, scale invariance systems become scale-invariant, in the sense that physical properties no longer depend on the length (or energy) scale at which they are probed. Many symmetry-breaking phase transitions are second-order, with the onsets of superfluidity, (anti)ferromagnetism and many phases of liquid crystals as famous examples.[5]

8.3 Universality

Universality looks at the behaviour of the system near a continuous phase transition. These can be discribed well by so called power laws. For classical phase transitions (driven by temperature) near critical temperature T_c, observables a_i depend in the following way on the reduced temperature $t = \frac{T-T_c}{T_c}$: $a_i(t) \sim t^{\alpha_i}$. One would expect that the set of critical exponents α_i depends on the precise form of the hamiltonian of the system, but it turns out these exponents can be captured by a limited numer of universality classes. This means that the physics near criticality is completely understood once it is understood for one member of the class.

8.4 critical exponents for spin systems

9 Benchmarking

9.1 dioganalsation

The performance of the MPO construction can be compared with the exact diagonalisation of the hamiltonian for a given number of sites. To obtain a faithful results, the number of sites should be as high as possible. In practice, diagonalisation of large matrices becomes slow and memory consuming. The size grows exponentially in the number of sites: $d^n \times d^n$. A double takes 8 bytes of memory.A Rough estimated of the amount of RAM R needed to store this complex array is:

$$R = d^{2n} \times 16bytes \tag{40}$$

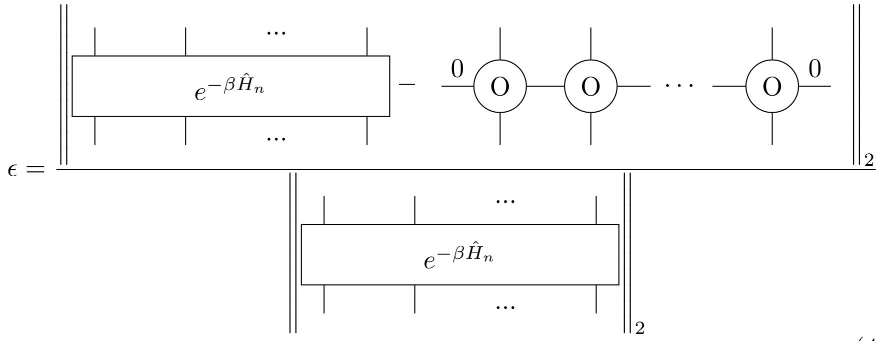
Which means a 14 site chain already takes up GB of RAM.

time complexity algoritms

9.1.1 norms

The schatten 2 norm is used in the following analysis, dentoted by $\| \cdot \|_2$. In the figures the relative error ϵ is reported.

trace norm, schatten p norm, ...



(41)

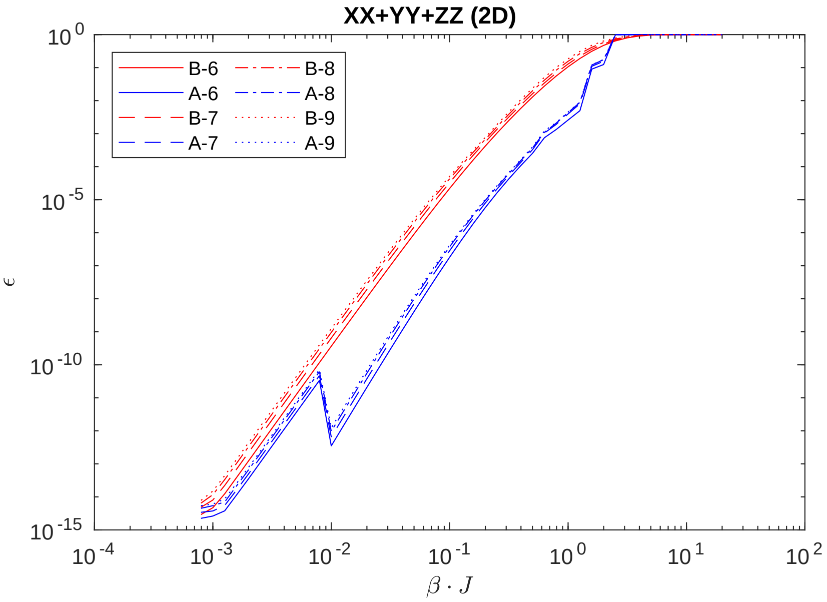
make version for cyclic

9.1.1.1 system size and cyclicity This norm can only be calculated for a finite number of sites. The influence of the number of sites for a linear and cyclic fig. 3 . As expected, the cyclic norm represents large systems better for the same number of sites. The linear norm keeps increasing with every added site.

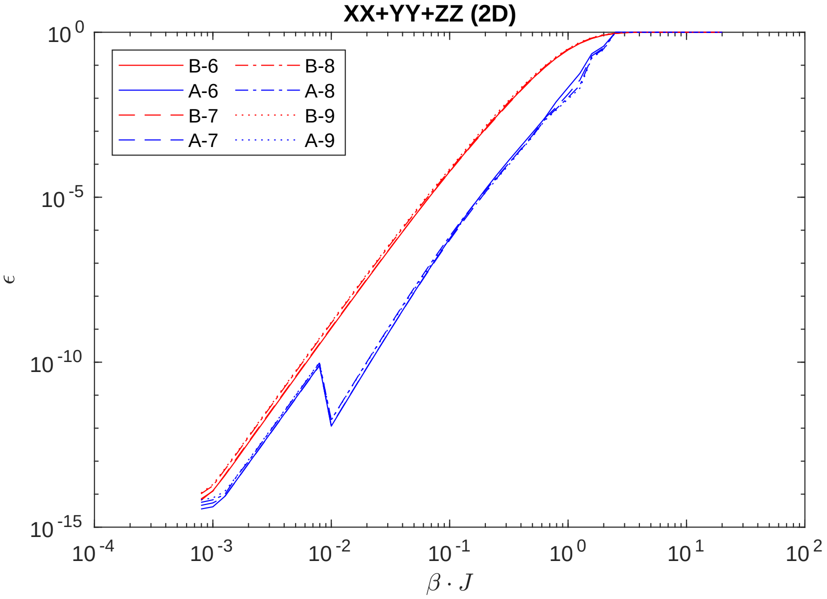
Calculating the cyclic norm comes at the extra cost of contracting a cyclic tensor network.

calculate complexity

In this chapter, the cyclic norm will be given for M=8 sites.



(a) test



(b) test

Figure 3: test

9.1.2 Ising

The first model used to benchmark the different types of MPO's is the transversal ising model. For type A the ϵ increases with β. As expected, the relative error decreases with increasing order.

The behaviour of type B is more chaotic. The error increases no longer monotonously. For small values of β, the order is truncated.

For type 5 fig. 5, there is a consitent improvement over type B.

larger orders need reimplementation (non matrix based)

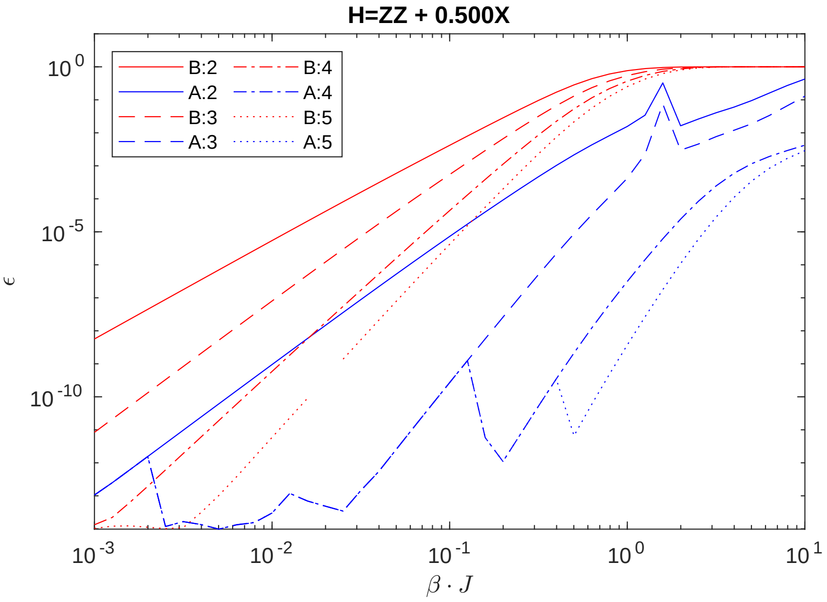


Figure 4: Comparison type A and B for Transversal Ising

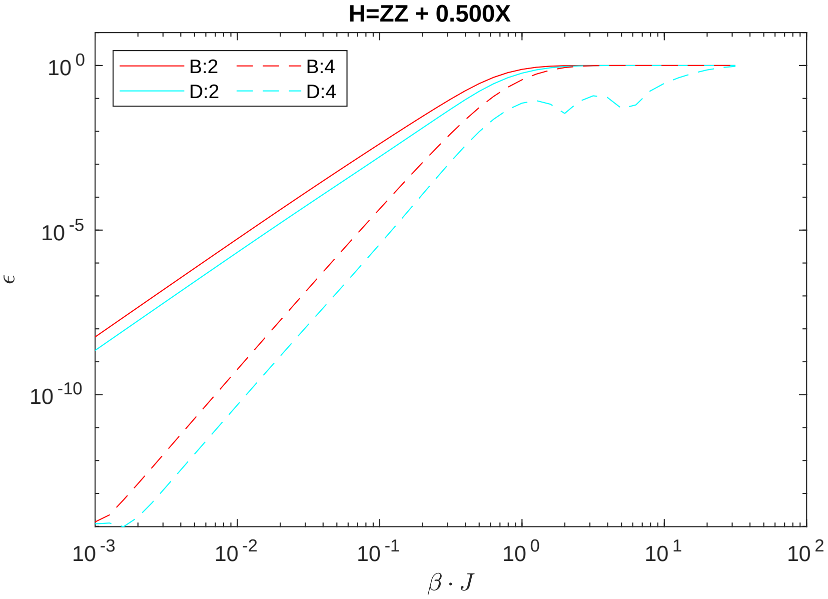


Figure 5: Comparison type C and B for Transversal Ising

9.1.3 Heisenberg

For the Heisenberg model, type A is also an improvement over type B. For large values of β, increasing the order does not help. Type 5 fig. 7 is more promising, but higher orders require to much memory to simulate.

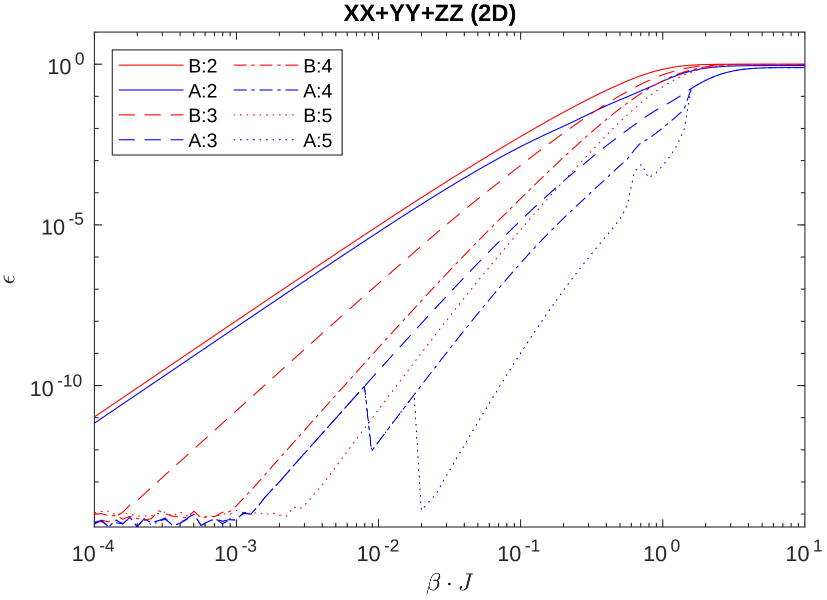


Figure 6: Comparison type A and B for Heisenberg

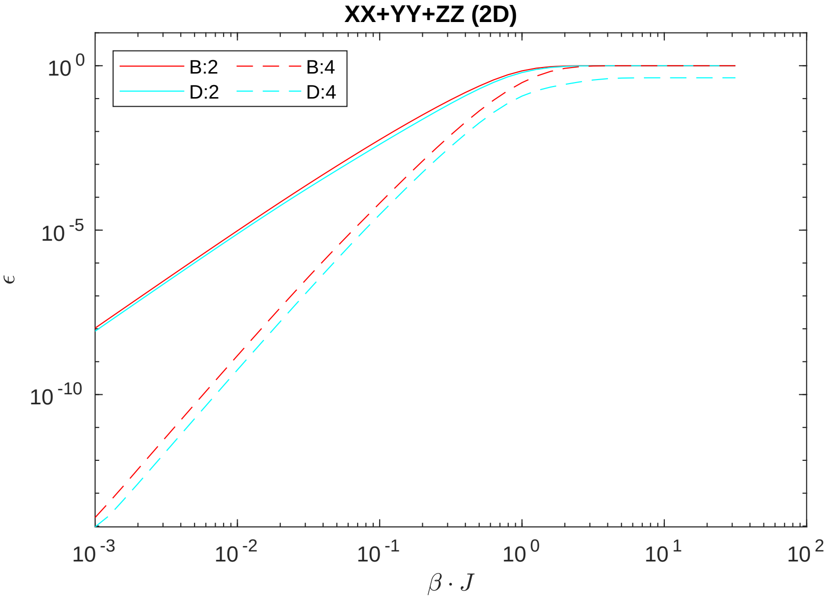


Figure 7: Comparison type C and B for Heisenberg

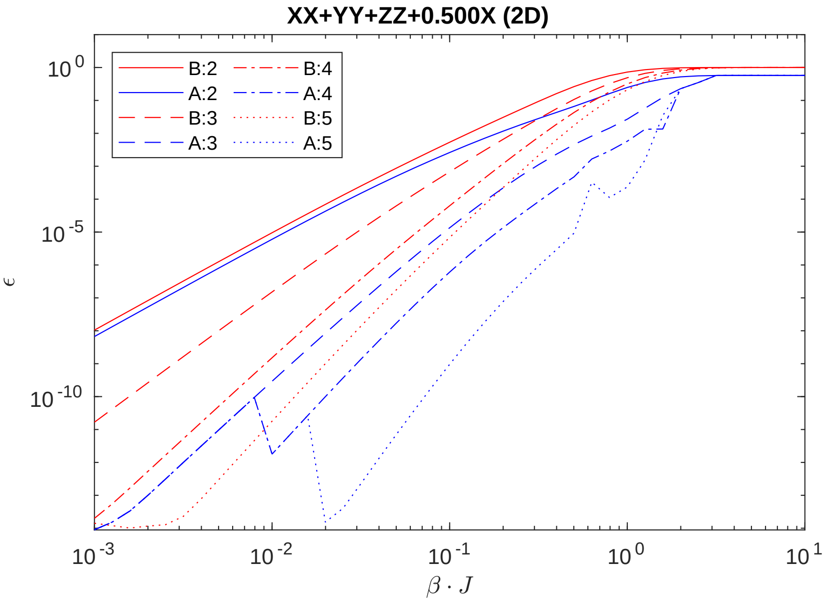


Figure 8: transversal XXX

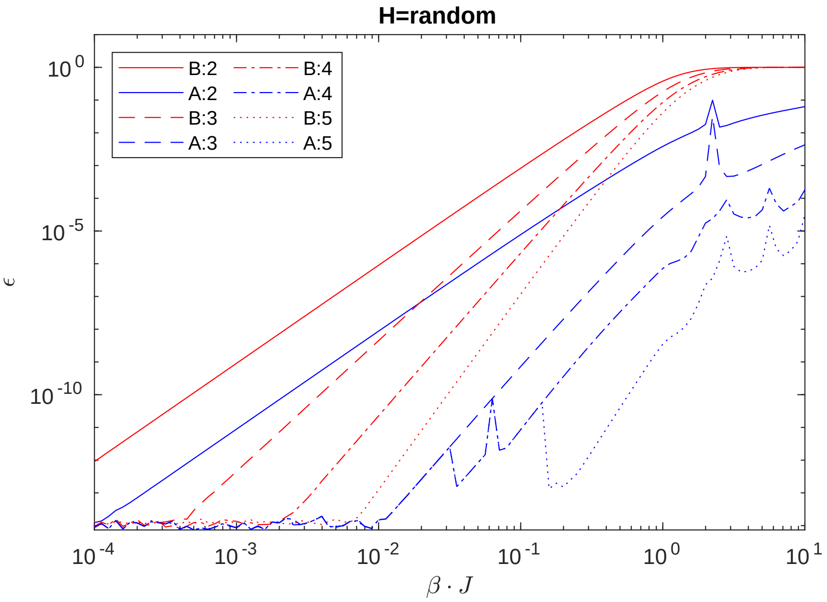
run with M=11

9.1.4 Random

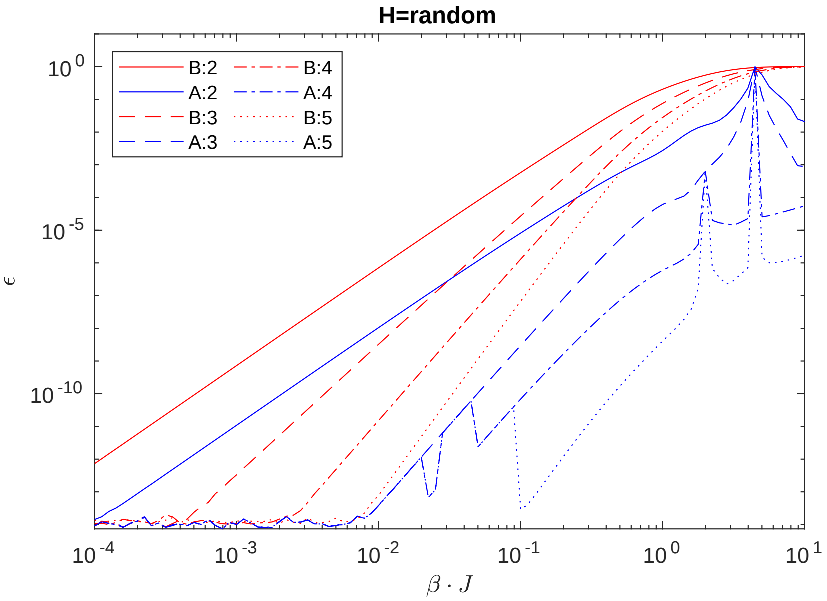
To give a representative overview for random hamiltonians, several simulations were run. The single site and nearest neighbourgh hamiltonians are generated by making hermitian matrices with random real and complex numbers between -1 and 1. In order to compare the different graphs, the engergy scale is set such that the norm of the 2 site hamiltonian is 1.

Clearly, the performance of type B is almost independent on the chosen random variables. For type A there is more variation. Still, A performs almost always better than B. For some random models, such as fig. 9b, the order is truncated at low order for high temperatures (see peak at $\beta J \approxeq 4$). It6s unclear why this behaviour emerges. Manually overinding the sefagaurd machanism

blabla

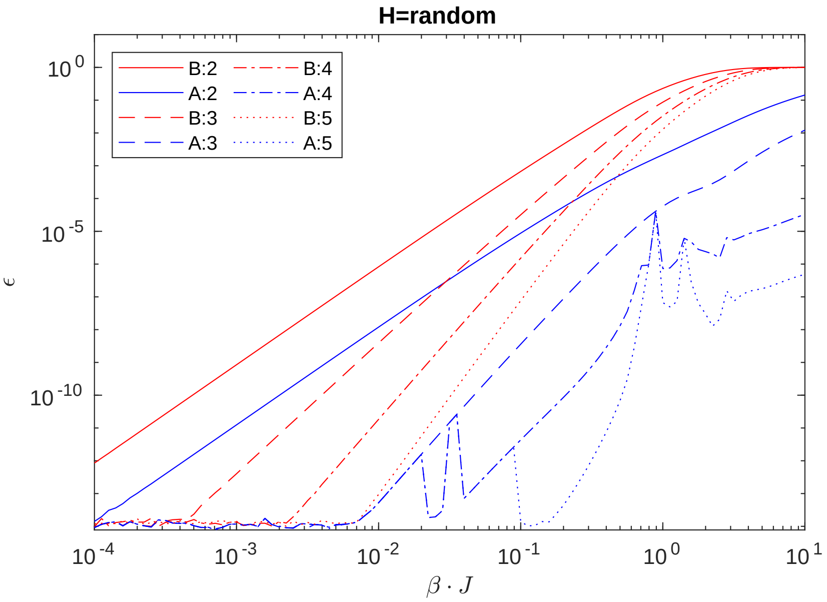


(a) test

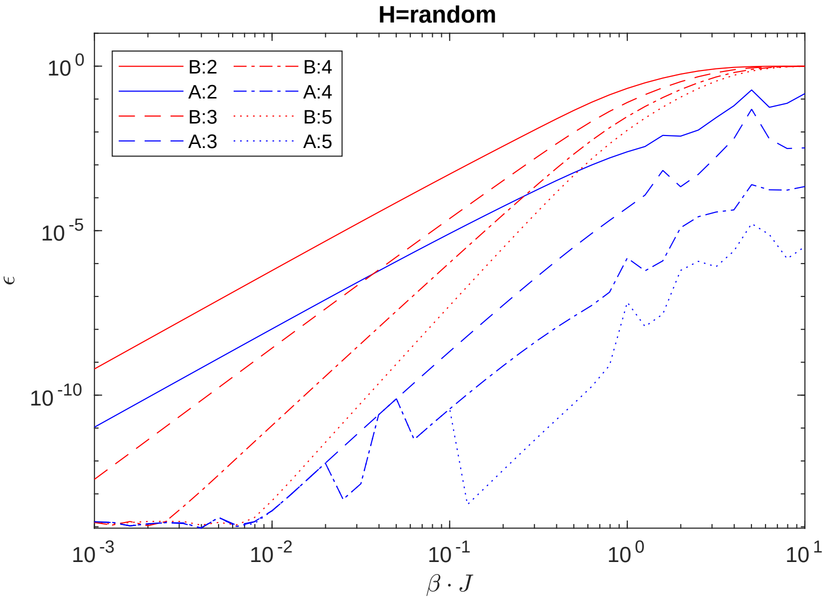


(b) test

Figure 9: test



(c) test



(d) test

Figure 9: test (cont.)

Also here type D improve the results of type B. For high β truncation seems necessary.

nog niet klaar

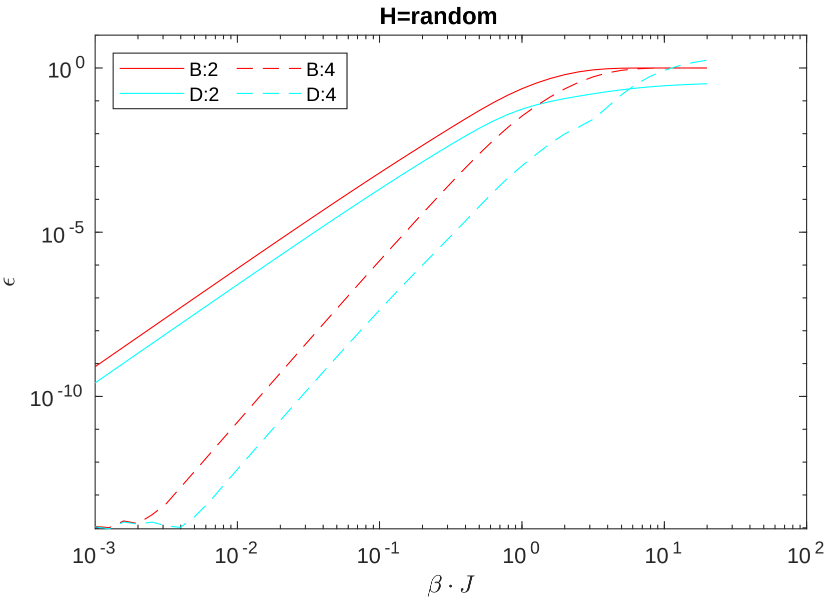


Figure 10: Comparison type C and B for random Hamiltonian

9.2 analytical results

References

[1] S. Paeckel, T. Köhler, A. Swoboda, S. R. Manmana, U. Schollwöck, C. Hubig, Time-evolution methods for matrix-product states, Annals of Physics 411 (2019). `arXiv:1901.05824`, `doi:10.1016/j.aop.2019.167998`.

[2] B. Vanhecke, L. Vanderstraeten, F. Verstraete, Symmetric cluster expansions with tensor networks, Physical Review A 103 (2) (2021). `arXiv:1912.10512`, `doi:10.1103/PhysRevA.103.L020402`.

[3] H. Nishimori, G. Ortiz, Elements of Phase Transitions and Critical Phenomena, Vol. 9780199577, Oxford University Press, 2011. `doi:10.1093/acprof:oso/9780199577224.001.0001`.

[4] G. Jaeger, The ehrenfest classification of phase transitions: Introduction and evolution, Archive for History of Exact Sciences 53 (1) (1998) 51–81. `doi:10.1007/s004070050021`.

[5] A. J. Beekman, L. Rademaker, J. van Wezel, An Introduction to Spontaneous Symmetry Breaking (sep 2019). `arXiv:1909.01820`, `doi:10.21468/scipostphyslectnotes.11`.