Todo list

Foreword

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.20

Permission of use on loan

The author(s) gives (give) permission to make this master dissertation available for consultation and to copy parts of this master dissertation for personal use. In all cases of other use, the copyright terms have to be respected, in particular with regard to the obligation to state explicitly the source when quoting results from this master dissertation.

add date

Abstract

Het abstract is maximum één bladzijde en bevat minstens: a) De informatie die werd vermeld op het titelblad (eigen vorm); b) Een summiere beschrijving van het werk (vijftien à twintig regels); c) Eventueel: drie tot vijf goed gekozen trefwoorden die het onderwerp best omschrijven.

Extended Abstract

De extended abstract heeft een standaardlengte van minimaal twee bladzijden, met een maximum van zes bladzijden.

Contents

List of Figures

Chapter 1

Introduction

> There is nothing new to be discovered in physics now. All that remains is more and more precise measurement.
>
> Lord Kelvin, 1900

1.1 Introduction

In 2015, there were about 5.6 million known physics papers in literature. At the current rate, this number doubles every 18.7 years [1]. Despite this enormous body of literature, there are a lot of things which are not completely understood. Some examples include a self-consistent theory of quantum gravity, the need for dark energy and matter in cosmology, the arrow of time, the matter-antimatter asymmetry. There even is no interpretation of quantum mechanics where everyone agrees upon.

But certainly not all open problems have to do with 'new' physics. In many areas of physics, computing the implications of relatively simple laws becomes exceedingly difficult for many particles. Of historical importance is the three-body problem, describing the trajectory of 3 gravitational bodies such as the earth, moon and sun. The general case is not solved, despite developments over the last 300 years.

In reality, the real challenge is to model the macroscopic properties of quantum many-body system with around 10^{23} particles. Needless to say, this not an easy task at all. Finding good and computable approximations is of primary importance in the fields of quantum chemistry, condensed matter physics, and materials science.

In computational chemistry, the many-body problem is tackled with methods which fall in one of the following categories: (post-) Hartree-Fock methods, density functional theory (DFT) and force-field methods. While they have many

vind bron en voorbeelden

applications , these methods are not fully able to capture all the properties of the so called strongly correlated matter.

Examples of phase of strongly correlated matter which are not yet understood include high-T superconductors, topological ordered phases, quantum spin liquids [2]. There exist different methods to investigate these exciting materials. A very limited number of models is quantum integrable, meaning they can be solved in a non pertubative way. Also, some properties of models near criticality can be determined exactly with conformal field theory (CFT). But for some systems, we can only simulate the behaviour with numerical techniques. To make progress, new fast and accurate numerical methods are needed, because exact diagonalisation becomes unfeasible for large systems.

Some examples of such numerical techniques, which will not be discussed, are: Dynamical Mean Field (DMFT) / Dynamical Cluster Approximation (DCA), Series expansion, Density Matrix Embedding Theory (DMET), Fixed-node Monte Carlo, Diagrammatic Monte Carlo, Variational Monte Carlo, Functional renormalization group (FRG) and Coupled-cluster methods. [3]

In this thesis, a technique is proposed that builds on the broad field of tensor networks.

1.2 Tensor networks

This is often referred to as the curse of dimensionality. The size of the Hilbert space of quantum states grows exponentially fast. This prevents an effecient description of all possible quantum states.

area law+picture

sign problem monte carlo, ...

write about tensor networks

Chapter 2

Tensor networks

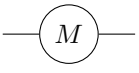
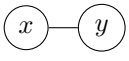
The aim of this chapter is to give a basic introduction to tensor networks from a computational viewpoint. First the graphical notation for tensors, which is ubiquitous in the field, will be introduced and explained. A incomplete classification of the different kinds of tensor networks will be discussed. Some routine tensor network manipulations are explained. Also, a selection of different algorithms, which are from a computational point of view the reason behind the success of tensor networks, are presented very briefly.

2.1 Tensor networks

2.1.1 Graphical notation

Before explaing tensor networks, some graphical notation should be introduced. This really is a way to conveniently write vectors, matrices an in general tensors without the need to introduce many labels. A tensor T is represented by a circle with a number of external legs, according to the number of external indices. Connected legs are summed. Some examples are shown in table 2.1. Every leg which is connected to multiple tensors, is contracted.

Table 2.1: Caption

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

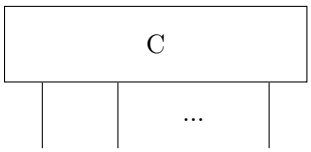


Figure 2.1: Caption

2.1.2 Representing a quantum state

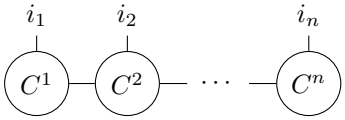
Tensor network come in many shapes and forms. Tensor networks are really used to represent a tensor with many legs. A general quantum state with N sites can be described in a given basis $|i\rangle$ in the following way:

$$|\Psi\rangle = \sum_{i_1 i_2 \cdots i_n} C^{i_1 i_2 \cdots i_n} |i_1\rangle \otimes |i_2\rangle \otimes \cdots \otimes |i_n\rangle \tag{2.1}$$

Here the tensor C holds all the information of the quantum state. The graphical representation can be seen in fig. 2.1.

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 wave function:

$$C^{i_1 i_2 \cdots i_n} = C^{1\,i_1}_{\alpha_1} C^{2\,i_2}_{\alpha_1 \alpha_2} \cdots C^{n\,i_n}_{\alpha_{n-1}} \tag{2.2}$$

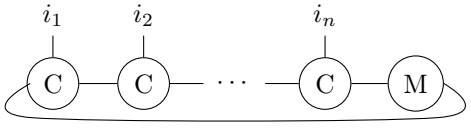


(2.3)

Where summation over shared indices is implied. It is always possible to find such a representation by means of matrix decomposition (see section 2.2). The summation over α_i are called a virtual bond and their dimension is denoted by χ. At this point, this is not yet an improvement because the bond dimension needs to be exponentially large in order to represent the tensor C exactly.

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

$$|\Psi\rangle = \mathrm{Tr}\big(C^{i_1} C^{i_2} \cdots C^{i_n} M\big) |i_1\rangle \otimes |i_2\rangle \otimes \cdots \otimes |i_n\rangle \tag{2.4}$$



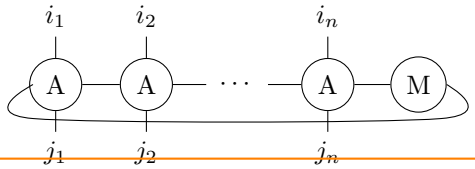
(2.5)

2.1.3 Classification

2.1.3.1 MPS

2.1.3.1.1 MPO Similarly, a Matrix Product Operator (MPO) is of the following form:

$$\hat{O} = \sum Tr(A^{i_1 j_1} A^{i_2 j_2} \cdots A^{i_n j_n} M) \times |i_1\rangle \langle j_1| \otimes |i_2\rangle \langle j_2| \otimes \cdots \otimes |i_n\rangle \langle j_n| \tag{2.6}$$



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

2.1.3.1.2 PEPS Exact contraction is hashtag P-Hard:

No exact canonical form

2.1.3.1.3 PEPO

2.1.3.1.4 Others MERA,TTN,

2.2 Tensor network manipulations

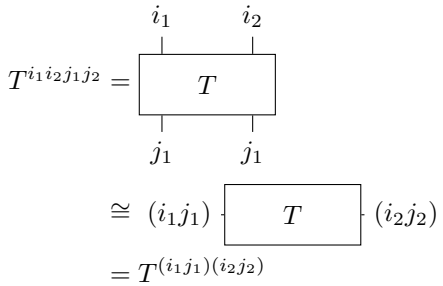
This section serves as an introduction of tensor network manipulations. The overview mainly focusses on MPS/MPO networks, but most of the oprations translate to the 2D case.

The MPS's are processed by transforming the tensor into a matrix, performing some matrix calculations and casting it back into its original form. In this way, the standard methods from linear algebra can be used. This section gives some examples how this is done in practice:

2.2.1 Basics

2.2.1.1 Grouping legs

One of the most basic manipulations is to group some legs of a tensor into one leg:

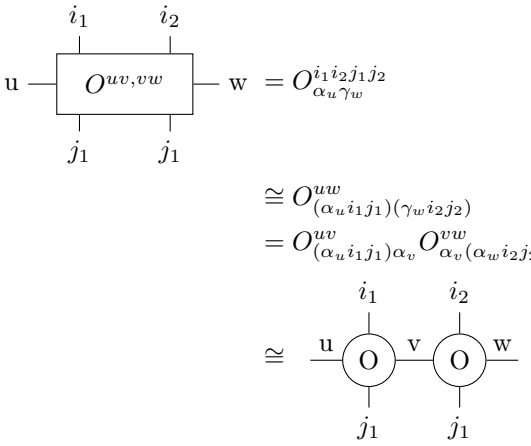


(2.7)

The dimension of the new leg is the product of the dimension of the individual legs. Contracting 2 merged legs with 2 merged legs is exactly the same as contracting them separately. The both The 4 leg tensor and matrix contain exactly the same information. Manipulating this in memory requires both permute and reshape commands. This requires some time, the internal representation of the matrix changes.

2.2.1.2 decomposition

The grouping above can be applied to decompose a tensor into 2 tensor with matrix techniques. An example, which will be needed later on, is give here.



(2.8)

The indices U,V and W represent blocks indices. Step 2 reshapes and groups the indices on to one index on the left and one on the right. The dimension of this index is the product of the separate 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:

$$\dim v = \min(\dim u, \dim v) + 2 \dim i \tag{2.9}$$

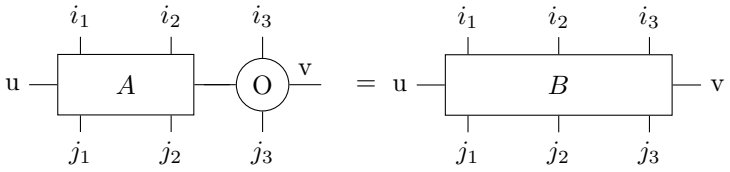
Many matrix decompositions exist. Some useful examples here are SVD decomposition, eigenvalue decomposition, QR, $\cdots$.

2.2.1.3 virtual levels

In the previous example, the levels were indicate with a block index or virtual level. The idea is to create seperate the contraction into blocks. This is completely analogous to matrix block multipliction. This wil be a more natural form to represent the algorithm. Of course, one can easily switch between block representation and the full one.

2.2.1.4 inverse

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



(2.10)

Again, the indices can be taken toghether in the following way: $\alpha = (ui_1j_1i_2j_2)$ and $\beta = (i_3j_3v)$:

$$A_{\alpha\gamma} O_{\gamma\beta} = B_{\alpha\beta} \tag{2.11}$$

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. Linear solver are generally much faster. As this is one of the core problems to solve both in 1D and 2D, this will be discussed in detail in section 4.4.

2.2.1.5 contraction order

contraction order

2.2.1.6 Gauge freedom

gauge

2.2.1.7 truncation

svd truncation

2.2.2 MPS algoritms

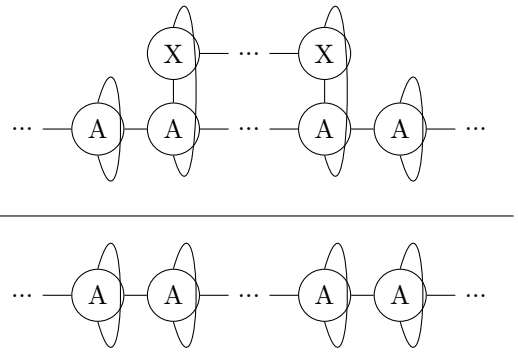
2.2.2.1 cononical form

schmidt decomp,

2.2.2.2 DMRG

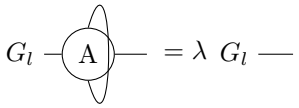
2.2.2.3 Expectation values

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:

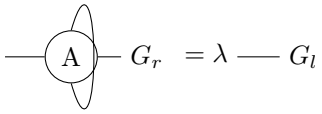
$\langle X \rangle =$ 

(2.12)

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.

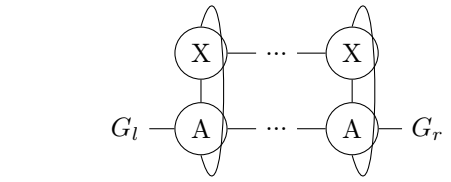


(2.13)



(2.14)

Equation eq. (2.12) can now be easily caclulated:



$$\langle X \rangle = \frac{}{\lambda^n \quad G_r \; G_r} \tag{2.15}$$

2.3 Tensor network algorithms

While there are many

2.3.1 MPS algorithms

2.3.2 Canonical form

2.3.2.1 DMRG

2.3.3 PEPS algorithms

2.3.3.1 PEPS contraction

2.3.3.1.1 CTMRG

2.3.3.1.2 Vumps

Chapter 3

Strongly correlated matter

3.1 Phases and Criticality

3.1.1 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. [4]. 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 [5], 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.

3.1.2 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: $[H, U] = 0$ [6]. 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.[6]

3.1.3 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.

3.1.4 Critical exponents for spin systems

The following table defines some of the critical exponents for the Ising system.

Symbol	name
m	magnetisation
ξ	correlation length
g	external field
t	reduced temperature
d	dimension

The 2 point correlation function is defined as $f(x,y) = \langle m(x)m(y)\rangle - \langle m(x)\rangle \langle m(y)\rangle$. At larger distances this decays exponentially fast (see **??**) $f(x,y) = e^{-\frac{|x-y|}{\xi}}$, where ξ is the correlation length.

for the ordered phase, the following relations hold: $m\ |t|^{\beta}$, $\xi(t) \approx |t|^{-\nu}$. At the critical temperature near a quantum phase transition $m \approx |g - g_c|^{\frac{1}{\delta}}$.

3.1.5 Finite size scaling

Finite size scaling was originally introduced in order to extract critical exponents from monte carlo simulations.

3.1.6 CFT

central charge

3.1.7 Quantum phase transitions

A traditional 2nd order phase transition is driven by a change in temperature. Quantum phase transitions on the other hand happen at zero temperature under influence of another parameter g of the model. At finite temperature, 2 things can happen: either there is a line connecting a classical 2nd order phase transition to the quantum phase transition, or the phase transition disappears at finite temperature [7].

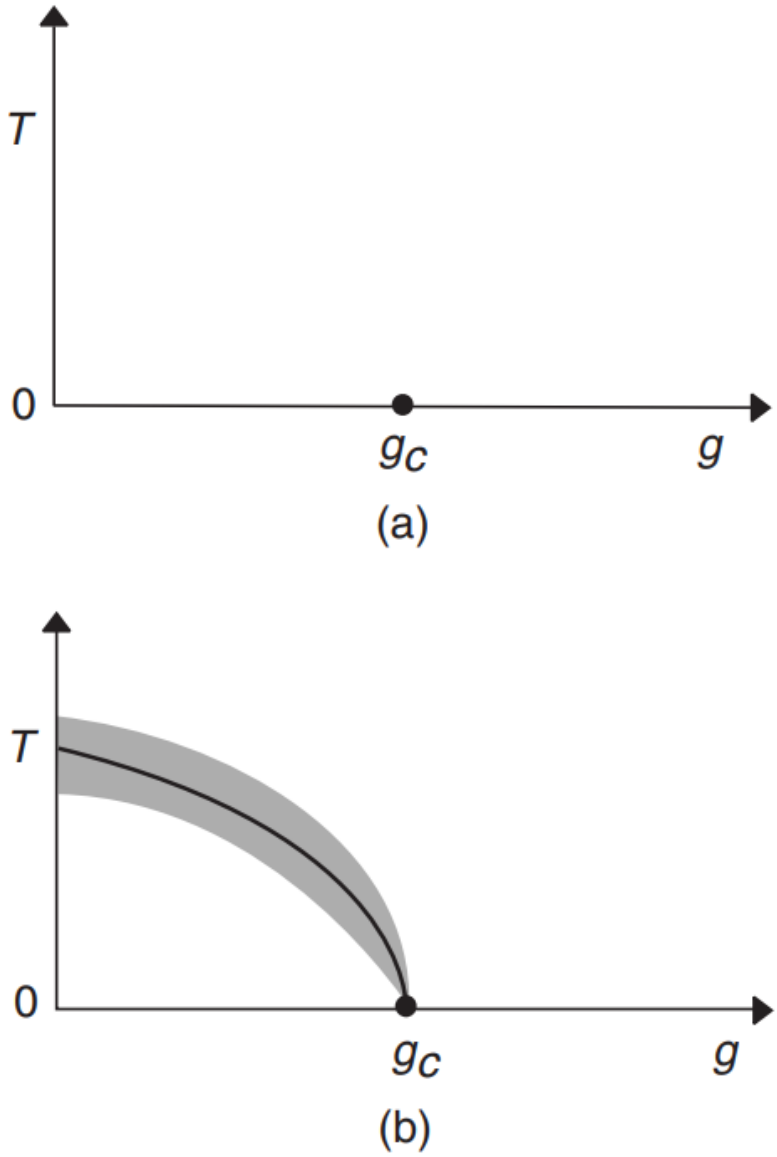


Figure 3.1: Two possible phase diagrams of a system near a quantum phase transition. In both cases there is a quantum critical point at $g = g_c$ and $T = 0$. In (b), there is a line of $T > 0$ second-order phase transitions terminating at the quantum critical point. The theory of phase transitions in classical systems driven by thermal fluctuations can be applied within the shaded region of (b). Figure and caption taken from [7].

3.1.8 Quantum to classical mapping

Quantum to classical mapping

3.2 Models

The goal of numerical techniques is to simulate the physics of real world systems. These are, to some extend, captured by different models. Models are a simplified mathematical description that captures some relevant physics of more complicated systems. This section introduces some specific models, their relevance and some properties. These models will be used later te benchmark the developed tensor network expansion.

3.2.1 Ising model

The prototypical example of a model in the field of strongly correlated matter is the Ising model. It was first introduced 1925 by Ernest Ising, as a model to capture ferromagnetism. He proved that for a linear chain, there is no phase transition at finite temperature. He wrongly concluded that this would also be the case in higher dimensions, but it turned out t be one of the deepest and far-reaching problems in 20th century [8].

The Ising model, in essence, assigns an energy contribution to neighbouring spins. These spins sit on a fixed position on a chain (1D) or lattice (2D/3D/...). In classical Ising, the operators in the Hamiltonian all commute with each other. An energy is assign between neighbouring spins and possible and energy for alignment with an external magnitic field in the same direction. In quantum Ising model, a transversal field is added. Often, the particles on the grid are spin 1/2 particles, but of course other particles are possible.

Many generalizations exist for the Ising model.

q potts,...

3.2.1.1 Classical Ising

The classical ising model is given by the following hamiltonian:

$$H = -J\left(\sum_{<ij>}\sigma_i\sigma_j + h\sum_i \sigma_i\right) \tag{3.1}$$

where $< ij >$ runs over all neighbouring lattice sites. The possible values of σ depends on the spin dimension. For spin 1/2 lattices $\sigma \in -1, +1$. g encodes the interaction strength of the external magnetic field.

The sign J determines the low temperature ground state. A positive J will tend to allign all neighbouring spins at low temperature. This is often called ferromagnetic, because all the aligned spins cause a macroscopic magnitisation. On the other hand, a negative J causes neighbouring spins to have an opposite sign.

Depending on the sign of the longitudinal field h, the spins tend to align or anti align with this external field. This lifts the degeneracy of the groundstate.

blabla

3.2.1.1.1 1D The classical 1D model was solved analytically by Lens.

3.2.1.1.2 2D In 2D, it becomes important to define the lattice. Here, and in the simulations, we will consider a square lattice. This model was famously solved by Lars Onsager in 1944, by using the transfer matrix method. In 2 dimensions, the Ising model has a phase transition at finite temperature. The critical temperature is $T_c = \frac{2J}{Tln(\sqrt{2}+1)}$.

Only the $h = 0$ case the is solved analytically. For higher dimensions, no analytical solution is known. For these cases, we need to use numerical techniques if we want to understand the behaviour of these models.

On different lattices, interesting thins can happen. For instance, the groundstate of an antiferromagnet on a triangular lattice is not obvious to determine. The spins tend to antiallign, but at least 2 of 3 spins on the corner of a triangle have to align. Remarkably, as will be explained in section 3.1, the physics at the phase transition does remain invariant when the lattice is changed.

3.2.1.2 Quantum Ising

As we all know, the real world behaves, certainly at small length and time scales, quantum mechanically. Therefore, it is important to understand how the quantum Ising model differs from the classical model. In the quantum Ising model, the operators no longer commute with each other. An example is the transversal Ising model given by the following hamiltonian:

$$\hat{H} = -J\left(\sum_{<ij>}\sigma_i^x\sigma_j^x + g\sum_i \sigma_i^z\right) \tag{3.2}$$

In the case that $g = 0$, this is the classical Ising model (in the $h = 0$ case).

3.2.1.2.1 1D Different to the classical case, the 1D model already contains a phase transition.

3.2.1.2.2 2D

3.2.2 Heisenberg

The heisenberg model is given by:

$$\hat{H} = -\left(\sum_{<ij>} J_x\sigma_i^z\sigma_j^z + J_y\sigma_i^y\sigma_j^y + J_z\sigma_i^z\sigma_j^z + h\sum_i \sigma_i^z\right) \tag{3.3}$$

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.

3.2.3 Random

It's also possible to construct random hamiltonians. in basis: hermitian H

3.3 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

3.3.1 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:

$$\begin{aligned} Z &= \sum e^{-\beta E_n} \\ &= \sum_n \left\langle n \left| e^{-\beta \hat{H}} \right| n \right\rangle \\ &= \mathrm{Tr}\left(e^{-\beta \hat{H}}\right) \end{aligned} \tag{3.4}$$

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:

$$p_i = \frac{\sum e^{-\beta E_i}}{Z} \tag{3.5}$$

An useful quantity is the density matrix ρ.

$$\begin{aligned} \rho &= \sum_j p_i \left|\Psi_j\right\rangle \left\langle\Psi_j\right| \\ &= \sum_j \frac{e^{-\beta \hat{H}}}{Z} \left|\Psi_j\right\rangle \left\langle\Psi_j\right| \end{aligned} \tag{3.6}$$

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

$$\begin{aligned} Z &= \mathrm{Tr}(\rho) \\ \langle X \rangle &= \mathrm{Tr}\Big(\rho \hat{X}\Big) \end{aligned} \tag{3.7}$$

3.3.2 Time evolutions

LSZ theorema herbekijken

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

3.3.2.1 ground state

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

3.3.3 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 [9].

[9]

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

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

3.3.3.2 global Krylov method

3.3.3.3 MPS-local methods

local Krylov TDVP

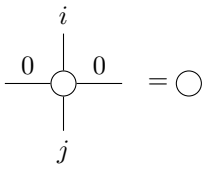
Chapter 4

Construction Cluster expansion

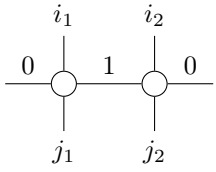
4.1 Introduction

4.1.1 Notation

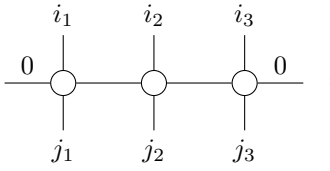
In the following, the external legs and virtual level 0 will be omitted:



$$= \bigcirc \tag{4.1}$$



$$= \bigcirc\!\!-\!\!\bigcirc \tag{4.2}$$

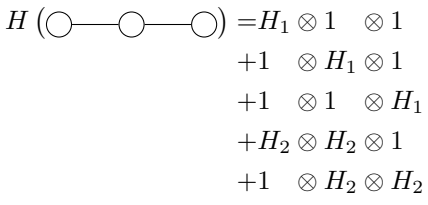


$$= \bigcirc\!\!-\!\!\bigcirc\!\!-\!\!\bigcirc \tag{4.3}$$

This hamiltonian consists of 1 and 2 site operators. Of course more general hamiltonians can also be captured.

$$\hat{H} = \left(\sum_{<ij>} H_2^i H_2^j + \sum_i H_1^i \right) \tag{4.4}$$

The same notation will be used to denote the hamiltonin evaluated on the given geometry:



$$\begin{aligned} H(\bigcirc\!\!-\!\!\bigcirc\!\!-\!\!\bigcirc) = & H_1 \otimes 1 \otimes 1 \\ & + 1 \otimes H_1 \otimes 1 \\ & + 1 \otimes 1 \otimes H_1 \\ & + H_2 \otimes H_2 \otimes 1 \\ & + 1 \otimes H_2 \otimes H_2 \end{aligned} \tag{4.5}$$

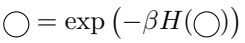
4.1.2 Idea

This chapter shows the main construction of dissertation. A cluster expansion is used to approximate $e^{\hat{H}}$ for every possible geometry. The goal is to make a MPO/PEPO which captures the tensor exponential in the thermodynamic limit.

symmetry, speed

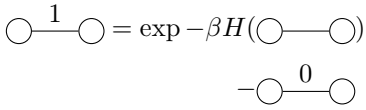
This cluster expansions introduced in [10]. The main idea is to make an extensive expansion by adding blocks which solve the model exactly on a local patch. Crucially, the expansion is not in the inverse temperature β but in the size of the patches. The local patches are separated by a virtual level 0 bond.

To make this somewhat more precise, the first steps of the expansion are shown here. The smallest patch, i.e. 1 site, encodes the exponential of that hamiltonian.



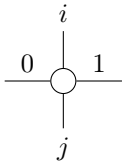
$$\bigcirc = \exp\left(-\beta H(\bigcirc)\right) \tag{4.6}$$

If there were no 2 site interaction, this already captures the full diagonilsation. Of course, such a model wouldn't be useful. The next step is to introduce 2 site interactions, where the one site interactions previously introduced interaction are subtracted from the diagonalised hamiltonian.

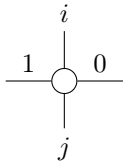


$$\bigcirc\overset{1}{-}\bigcirc = \exp -\beta H(\bigcirc\!\!-\!\!\bigcirc) - \bigcirc\overset{0}{-}\bigcirc \tag{4.7}$$

At this stage, all seperated networks with maximally 2 connected sites in a row are diagonalised exactly. Notice that here, 2 new blocks are introduced:

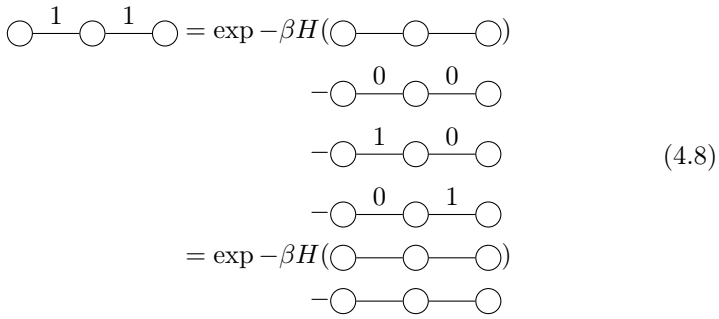


and of course also



. As can be seen, the dimension of sublevel 1 needs to be d^2, with d the dimension of physical level. Although

different possible constructions already differ in the next step, one more step is added to make te construction and notation clear.



(4.8)

It is clear that the right-hand side of eq. (4.8) can alsoe be ommited, as it is just evaluating the exponentiated hamiltonian on the same geometry as the left hand side and substructing all possible contractions of the blocks which were added previously.

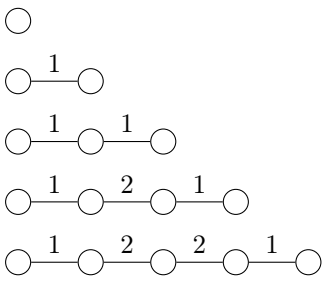
4.1.3 Preview

In the following sections some possible configurations in 1D and 2D will be discussed. At this point, the focus is on the construction and its bond dimension. The results will be discussed later in chapter 5. T

4.2 Construction MPO

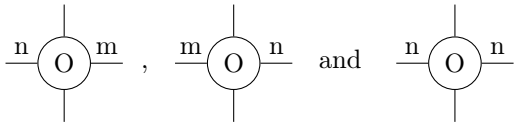
4.2.1 Type A

This type was originally proposed in [10]. The first few blocks in the expansion are the following:



(4.9)

The following types of blocks appear in the cluster expansion



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

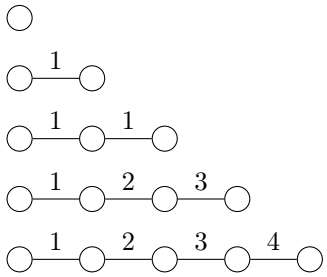
The O^{nn} block is in defined for a chain with an odd number of sites. The contraction of O^{nm} and O^{mn} is defined by for a chain with even order. The decomposition is defined up to a gauge transformation.

4.2.1.0.1 Dimension In this scheme, virtual level n has dimension d^n. Of course, this dimension can be lowered if some error is allowed for the longest chain.

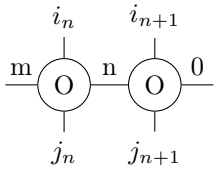
4.2.1.0.2 discussion Type A can form long chains, which where not explicitly optimised for. The question arise whether this will results in accurate results for cyclic systems or not.

4.2.2 Type B

Type B only contains blocks of the following form; O^{mn} and O^{n0}.The first few blocks are:



(4.10)

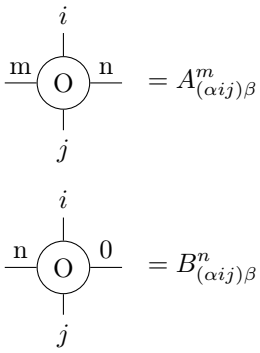


$= U^n \Sigma V^\dagger$ (4.11)

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

4.2.2.0.1 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:



(4.12)

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

$$\begin{aligned} S &= A^m A^n = A'^m A'^n \\ T &= A^m B^n = A'^m B'^n \end{aligned} \tag{4.13}$$

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

$$\begin{aligned} S^\dagger &= (U\Sigma_1)Q^\dagger \\ T^\dagger &= (V\Sigma_2)Q^\dagger \end{aligned} \tag{4.14}$$

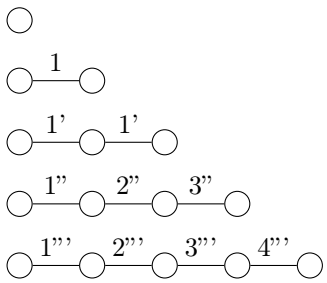
The new bond dimension is the $\dim n' = d^2 \cdot \min(\dim n - 1, \dim(n+1))$. This is higher than the dimension of type A.

4.2.2.0.2 Discussion The bond dimension is larger than type A, but the long chains from A are absent. The left inverse is always well defined and doesn't need any computation, because hermitian matrix U can be inverted easily. One major drawback is that for long chains, the virtula bonds are very large before they can be shrunk with the gsvd procedure.

4.2.3 Type C

primed virtual levels

This type implements the same strict type as Type B, but 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:



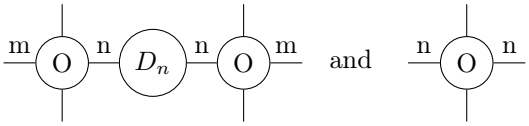
(4.15)

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.

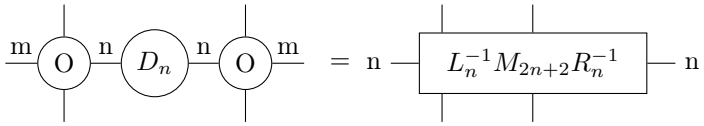
4.2.3.0.1 discussion As can be expected from the construction, the bond dimension grows very fast. This type is just as precise as Type B.

4.2.4 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:



Similar to type A,



$$= U\Sigma V^\dagger \tag{4.16}$$

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

4.2.4.0.1 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

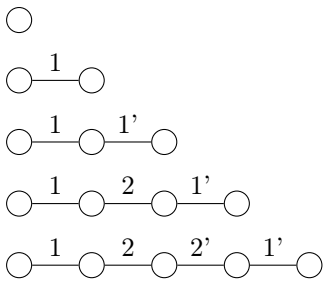
4.2.4.0.2 matrisation The cost of this type lies in the fact that it has no compact way of casting it to a matrix. The following works, but has quite a large dimension:

O_{00} O_{01} O_{10} O_{12} O_{21}	$-2O_{01}$ $-2O_{12}$	O_{01} O_{12}	$O_{01}D_1^{1/2}$
O_{10} O_{21}			
O_{10} O_{21}		O_{11} O_{22}	
$D_1^{1/2}O_{10}$			$D_1^{-1/2}O_{12}D_2^{1/2}$ $D_2^{1/2}O_{21}D_1^{-1/2}$

fix this

4.2.5 Type E

Again, this is a strict variant which needs exactly twice the bond dimension of type A. The idea is to split every chain in a left and a right part. For the left part, the numbers increase while right part they decrease. This construction carries over well to higher dimensions. The first few blocks are:



(4.17)

The construction is very similar to type A

4.3 Construction PEPO

4.3.1 Linear block

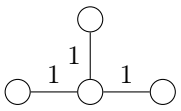
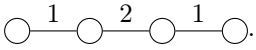
In the following, linear blocks means that the graph is a rooted tree.

4.3.2 Blocks with loops

4.4 Framework implementation

The construction can be written down quite compactly as done in the previous section, but actually implementing the code in full generality is somewhat more complex. In practice, the 1D and 2D implementation were performed separately. where a part of the code but mainly the ideas and lessons learned from 1D where

taken to the 2D code. Of course, the 1D construction is just a subset of the 2D implementation. In fact, the 2D implementation even outperforms the 1D implementation for reasons with will be explained later. The main focus of this chapter will go to the 2D implementation. At the end some particular optimisations in 1D will be highlighted.

Solving the blocks introduced in the previous sections need 2 different approaches. The graphs of the maps can be split in 2 groups. The first one, considers problems where there are no loops and at most one node with more than 2 legs Examples include:  and  . These will be reduced to a standard matrix problem, and solve with matrix (pseudo-) inversion.

The other group, of course, constitutes the nonlinear problems. This includes every problem where a block (or rotated version) occurs more than once, problems which include loops, ...

4.4.1 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 $I^m = (i_1^1 i_2^1 \cdots i_{n_1}^1)$, then the problem can in general, after some tedious tensor leg bookkeeping, be rewritten in the following form:

$$\begin{aligned} &A_{I_1 I_2 \cdots I_n \alpha^1 \alpha^2 \cdots \alpha^m} X_{\alpha^1 \alpha^2 \cdots \alpha^m j} \\ =&B_{I_1 I_2 \cdots I_n j} \end{aligned} \tag{4.18}$$

Here i_N^M 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:

$$\begin{aligned} A^1_{I^1 \alpha^1} X_{\alpha^1 I^2 \cdots I^m j} &= B_{I_1 I_2 \cdots I_n j} \\ A^2_{I^2 \alpha^2} X_{\alpha^1 \alpha^2 I^3 \cdots I^m j} &= B_{\alpha^1 I_2 \cdots I_n j} \\ &\vdots \\ A^m_{I^m \alpha^m} X_{\alpha^1 \alpha^2 \cdots \alpha^m j} &= B_{\alpha^1 \alpha^2 \cdots \alpha^{m-1} I_m j} \end{aligned} \tag{4.19}$$

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 decomposion 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

4.4.2 Nonlinear solver

In some cases, the above solver does not return the best possible solution to a given problem. The reason is that it is not able to incorporate symmetries or solve problems where the new blocks appear more than once. A new solver is needed which does not rely on methods from linear algebra, but on more general non-linear leat squares solvers.

In essence, the non-linear least squares solver needs as input a vector with the error values $\vec{f}(\vec{x})$, and if possible also the jocabian, i.e. $J_{I,J} = \frac{\partial f_I}{\partial x_J}$. An improved point x is chosen by the algoritm, untill some convergence criterium is reached. The implementation uses matlab fsolve routine, which uses Levenberg-Marquardt algoritm under the hood.

4.4.2.0.1 automatic differentiation With some care, the jacobian can be calculated for a general tensor network in an automated way. Suppose we want to differentiate the contracted tensor $T^{i_1 \cdots i_n}$ with respect to one of the PEPO blocks $x_n = O^{i_n}_{\alpha\beta\gamma\delta}$. Denote $I = (i_1 \cdots i_n)$ and $J = (i_m \alpha\beta\gamma\delta)$, and this block only occurs once. Then $J_{IJ} = \frac{\partial T^{i_1 \cdots i_n}}{\partial O^{i_m}_{\alpha\beta\gamma\delta}} = T^{i_1 \cdots i_n}_{i_m \alpha\beta\gamma\delta} \delta^{i_n}_{i_m}$ amounts to contracting the network with the tensor x_m removed, and treating the non contracted indices as external ones. If a tensor appears in multiple places, the sum of these contributions has to be taken into account.

source

4.4.2.0.2 Symmetry The non-linear solver can handle rotated and permuted blocks. For instance, a simple loop (square) can be solved by rotating one tensor $T^I_{\alpha\alpha 00}$ 4 times, once for every corner. Another example is the following decomposition: $X^I_\alpha X^J_\alpha = T^{IJ}$.

4.4.3 Optimasation

4.4.3.1 Bookkeeping

One imporatant aspect of programming these general solvers is to devise a scheme that keeps track of all the involved tensors and transforms to problem to the form discribed above. In the code, the geometric info is represented by

a map. This keeps track of the neighbourghs for each site, numbers the inernal and external legs and a list to perform the contractions.

The framework provides some tools to transform these maps into other maps, for instance by removing 1 site.

4.4.3.2 Fast contraction

One particular task is to determine all the possible combinations of virtual levels for a given geometry. Simply looping over all possible combinations scales as n^m, with the number of virtual levels and m the number of internal legs. This problem can be restated as a PEPS contraction in the following way: for each site make a tensor $T_{i(\alpha\beta\gamma\delta)} = \delta^i_{(\alpha\beta\gamma\delta)}$ where i encodes the non-zero combinations of legs $(\alpha\beta\gamma\delta)$. After setting the boundary conditions, the PEPS network can be contracted and the resulting tensor gives, after decoding, all the possible contraction. Due to its sparsity, this performs quit fast.

tikz figure

4.4.3.3 Normalisation

For many of the end results, the PEPO cells can be divided by a normalisation factor. Normalising the calculations is important, because $\exp\left(\hat{H}\right)$ scales exponentially in the number of sites. Luckily, the exponential can be calculated directly in normalised form. Suppose H is the matrisation of the hamiltonian evaluated for a certain geometry. This is a hermitian matrix and can be diagonalised $H = QDQ^\dagger$ with Q unitary. Then

$$exp(H_{d^N} - N\alpha I) = Qexp(D - N\log(\alpha)I)Q^\dagger \tag{4.20}$$

$$= Q \begin{bmatrix} exp(D_{11} - N\log(\alpha)) & & \\ & \ddots & \\ & & exp(D_{d^N d^N} - N\log(\alpha)) \end{bmatrix} Q^\dagger \tag{4.21}$$

$$= \frac{exp(H_{d^N})}{\alpha^N} \tag{4.22}$$

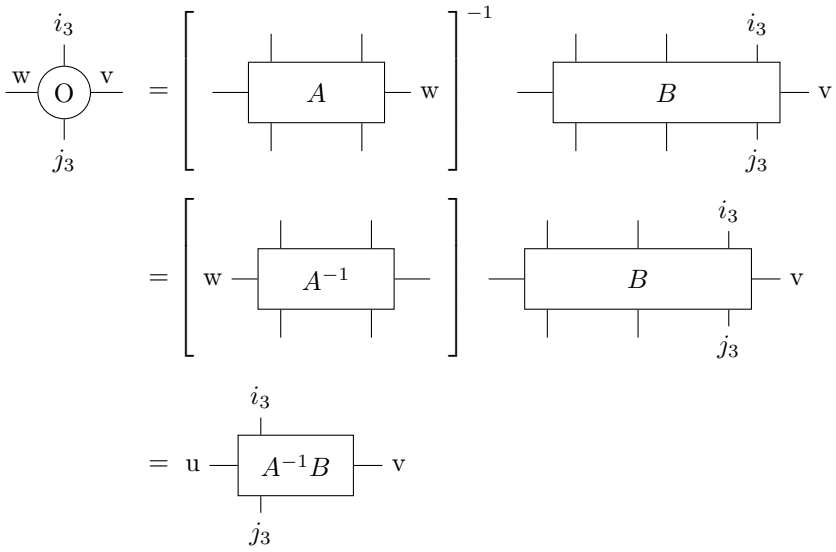
. With I the unit matrix. Next to a global normalisation factor, every block calculation calculates a specific normalisation factor such that the largest eigenvalue of $exp(H)$ is of the order 1.

4.4.3.4 Internal representation

Two main internal 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 doesn't support multidimensional matrices by default, this library is used.

4.4.4 1D implementation

The solution will be denoted by



(4.23)

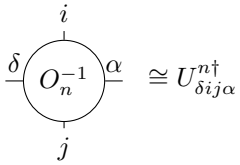
For the first equation the unmarked legs on the same positions need to be contracted with each other. The second line the mirrored positions are contracted.

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.

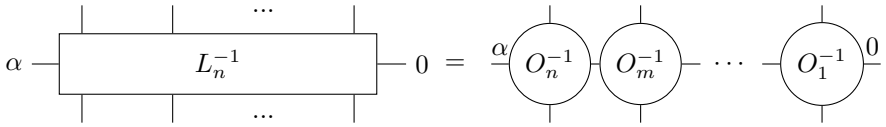
$$\begin{aligned} U^n_{(\alpha ij)\beta} A_{\beta\gamma} &= B_{\alpha ij\gamma} \\ A_{\delta\gamma} &= U^{n\dagger}_{\delta(\alpha ij)} B_{\alpha ij\gamma} \end{aligned} \tag{4.24}$$

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



$$\cong U^{n\dagger}_{\delta ij\alpha} \tag{4.25}$$

With the notation from eq. (4.23) we have:



(4.26)

update

The inverse can be applied sequentially.

4.4.4.1 buffering results

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

4.4.4.2 Fast inverses

4.4.5 How to use this code

All the code neede to generate all the results from this dissertation is available on my github page `https://github.com/DavidDevoogdt/Thesis_Tensor_Networks`. The starting points to explore the code are in the readme file.

4.4.5.1 Source code structure 1D

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.

source code 2D

4.4.5.2 Source code structure 2D

Chapter 5

Results

> With four parameters I can fit an elephant, and with five I can make him wiggle his trunk
>
> John von Neumann

5.1 Benchmarking

5.1.1 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:

$$R = d^{2n} \times 16bytes \tag{5.1}$$

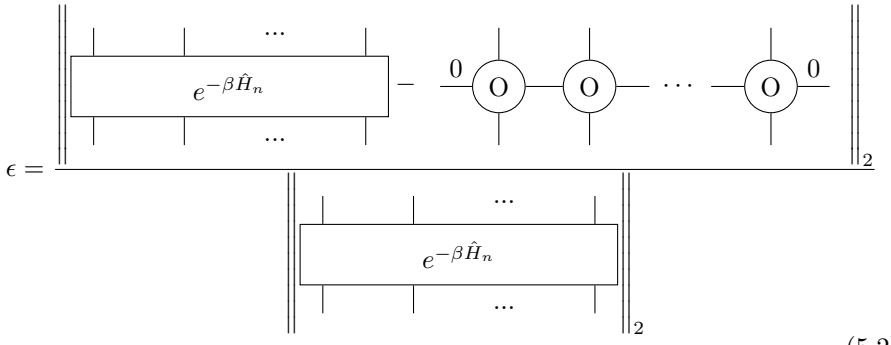
Which means a 14 site chain already takes up GB of RAM.

time complexity algoritms

5.1.1.1 norms

trace norm, schatten p norm, ...

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



(5.2)

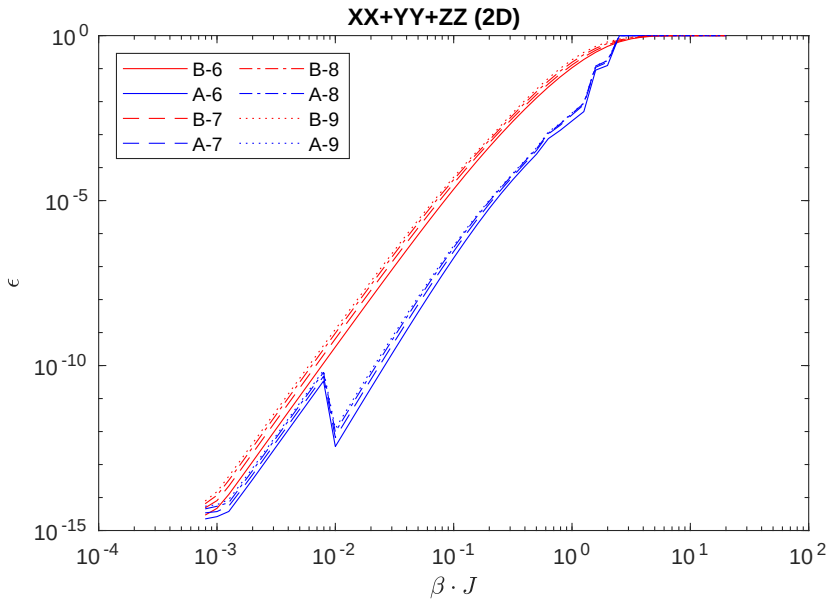
make version for cyclic

5.1.1.1.1 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. 5.1 . As expected, the cyclic norm represents large systems better for the same number of sites. The linear norm keeps increasing with every added site.

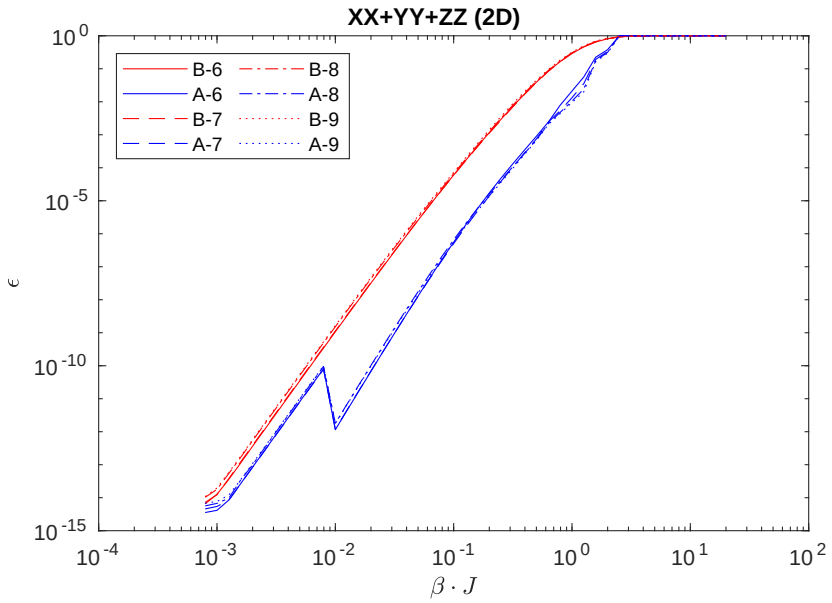
calculate complexity

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

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



(a) test



(b) test

Figure 5.1: test

5.1.2 analytical results

5.2 Benchmarking

5.2.1 Inversion procedure

5.2.1.1 Sequantiual pseudo-inversion

write this cleanly

5.2.1.1.1 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:

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.

The only parameter in the construction is σ_0 As can be seen in fig. 5.2, mainly the construction for small values of β get affected by the choice of σ_0.This can be seen in fig. 5.2. 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.

5.2.1.2 Full pseudo-inversion

5.2.2 Models

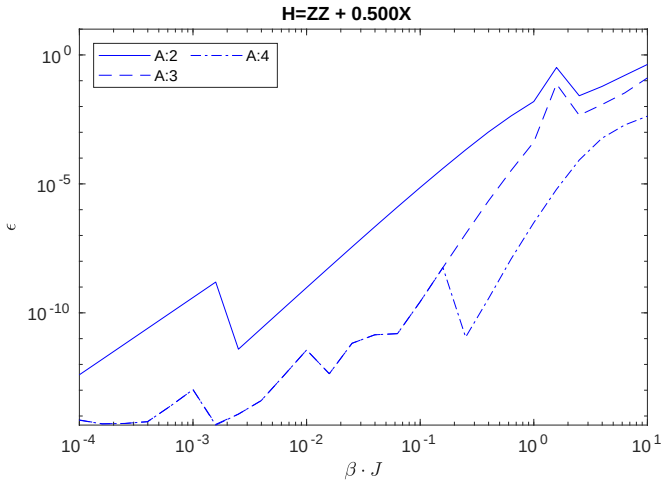
5.2.2.1 Ising

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.

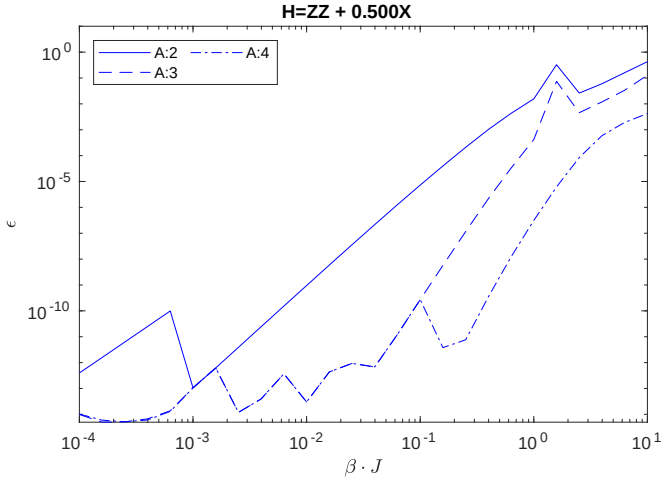
The behaviour of type B is more chaotic. The error increases no longer monotonously. For small values of β, the order is truncated.

larger orders need reimplementation (non matrix based)

For type 5 fig. 5.4, there is a consitent improvement over type B.

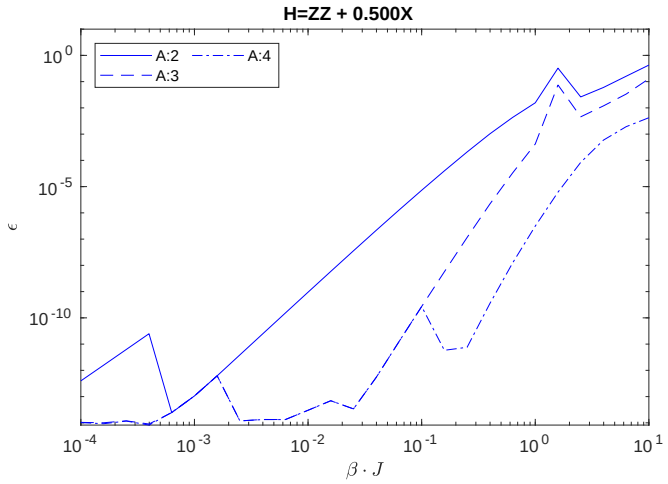


(a) 10^{-10}

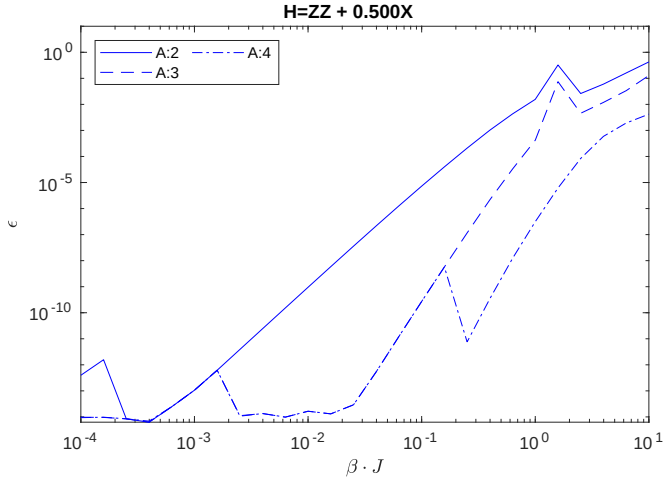


(b) 10^{-11}

Figure 5.2: A figure with two subfigures

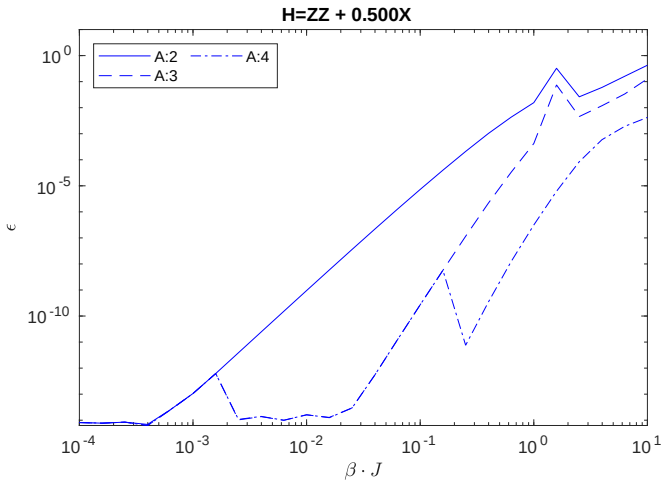


(c) 10^{-12}



(d) 10^{-13}

Figure 5.2: A figure with two subfigures



(e) 10^{-14}

Figure 5.2: A figure with two subfigures

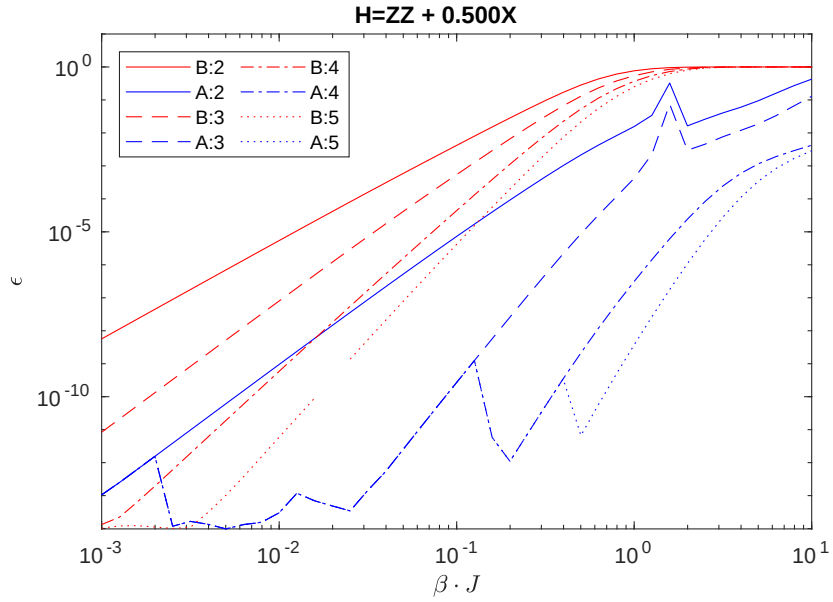


Figure 5.3: Comparison type A and B for Transversal Ising

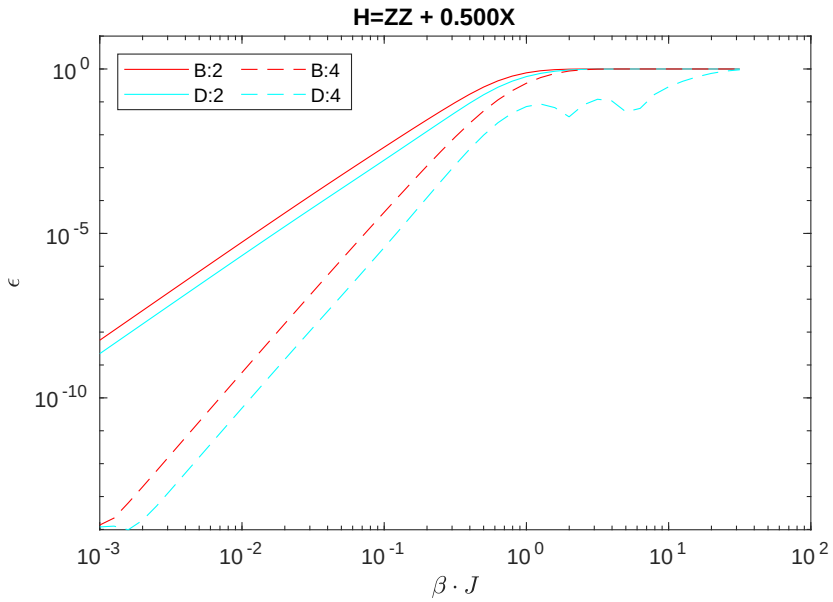


Figure 5.4: Comparison type C and B for Transversal Ising

5.2.2.2 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. 5.6 is more promising, but higher orders require to much memory to simulate.

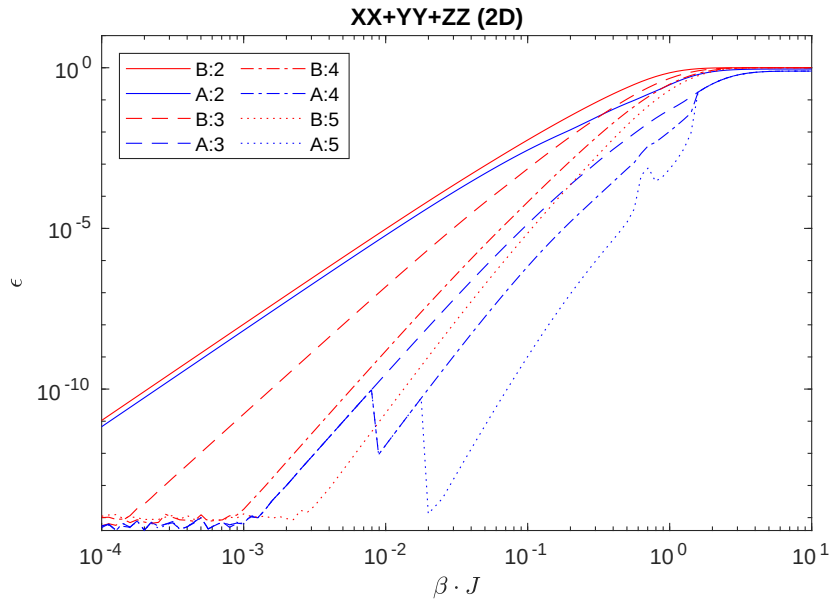


Figure 5.5: Comparison type A and B for Heisenberg

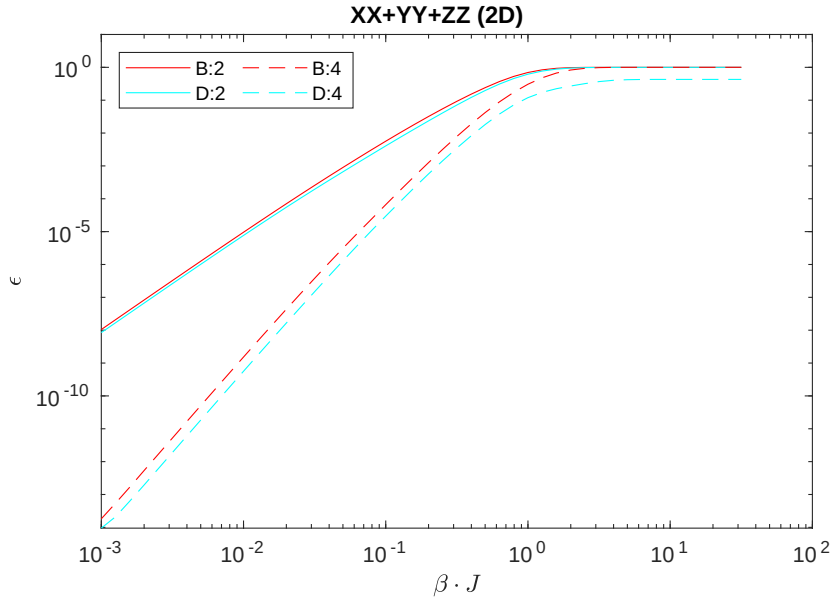


Figure 5.6: Comparison type C and B for Heisenberg

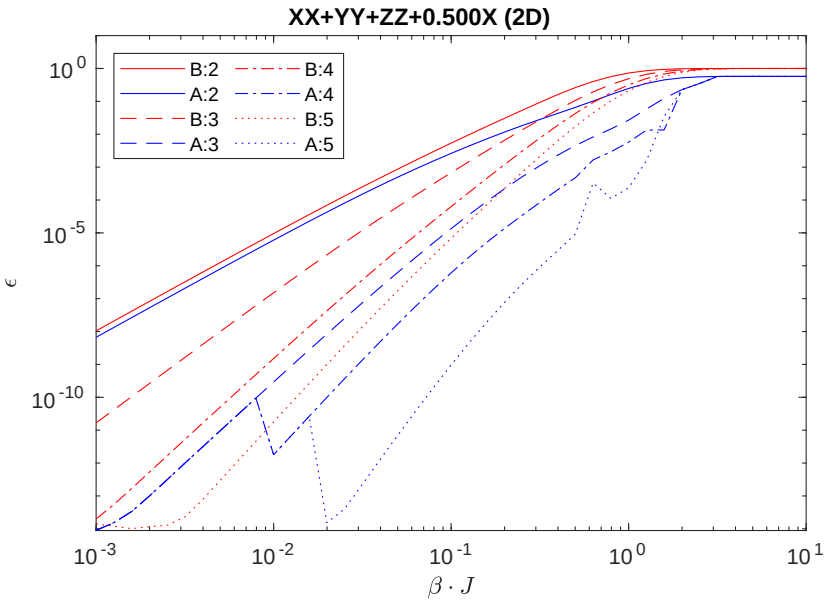


Figure 5.7: transversal XXX

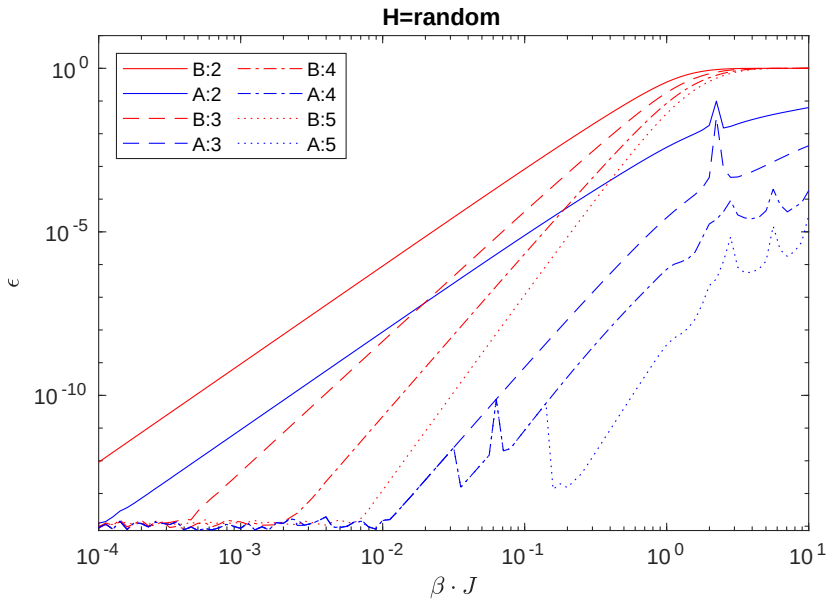
run with M=11

5.2.2.3 Random

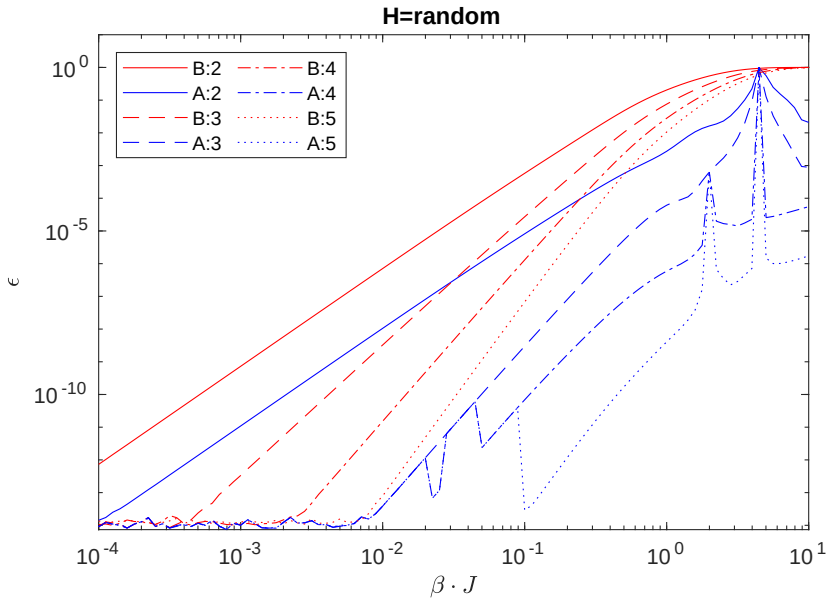
To give a representative overview for random hamiltonians, several simulations were run. The single site and nearest neighbourgh hamiltonians are generated by making hermitian matrices with random real and complex numbers between -1 and 1. In order to compare the different graphs, the engergy scale is 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. 5.8b, 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

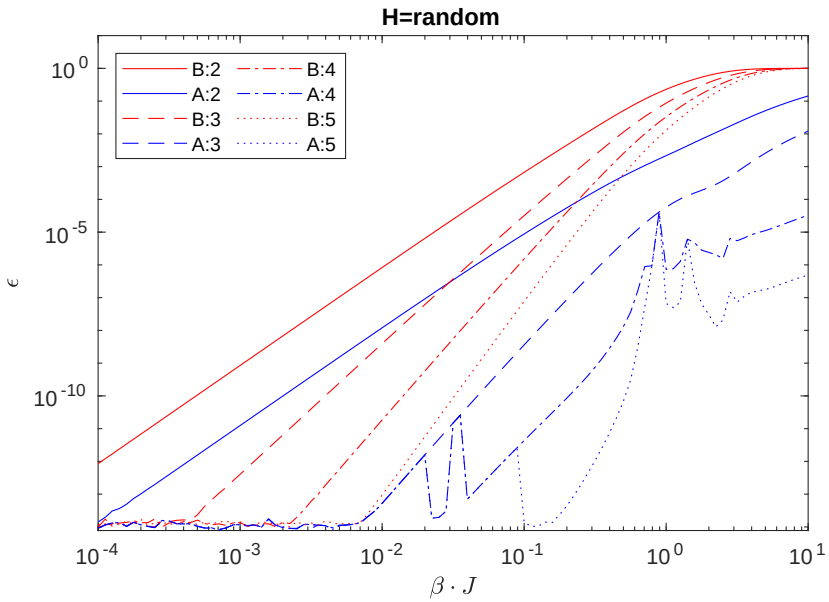


(a) test

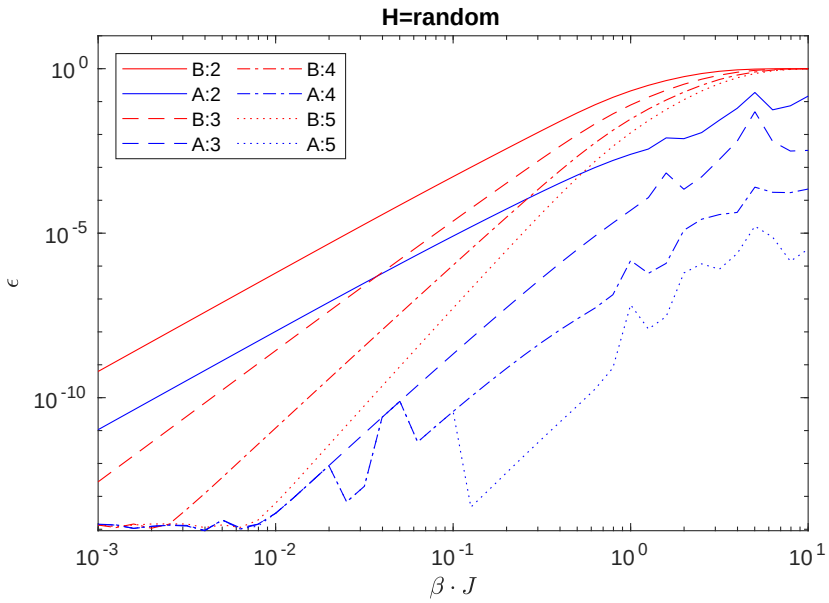


(b) test

Figure 5.8: test



(c) test



(d) test

Figure 5.8: test (cont.)

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

necessary. nog niet klaar

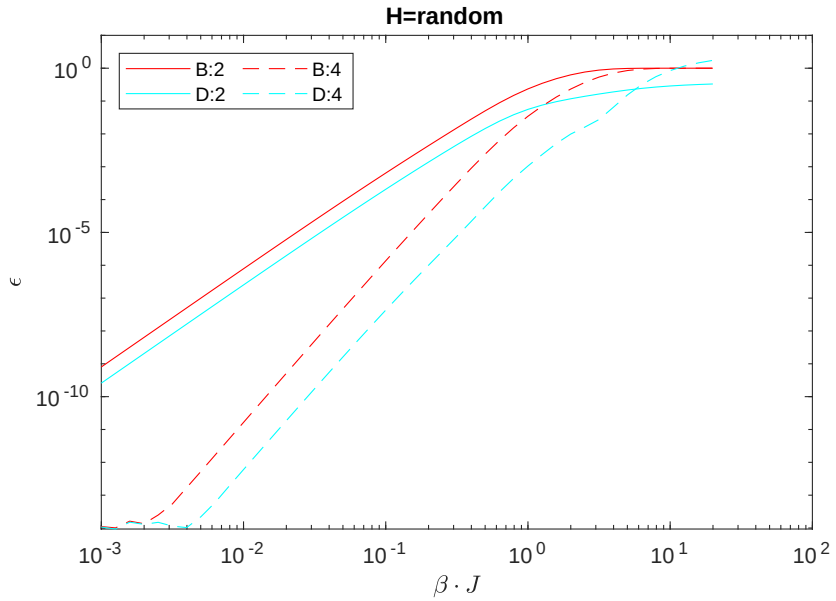


Figure 5.9: Comparison type C and B for random Hamiltonian

Chapter 6

Conclusion and lookout

Appendices

Appendix A

Source code and documentation

Bibliography

[1] R. Sinatra, P. Deville, M. Szell, D. Wang, A. L. Barabási, A century of physics (oct 2015). arXiv:1612.00079, doi:10.1038/nphys3494.

[2] R. Orús, A practical introduction to tensor networks: Matrix product states and projected entangled pair states (oct 2014). arXiv:1306.2164, doi:10.1016/j.aop.2014.06.013.

[3] P. Corboz, Lecture 1: tensor network states, Tech. rep.

[4] 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.

[5] 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.

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

[7] S. Sachdev, Quantum phase transitions, Physics World 12 (4) (1999) 33–38. doi:10.1088/2058-7058/12/4/23.
URL https://iopscience.iop.org/article/10.1088/2058-7058/12/4/23https://iopscience.iop.org/article/10.1088/2058-7058/12/4/23/meta

[8] A. Taroni, Statistical physics: 90 years of the Ising model (dec 2015). doi:10.1038/nphys3595.
URL www.nature.com/naturephysics

[9] 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.

[10] 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.