Cluster Expansion of Thermal States using Tensor Networks

David Devoogdt

Academic year 2020-2021

Abstract

Simulating many-body quantum systems is one of the challenges faced by modern physics. One particular problem is constructing the operator $e^{-\beta \hat{H}}$, which is used to compute the partition function of a quantum models. This is needed to make predictions about the macroscopic behaviour of quantum systems. Of particular interest are the phase transitions. This dissertation introduces the rich field of Tensor Networks as a method to perform these calculations. The physics of strongly correlated matter is discussed. The main contribution of this work, the novel cluster expansions, are described, together with the computational tools to solve the equations efficiently. The results are promising. Large imaginary time steps can be taken. Calculating phase transitions of the Transverse Field Ising model in 2D produces results that can challenge recent developments in literature.

1 Overview

Understanding the many-body quantum problem remains a challenge. Strongly correlated matter has many interesting phases which are not yet understood, including high-T superconductors, topological ordered phases and quantum spin liquids [2]. One method to simulate these materials numerically are tensor networks (TN). This work introduces a new TN method to calculate the exponential of a Hamiltonian with local interactions. This allows to evolve a quantum state in time and calculate the macroscopic properties at a finite temperature. As an example, the new method is used to calculate phase transitions of the Transverse Field Ising (TFI) model.

2 Tensor Networks

2.1 MPS

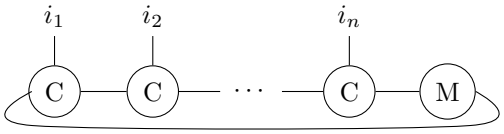
A general quantum state is given by

$$|\Psi\rangle = \sum_{i_1 i_2 \cdots i_n} C^{i_1 i_2 \cdots i_n} |i_1\rangle \otimes |i_2\rangle \otimes \cdots \otimes |i_n\rangle . \quad (1)$$

The tensor C needs d^n numbers to describe the full wave function, where d is the dimension of the local Hilbert space. Luckily, the relevant low energy states for many systems found in nature obey the so called 'area law', which only make up a very small corner of the full Hilbert space. In uniform matrix prodcut states (MPS), the tensor $C^{i_1 i_2 \cdots i_n}$ is subdivided into the product of n tensors C and a matrix M that contains the boundary conditions

$$C^{i_1 i_2 \cdots i_n} = Tr(C^{i_1} C^{i_2} \cdots C^{i_n} M). \quad (2)$$

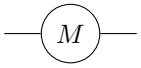
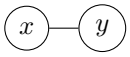
If M is the identity matrix, the chain is closed cyclically. Tensor networks are typically denoted in their graphical form (see table 1 for examples). External lines denote free indices, connected lines implies a summation over the shared index, analogous to matrix multiplication. Equation (2) is in this graphical notation becomes



(3)

An MPS has 2 dimension, the physical dimension of the particles d and the dimension χ of the bonds between the tensors. The cluster expansion will rely

Table 1: Examples of graphical notation.

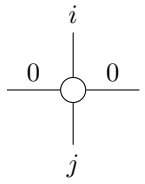
conventional	Einstein	tensor notation
$\vec{x}$	x_α	
M	$M_{\alpha\beta}$	
$\vec{x}\cdot\vec{y}$	$x_\alpha y_\alpha$	

on "virtual levels". This is the division of the MPS in blocks, analogous to dividing a matrix into block matrices. Every virtual level has its own associated dimension. The dimensions sum up to the total bond dimension χ.

2.2 MPO

A matrix procuct operator (MPO), is similar to an MPS but has 2 physical legs i and j. The following compact notation is used in this paper

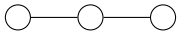
$$O^{00} = \quad = \bigcirc \,. \tag{4}$$



This is the MPO with virtual level 0 and physical indices i and j, which will both be omitted. Non-zero virtual indices are shown. Summation over shared virtual index 1 on 2 neighbouring sites is denoted as

$$O^{01}O^{10} = \bigcirc\!\!\overset{1}{-}\!\!\bigcirc\,, \tag{5}$$

while contraction over all possible virtual levels on 3 sites is denoted by

 . (6)

3 Cluster Expansion

The novel method to construct the operator $e^{-\beta\hat{H}}$ with cluster expansions was first introduced in [3]. The goal is capture the exponential of a Hamiltonian operator $\hat{H}$ of the form

$$\hat{H} = -J\left(\sum_{\langle ij\rangle} H_2^i H_2^j + \sum_i H_1^i\right). \tag{7}$$

This Hamiltonian consists of 1 and 2 site operators. The first summation $\langle ij\rangle$ runs over all neighbouring sites. The precise form of the Hamiltonian is not important. The notation for the contraction of the tensor network will also be used to denote the Hamiltonian evaluated on the given geometry

$$\begin{aligned} H\left(\bigcirc\!\!-\!\!\bigcirc\!\!-\!\!\bigcirc\right) =& H_1 \otimes 1 \quad \otimes 1 \\ &+1 \quad \otimes H_1 \otimes 1 \\ &+1 \quad \otimes 1 \quad \otimes H_1 \\ &+H_2 \otimes H_2 \otimes 1 \\ &+1 \quad \otimes H_2 \otimes H_2\,. \end{aligned} \tag{8}$$

3.1 Idea

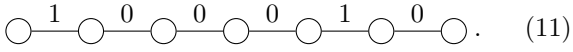
The main idea is to make an extensive expansion by adding blocks which solve the model exactly on a local patch. Crucially, the expansion is not in the inverse temperature β but in the size of the patches. The local patches are separated by a virtual level 0 bond. To make this somewhat more precise, the first steps of the expansion are shown here. The smallest patch, i.e. 1 site, encodes the exponential of that Hamiltonian

$$\bigcirc = \exp\left(-\beta H(\bigcirc)\right). \tag{9}$$

If there were no 2 site interactions, this already captures the full diagonalisation. The next step is to introduce 2 site interactions, where the one site interactions are subtracted from the diagonalised Hamiltonian.

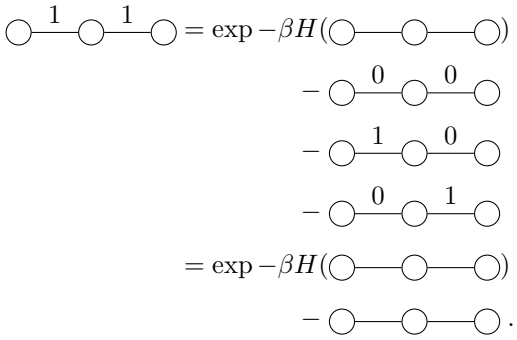
$$\begin{aligned} \bigcirc\!\!\overset{1}{-}\!\!\bigcirc = \exp -\beta H(\bigcirc\!\!-\!\!\bigcirc) \\ -\bigcirc\!\!\overset{0}{-}\!\!\bigcirc \end{aligned} \tag{10}$$

Contraction of larger network lead to many terms, such as

 . (11)

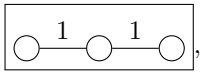
The beauty of this lays in the fact that disconnected regions(regions separated by level 0) combine in exactly the right way to capture all the terms appearing in the series expansion of the exact tensor exponential with as largest patch 2. In other words it is exact up to order 2 in β [3]. Only the terms of the exponential which acts on 3 or more neighbouring sites at once, are not accounted for.

Notice that in eq. (10), 2 new blocks are introduced. The dimension of virtual level 1 needs to be d^2, with d the dimension of physical level. Although the types already differ in the next step, one more blocks is shown to make the notation clear:



(12)

This is called a cluster expansion of order 3, because there are 3 connected sites solved exactly. The right-hand side of eq. (12) can be omitted, as it is just evaluating the exponentiated Hamiltonian on the same geometry as the left-hand side and subtracting all possible contractions of the blocks which were added previously. This very compact notation will be able to capture the essence of the different constructions. Because it is important for the remainder of the chapter, it is stressed that for an equation similar to



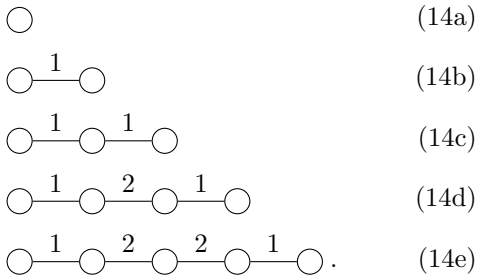
, (13)

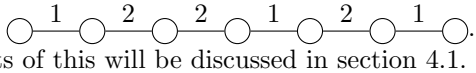
the right-hand side of eq. (12) is implied. This fully defines one new block as in eq. (12), or the contraction of 2 blocks as in eq. (10). In the following section, different types will be discussed. For every geometry, one or multiple new blocks are defined.

3.2 1D

3.2.1 Type A

The first few blocks in the cluster expansion are

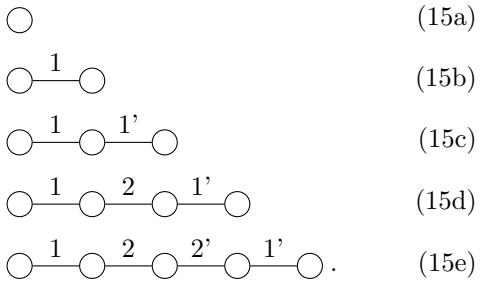


Virtual level l needs a bond dimension of d^{2l} to solve the equations exactly. The introduction of O^{nn} blocks could lead to long chains in the contraction, e.g. 

. The results of this will be discussed in section 4.1.

3.2.2 Type E

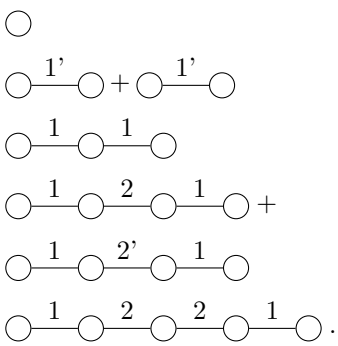
To remedy this behaviour, type E only has these blocks



The bond dimension χ is twice as large, because for every virtual level there is also a primed level. With these blocks, it is impossible to make a patch longer than the order of the largest chain. This generalises well to higher dimensions.

3.2.3 Type F

Both type A and F have potentially ill conditioned inverses. The blocks of type F are



(16a) (16b) (16c) (16d) (16e)

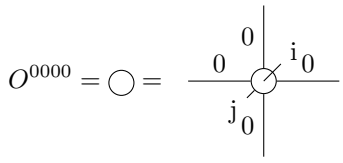
The blocks O^{nn+1} are chosen unitary up to a constant factor. The primed blocks solve the chains of even order. This requires twice the bond dimension of type A, but is guaranteed to have well conditioned inverses.

3.3 2D

In hindsight of the results, the construction in 2D is a generalisation of type A. The linear blocks, for which the defining equation is described by a tree graph, will resemble the blocks previously introduced. The non-linear blocks are used to account for loops.

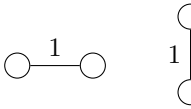
3.3.1 Linear blocks

The extension of an MPO to 2D is called a PEPO (Projected Entangled Pair Operator). These PEPO's are graphically depicted in this work by the same symbol

$O^{0000} =$ 

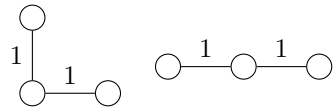
. (17)

The construction starts off as



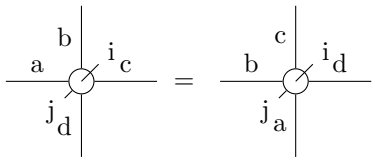
. (18)

For the order 3 blocks, 6 different options are possible. They are



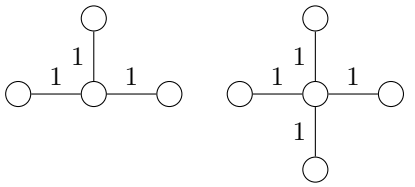
(19)

and their rotations over 90 degrees. Solving all blocks, imposing rotation symmetry over 90 degrees



(20)

reduces the number of blocks that need to be solved significantly. Also, $\perp$ and $+$ blocks should be added

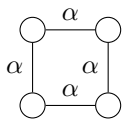


. (21)

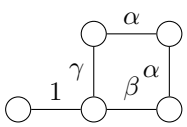
From here on, it can again be generalised to higher orders. Care has to be taken that for before solving a new patch, all smaller patches that fit the geometry are solved first.

3.3.2 Nonlinear blocks

Not all finite size patches are covered with the blocks introduced in the previous section. The lowest order blocks not covered are



(22)



. (23)

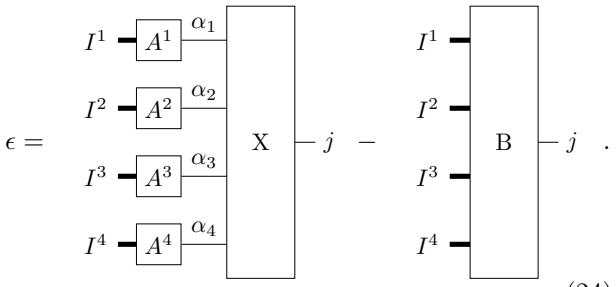
These are nonlinear equations, and will be treated separately in section 3.4. All virtual levels for solving loops are denoted by Greek letters. For $d = 2$, the bond dimension of α can be as low as 6. To connect the loops to the linear blocks, corner pieces similar to eq. (23) are used. Directly connecting virtual level 1 to the loops with $O^{\alpha\alpha 01}$ also solves the local

patch, but cause diverging results when 2 or more corner pieces with extensions connect. One corner can connect to 2 chains, with for each chain a maximum length equal to the order of the linear blocks. Going further is possible, but comes at an ever-increasing bond dimension cost.

3.4 Solvers

3.4.1 Linear Solver

Now that it is clear how the construction works, we focus on how to solve them numerically. Take as example $X = O^{1111}$ from eq. (21). The involved tensors are reshaped and reordered to bring it in the following standard form



(24)

Here B is the exponentiated Hamiltonian on the + geometry minus the contraction of all previous added blocks. ϵ is the residual error, and should be zero up to machine precision when the solver is finished. Solving this equation by inverting the matrices A^i separately results in numerical unstable results, due to the ill-conditioned blocks A^i inherent to the construction. In practice this instability starts happening at virtual level 2. Taking the pseudoinverse (with all singular values $\sigma_0 < 10^{-12}$ taken to be exactly zero) of the tensor product $A = A^1 \otimes A^2 \otimes A^3 \otimes A^4$ resolves the problem, but is computationally too expensive for long side chains. The problem is resolved by taking the SVD decomposition of each matrix $A^i = U^i \Sigma^i V^{i\dagger}$. The unitary matrices are inverted by taking the Hermitian transpose, and the matrix $\Sigma = \Sigma^1 \otimes \Sigma^2 \otimes \Sigma^3 \otimes \Sigma^4$ is pseudoinverted. This is fast because Σ is very sparse. The procedure also works for e.g. the corner pieces of eq. (23), and by extension every connected patch. When the patch contains 2 tensors such as eq. (14d), it is split with an SVD decomposition.

3.4.2 Nonlinear Solver

A nonlinear solver takes steps in the direction that lowers the residual error ϵ. This procedure can be sped up if the gradient is known. Inspection of eq. (24) shows that the derivative $\frac{\partial \epsilon_{(I^1 I^2 I^3 I^4)j}}{X_{(\alpha_1\alpha_2\alpha_3\alpha_1)j}} = A_{(I^1 I^2 I^3 I^4)(\alpha_1\alpha_2\alpha_3\alpha_4)}$. Or more simply put, the contraction of the network but with X removed. This is extended to more complex situation with the chain rule for derivatives. The solver handles multiple blocks which are identical up to a permutation.

3.4.3 Sequential Linear Solver

Another useful way to solve the nonlinear equations is by employing the linear solver described before, where each appearing tensor is solved individually. The algorithm keeps sweeping over all tensors until convergence is reached. It's necessary to choose a suitable step size.

4 Results

The cluster expansions are tested in 2 ways. In 1D and 2D, the error ϵ is calculated by comparing the contracted PEPO against the exact exponential. In 2D, some critical temperatures are cacluated for the TFI model.

4.1 Exact exponentials

In 1D, the error is calculated on a cyclic system of 11 sites. The results are shown in fig. 1. All the cluster expansion get better with increasing order. Type A outperforms the other 2 types by quite some margin. This conclusion is also true for the Heisenberg model and random 2 site Hamiltonians (not shown here). The results in 2D show a similar trend. The error in 2D is of the same order of magnitude. A plaquette term eq. (22) is needed, the extensions eq. (23) are optional. The results show that real time evolution ($t = -i\beta$) also performs well.

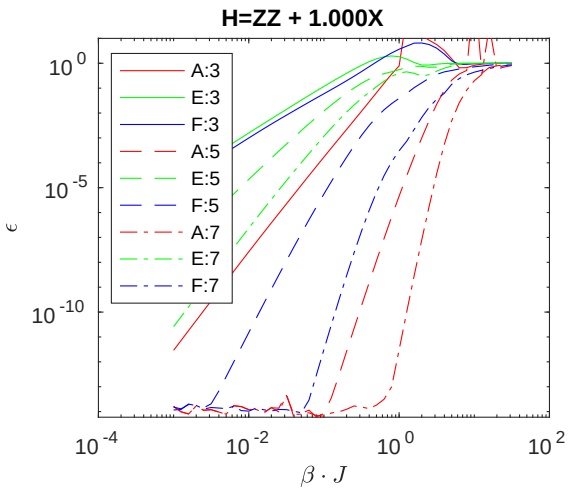


Figure 1: Comparison type A, E and F for TFI. Error evaluated on cyclic chain.

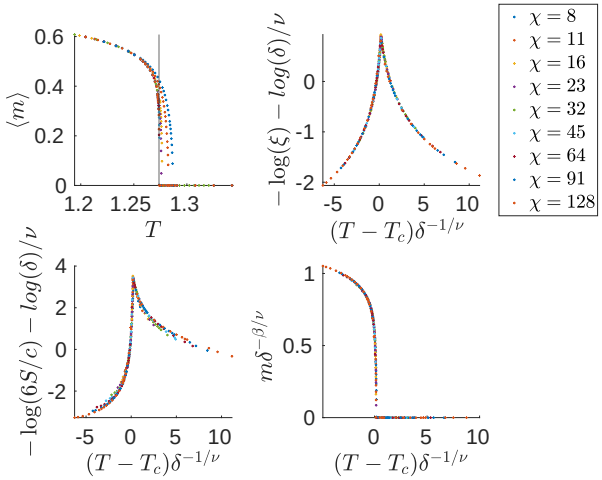


Figure 2: Data collapse for $g = 2.5$ phase transition of TFI Model. Data points are taken from $T \in [T_c - 0.08, T_c + 0.08]$.

4.2 2D TFI model

The TFI model is of the form eq. (7), with $H_1 = g\sigma^x$ and $H_2 = \sigma^z$. For low temperature, the spins tend to align forming a ferromagnet. For high T, the magnetisation disappears. The transversal field g causes the spins to align in the direction of the transversal field. A cluster expansion of order 5 with loops is used to simulate the phase transition of the TFI model at g=0 (classical Ising) and g=2.5. The simulated magnetisation $\langle m^z \rangle$ and the data collapse of m, entropy S and correlation length ξ for g=2.5 is shown in fig. 2. δ is a measure for the (inverse) size of the system. Figure 2 shows a very clear data collapse. For g=0, the fitted critical temperature is $T_c = 2.691(9)$, in agreement with Onsager's analytical solution $T_c = 2.69185$. For $g = 2.5$, the fitted value $T_c = 1.2736(6)$ agrees well with values from literature, e.g. $T_c = 1.2737(2)$ obtained with a competing tensor network technique and $T_c = 1.2737(6)$ obtained with quantum Monte Carlo. [1] Higher precision can be achieved with cluster expansions of higher order. The properties were calculated for infinite lattices using VUMPS.

5 Conclusion and Outlook

The results agree well with exact exponentiation and with critical values from literature, proving that these cluster expansion can compete with other methods. With the current framework, it is possible to simulate the quantum critical point of the Ising model. The linear blocks and plaquette term seem to be sufficient to make an accurate cluster expansion, making the generalisation to 3D within reach.

References

[1] Piotr Czarnik and Philippe Corboz. Finite correlation length scaling with infinite projected entangled pair states at finite temperature. *Physical Review B*, 99:245107, 2019.

[2] Román Orús. A practical introduction to tensor networks: Matrix product states and projected entangled pair states, oct 2014.

[3] Bram Vanhecke, Laurens Vanderstraeten, and Frank Verstraete. Symmetric cluster expansions with tensor networks. *Physical Review A*, 103(2), 2021.