On Simulating Lattice Gauge Theories with Globally Symmetric Tensor Networks Libraries

Manu Canals

Master's Thesis

Master of Science in Quantum Science & Technology
Ludwig-Maximilians-Universität München
Technical University of Munich

at the Faculty of Physics (LMU),

supervised by

Prof. Dr. Luca Tagliacozzo
Prof. Dr. Fabian Grusdt

in collaboration with

Prof. Dr. Jan von Delft

Munich, 27 October 2023

Über die Simulation von Gittereichtheorien mit global symmetrischen Tensor-Netzwerk-Programmierbibliotheken

Manu Canals

Masterarbeit
Master of Science in Quantum Science & Technology
Ludwig-Maximilians-Universität München (LMU)
Technical University of Munich (TUM)

an der Fakultät für Physik (LMU),

betreut von
Prof. Dr. Luca Tagliacozzo
Prof. Dr. Fabian Grusdt

in Zusammenarbeit mit
Prof. Dr. Jan von Delft

München, 27 Oktober 2023

Abstract

In locally invariant field theories, i.e. gauge theories, only a subspace of the Hilbert space contains physical states: those with (unbroken) local symmetry. The tensor-network variational *ansatz* from Tagliacozzo et. al [1] exclusively represents such states while preserving the entanglement area law, efficiently exploring the reduced Hilbert space. However, the locally symmetric *ansatz* has a sparse structure not directly compatible with symmetric libraries. We overcome this problem by *doubling* the symmetry, e.g. from $\mathbb{Z}_2$ to $\mathbb{Z}_2 \otimes \mathbb{Z}_2$, and benchmark the new *ansatz* for a $\mathbb{Z}_2$ lattice gauge theory. Additionally, we show that *all* physical states can be represented by this *ansatz*. Finally, tensor networks with loops, such as the one in this work, can have problematic internal correlations. We propose a solution to remove them from the tensor network.

Acknowledgments

First, I want to express my gratitude towards Prof. Dr. Luca Tagliacozzo. He has been my direct supervisor, guiding and advising me throughout the year while providing a supportive working environment. I sincerely appreciate the trust and confidence he has placed in me from my Bachelor's thesis to this Master's thesis. He offered me opportunities in the past and for my future that significantly impacted my professional trajectory.

I want to demonstrate my appreciation towards Prof. Dr. Jan von Delft, who collaborated with us during part of the thesis. He offered me a place at his Chair during the thesis, which allowed me to experience how a research group works from within and contribute to the project with stimulating discussions. For that, I am thankful. My gratitude extends to the members of the group with whom I have been in contact for a year. I want to mention Gün, Johannes, Frederic, Marcel and Ming.

My second supervisor, with whom we organised the thesis and planned a collaboration that unfortunately could not be materialised, was Prof. Dr. Fabian Grusdt. I want to acknowledge his help arranging the thesis and his support in providing material for the project and examining the thesis.

A fundamental role during this year was played by my environment in daily life, offering emotional support and mental well-being in crucial moments. I want to mention my friend Marc specifically, who truly made me feel at home and the complete opposite of loneliness, and my friends Emilio, Fred, Marwa, Mireia and Sali, in alphabetical order. A special note as well to my flatmates Adrian, Elena and Julian. Lastly, I owe so much to Albert, Edu, Laura and Mario from Barcelona, which I carry with me wherever I go.

I would not be here and be who I am without my family. They always provided a safe space, made me feel loved, and supported and stimulated my trajectory in every aspect of my life. As I grow, I realise more and more how precious this is. I am so proud and happy of the family I have. Vicent, Lola, Marcel i Martí (i Milo), us estimo.

Thanks,
Manu

Contents

Contents v

1 Introduction 1

2 Gauge Theory 7
- 2.1 The Model . . . 7
- 2.2 Physics of the Model . . . 8
- 2.3 Symmetries and Gauss' Law . . . 9
- 2.4 Conclusion . . . 11

3 Tensor Networks and Symmetries 13
- 3.1 Tensor Networks . . . 13
- 3.2 Symmetric Tensors and Libraries . . . 15
  - 3.2.1 Representation of a Group . . . 16
  - 3.2.2 $\mathbb{Z}_2$ Sum-rule . . . 17
- 3.3 Tensor Networks Algorithm . . . 18
  - 3.3.1 Evolution with Tensor Networks . . . 19
  - 3.3.2 Truncation . . . 21

4 *Ansatz* 23
- 4.1 *Ansatz* . . . 23
  - 4.1.1 Copy Tensor . . . 24
  - 4.1.2 Vertex Tensor . . . 25
- 4.2 Imposing Gauss' Law . . . 25
- 4.3 Conclusion . . . 27

5 $\mathbb{Z}_2 \otimes \mathbb{Z}_2$ Proposal 29
- 5.1 Non $\mathbb{Z}_2$ Symmetric *Ansatz* . . . 29
- 5.2 Proposed Solution . . . 30
  - 5.2.1 New Symmetry . . . 30
  - 5.2.2 Copy Tensors . . . 31

5.2.3 Vertex Tensors . . . 33
5.2.4 Revisit *Ansatz* . . . 34
5.3 Consequences . . . 35
5.4 Practical Example and Conclusion . . . 36

6 Reach of the *Ansatz* 39
6.1 Create Initial Gauge State . . . 39
6.2 Create General Gauge Preserving Operator . . . 41
6.3 All Gauge States . . . 42
6.4 Recover TN Structure . . . 43
6.4.1 Individual Terms . . . 43
6.4.2 Linear Combination . . . 45
6.5 Relate To the *Ansatz* Structure . . . 47

7 Benchmark Results 49
7.1 Infinite Case . . . 49
7.1.1 Convergence . . . 50
7.1.2 Accuracy . . . 51
7.1.3 Adaptive Evolution . . . 52
7.1.4 *Bouncing* Off the Ground State . . . 54
7.2 Finite 2×2 Lattice . . . 56
7.2.1 *Bouncing* Off the Ground State . . . 56
7.2.2 Resetting Evolution . . . 57
7.2.3 Known Fix-point . . . 57
7.3 Candidate Problem . . . 59

8 Internal Correlations 63
8.1 Internal Correlations . . . 64
8.2 Finding Internal Correlations . . . 66
8.2.1 Rephrase the Problem . . . 67
8.3 Proposed Solutions . . . 68
8.3.1 Disentangle Problem . . . 68
8.3.2 Find Y Manually . . . 69
8.4 Find Permutation . . . 71
8.4.1 Strategy . . . 72
8.4.2 Steps . . . 73

9 Summary and Outlook 75

A Advanced Concepts in TN 77
A.1 Applying the Plaquette Term . . . 78
A.2 Energy Calculation . . . 80
A.3 Gauge Freedom in TN . . . 81

B Derivations of the Proof **83**
B.1 First Derivation 83
B.1.1 Assume $[\theta, A_i] = 0\ \forall i$ 83
B.1.2 Assume θ is gauge preserving 84
B.2 Second Derivation 84

Bibliography **87**

Chapter 1

Introduction

Many-body problems appear in many areas of physics and other fields of study. Solving these problems is a challenging task. Computational difficulties quickly appear if one works directly with quantum many-body states. The number of parameters necessary to describe them grows exponentially fast with the system's size. To express a state of an N spin s system, one needs $(2s+1)^N$ coefficients. Already for ~ 50 spins $1/2$, storing 2^{50} coefficients in a computer is rather impossible. In fact, for systems of macroscopic sizes like Avogadro's number $\sim 10^{23}$ of constituents, the amount of coefficients in a general state is exponentially larger than the estimated number of atoms in the observable universe!

This limitation can be avoided by performing certain approximations on the problem, for example, a mean-field approximation. In this case, the many-body problem is reduced to a one-body problem, and the state of the system can be written as a product state $|\Psi_1\rangle \otimes \ldots \otimes |\Psi_N\rangle$. This state can be described with just $(2s+1)N$ coefficients. Even if restricted, this model is valid to explain to some extent quantum many-body phenomena like phase transitions [2] or superconductivity [3], to give some examples.

This fact —that in some cases, product states succeed at describing physics— hints at the possibility that a state of physical interest is not just any random state of the Hilbert space. These interesting states are typically ground or excited states of a local Hamiltonian. Inspired by Ref. [4], consider states in thermal equilibrium at temperature T, which are completely described by $\rho \propto e^{-\mathcal{H}/T}$. In most or all known systems, the Hamiltonian $\mathcal{H}$ includes at most k-body interactions with finite k which typically k=2. Therefore, the operator $\mathcal{H}$ can be completely parametrized with $(N,k)\times(2s+1)^{2k}$ coefficients, where (N,k) is the number of groups of k spins and a general Hamiltonian acting on k spins has $(2s+1)^{2k}$ components. In other words, equilibrium states in nature, ρ, need a few parameters to be described instead of the exponentially many

needed for general states. We can phrase this as follows: states of physical interest live in a *corner* of the Hilbert space.

We see an efficient representation of these states could be possible using the previous information. The next non-trivial natural step is to find such representation. Starting in the 90s, results from quantum information theory further characterised this corner of the Hilbert space, laying strong theoretical foundations for efficient representations. Notably, for Hamiltonians with finite-range interactions and a gap, the ground state was found to fulfil the area law of entanglement, as reviewed in Ref. [5]. This law is formulated for a partition of a system into two regions. It states that the entanglement between the two regions scales with the border's size and not with the volume of the regions[1]. The area law is slightly modified depending on the model's gap, temperature and fermionic constituents [6, 7]. In summary, states from the corner of the Hilbert space have this particular defining entanglement properties, which can be utilised to find an efficient representation of states.

Tensor Network (TN) states are precisely built to fulfil these entanglement properties, thus belonging to the reduced corner of the Hilbert space, i.e. states of physical interest. These states are represented as a network of contracted tensors. Tensors and their contractions are explained in Ch. 3. For now, two contracted tensors can be thought of visually as two nodes connected through a bond. Typically, one tensor (node) is associated with each physical site and is contracted (connected) with its neighbouring tensors. Consider now a border that separates the system into two regions. The number of bonds split by the border grows with the border's size. Crucially, the entanglement between the regions is directly proportional to the number of split bonds. Therefore, the entanglement also grows with the border's size, fulfilling the area law.

One of the most known and studied TN state is the matrix product state (MPS), a 1D train of 3-legged tensors contracted through two of their legs. Its introduction was related to the density matrix renormalisation group (DMRG) [8]. This method, introduced in 1991, reached extraordinarily precise results and was extensively used for 1D many-body problems [9, 10]. In 1995, Ostlund and Rommer realised DMRG results could be expressed as an MPS [11, 12], although this term was only coined later [13]. The success of DMRG extended to TN, making the MPS the first successful TN. Afterwards, other families of TN states followed. A generalisation to higher dimensions was introduced in 2004 [14, 15], the so-called projected entangled pair states (PEPS). See Ref. [16] for a modern review. Tree Tensor States [17] and Multidimensional Entanglement Renormalisation Ansatz [18, 19] are other TN states which can accommodate other entanglement properties, e.g. needed for critical systems.

[1] The latter case would be expected for general states.

With these TN tools, working directly with states of the corner Hilbert space is no longer computationally impossible. That allowed the development of many numerical methods in the last couple of decades, for example, the implementation of real or imaginary time evolutions with the infinite time-evolving block decimation algorithm (iTEBD) [18, 20, 21], the development of renormalisation algorithms through a coarse-graining of the TN [4, 22], or the use of the tangent space of an MPS manifold [23, 24]. See Ref. [25] for a more extensive review. Connecting these to the present work, here we use a variation of the PEPS to implement an imaginary time evolution similar to iTEBD.

In recent years, a particular research direction on TN has been its applications in lattice gauge theories. See Ref. [26] for a broad review. For our purposes, gauge theories are field theories invariant under local transformations, called *gauge* transformations. The most prominent gauge theory is the Standard Model, a non-Abelian gauge theory describing three of the four fundamental forces. The theory is invariant under the local action of elements of the $U(1)\times SU(2)\times SU(3)$ group, giving rise to the electromagnetic, weak and strong interactions, respectively [27]. Moreover, gauge theory can also be used to study condensed matter phenomena like high-T_c superconductivity, Bose-Einstein condensation or superfluidity [28–31]. Gauge theories formulated on a lattice, i.e. lattice gauge theories (LGT), were developed in the 70s when Wilson addressed the problem of the confinement of quarks [32]. Using Lagrangian mechanics, he generalised Wegner's Ising gauge theory [33] to continuous, Abelian gauge groups. Later that decade, Kogut and Susskind re-expressed these lattice gauge theories in the Hamiltonian formalism [34]. At the same time, Creutz used transfer matrix formalism of statistical mechanics to relate those two approaches [35]. The model of this thesis is based on the review of 1979 by Kogut [36].

As stated in Elitzur's theorem [37], local gauge invariancy cannot be spontaneously broken on physical states. Therefore, physical states have several constraints they must fulfil. For example, these restrictions are equivalent to fulfilling Gauss' law around each vertex in the model considered in Ch. 2. Consider the states of physical interest in gauge theories, i.e. low-energy constrained states, and their entanglement properties. A priori symmetry constraints could make these states more entangled, modifying the area law presented before and not allowing the application of the TN machinery. However, these constraints can be formulated as the low-energy sector of a local spin model, i.e. these constrained physical states still are the low-energy sector of a (gapped) local Hamiltonian, for which the area law applies. Indeed, one can consider a $\mathcal{H}_{\mathrm{SYM}}$ such that its (degenerated) ground states fulfil the previous symmetry constraints and $[\mathcal{H}_{\mathrm{SYM}}, \mathcal{H}_{\mathrm{LGT}}] = 0$, where $\mathcal{H}_{\mathrm{LGT}}$ describes the system —see Kitaev's toric code model [38]—. Then, the low-energy sector of $\mathcal{H}=\mathcal{H}_{\mathrm{LGT}}-a\mathcal{H}_{\mathrm{SYM}}$ for $a\gg 1$ is the degenerated constrained ground state's subspace of $\mathcal{H}_{\mathrm{SYM}}$, effectively describing $\mathcal{H}_{\mathrm{LGT}}$ with the symmetry constrains.

The area law applies to this local gapped Hamiltonian $\mathcal{H}$ and consequently, TN tools can be used to describe the Hilbert space's corner of lattice gauge theories.

The review by Bañuls and Cichy [26] comments on many studies of TN applied to LGT, some of which we mention here. For one spatial dimension, the first explicit application of the MPS formalism was in 2005 for a $\mathbb{Z}_2$ model on a spatial ladder. However, before the full TN formalism was developed, DMRG had already found success in 2002 for the Schwinger QCD toy model [39]. This same model was later studied fully in MPS formalism by Bañnuls in 2013 [40]. The first application of TN to two spatial dimensions LGT was done by Tagliacozzo and Vidal in 2010 [41]. They presented the first explicit gauge-invariant TN *ansatz* and reproduced the known phase diagram of the considered $\mathbb{Z}_2$ system with it. A couple of years later, Taglicozzo et al. presented a general *ansatz* for pure gauge LGT and any discrete or continuous group [1]. Other 2D *ansätze* with different approaches were later proposed [42–44] and techniques combining Monte Carlo methods [45, 46] and renormalisation groups were studied [47].

This thesis focuses on the *ansatz* introduced by Taglacozzo et al. [1] for a $(2+1)D$ $\mathbb{Z}_2$ LGT. We specifically consider time evolution on an infinite lattice, diverging from recent works that explore various scenarios like finite sizes, variational Monte Carlo methods or exact diagonalisation [42–47]. A comprehensive review of the *ansatz* is presented in Chapter 4. Essentially, the *ansatz* involves the assignment of tensors to each vertex and link of a square lattice. However, only the link tensors act on a physical site. Each vertex tensor is contracted with the four neighbouring link tensors, rendering them 4-legged tensors. Accordingly, link tensors have three legs, one physical and two virtual. Similarly to PEPS, this construction fulfils the area law. Notably, the implementation of Gauss' law on the vertex tensors ensures that only physical states are represented by the TN.

Consider the explicit effect of symmetries on the TN [48, 49]. Symmetries of the system are reflected in the components of the TN's tensors. On the one hand, global symmetries impose sum-rules that the components must fulfil. Otherwise, they must be zero, as explained in Sec. 3.2. This fact implies that tensors of symmetric TN are sparse and that computational advantages can be gained. There exist multiple numerical libraries that efficiently implement symmetric tensors. We used QSpace [50] for the numerical calculations of this work, provided by Prof. Dr. Jan von Delft and which is extensively used in his Chair. On the other hand, TN with local symmetries, such as the considered *ansatz*, have tensors with components which do not directly fulfil the sum-rules from global symmetries. Accordingly, symmetric libraries mentioned before are not directly compatible with TN, such as the considered *ansatz*. This implies that we cannot naively use those generic libraries and

might be forced to hard-code local symmetries in our code if we still want to leverage the sparse structure of the tensors. We present in this work a way of circumventing this issue, making the *ansatz* and possibly other locally symmetric TNs compatible with generic symmetric tensor libraries. The main idea is to consider these local invariant TNs under another symmetry, namely, the $\mathbb{Z}_2$ *ansatz* is regarded as a $\mathbb{Z}_2 \otimes \mathbb{Z}_2$ *ansatz*.

Organisation of the contents. Chapter 2 introduces the $\mathbb{Z}_2$ lattice gauge theory model used in this study. There, a physical interpretation is given, which connects the physical subspace with Gauss' law. In Ch. 3, some basics of TN and its symmetric (sparse) cases are reviewed. The imaginary time evolution algorithm used to find the ground state is also explained in that chapter, although technical details of its implementation are left in Appendix A. The *ansatz* from Tagliacozzo et al. [1] is presented in Ch. 4, using TN notation and the explanation of Gauss' law from the previous chapters. The first of four results obtained in this thesis is presented in Ch. 5, where the *ansatz* is expressed with another symmetry, $\mathbb{Z}_2 \otimes \mathbb{Z}_2$, to overcome its previous incompatibilities. Chapter 6 contains an additional theoretical result, where it is shown that all physical states *can* be represented by the *ansatz*. The ground state search benchmark results are displayed in the following chapter, Ch. 7. Its limitations are commented on and a candidate problem in the simulation is considered: internal correlations. This issue is explained in more detail in Ch. 8, where an independent proposed solution is included. This solution is the fourth and last result of the thesis. In Ch. 9, a summary and an outlook conclude this work.

Reader Guidelines. Chapter 2 and 3 can be omitted for readers experienced in those toppics. The following chapter introduces the *ansatz* making use only of Eq. (3.11) for state's notation and the derivation of Gauss' law of Eqs. (2.11−2.13). Furthermore, the *ansatz* reformulation in Ch. 5 should not require additional concepts.

The theoretical result in Ch. 6 and the numerics from Chs. 7 and 8 are independent of each other, except for one technical comment on Sec. 6.4.1. Therefore, one may jump directly to numeric results. The benchmark results of Ch.7 can be followed considering the Hamiltonian from Eq. (2.3).

Finally, even though considering internal correlations is motivated throughout Ch. 7, our formulation of the problem and proposed solution are an almost independent chapter of this thesis. It may only require visiting Appendix. A.3 to review TN's gauge freedom.

Chapter 2

Gauge Theory

Gauge theories are distinguished from other theories because of their local symmetries. For the purpose of this thesis, a gauge theory is a field theory which is invariant under local transformations. These local transformations are also called *gauge* transformations.

Not only is the theory (model) symmetric, but also the physical states. Elitzur's theorem states that local gauge symmetries can not be spontaneously broken [37]. Contrary to global symmetries —where, in some cases, the physical state can spontaneously break the symmetry, regardless of the symmetry of the theory—, local gauge symmetries must be present in the physical state as well. This restriction on the local transformations gives rise to familiar laws, like Gauss' law, in the context of gauge theories, as expressed in Eq. (2.13).

In this work, we use the lattice approach to gauge theories, i.e. lattice gauge theory [32–36]. Concretely, we use the model presented in Sec. V.E of Kogut's 1979 review [36]. The model's 2D Hamiltonian is given in Eq. (2.3). As shown in the review, under a particular limit of the parameters, the model is equivalent to the (2+1)D Ising gauge theory [33] and, to some extent, it exhibits a resemblance to electrodynamics.

In this chapter, the model is introduced firstly in Sec. 2.1 and then a physical interpretation is given in Sec. 2.2. In the following section, Sec. 2.3, local symmetries are addressed —resulting in Gauss' law— and the model is slightly modified to accommodate for static background charges. Section 2.4 contains a short conclusion and connection to the following work.

2.1 The Model

We give now the model. Consider an infinite 2D square lattice with vertices i, links l and plaquettes p. Plaquettes are the basic squares of the lattice, formed

by four links. Let $\mathcal{V}$, $\mathcal{L}$ and $\mathcal{P}$ be the set of all vertices, links and plaquettes of the lattice, respectively. Finally, we use the following notation in future expressions,

$$l \in p: \quad \text{links forming the plaquette } p, \tag{2.1}$$

$$l \in i: \quad \text{links emanating from vertex } i. \tag{2.2}$$

On the model, one spin 1/2 site is placed on each link of the infinite lattice while vertices remain empty[1]. Accordingly, there is a 2-dimensional Hilbert space on each link l, where the usual Pauli matrices σ_x^l, σ_y^l and σ_z^l may act on. The Hamiltonian is the following[2], with real couplings J and h,

$$\mathcal{H} = -J \sum_{p \in \mathcal{P}} B_p + h \sum_{l \in \mathcal{L}} \sigma_x^l \quad \text{with} \quad B_p = \prod_{l \in p} \sigma_z^l \quad , \tag{2.3}$$

and where B_p is a plaquette operator.

The model presented here is one of the simplest, a $\mathbb{Z}_2$ lattice gauge theory. That allows us to focus on the technical side of manipulating the model while still displaying a simplified version of the more complex physics, e.g. a simplified version of the electromagnetic field and Gauss' law, Sec. 2.2.

It is worth noticing that there exist several options to formulate a model on this lattice. Sites could be placed at vertices, links or both, and the local system assigned to each site could range from various types. Each possibility can give rise to different physics —with distinct local gauge symmetries— when identifying the degrees of freedom of the sites with physical quantities.

An example of such identification is given in Sec. VI.A and VI.C of Kogut's review [36]. There, angular variables are placed on the links and identified with the vector potential of electrodynamics, while its conjugate momentum is identified with the electric field. Then, the model gives rise to conventional electrodynamics. The next section presents a similar physical interpretation for the considered model.

2.2 Physics of the Model

Similar to the identification from Kogut's review, one identifies eigenstates of the Hamiltonian's operators with physical quantities. In other words, each operator is identified with a physical property, an observable. Accordingly, its eigenstates represent states with a concrete value of that property, i.e. the eigenvalue. Consider writing the eigenstates of σ_x on each link as

$$\sigma_x |a\rangle = (-1)^a |a\rangle \quad , \quad a \in \{0, 1\} \longleftrightarrow E \ , \tag{2.4}$$

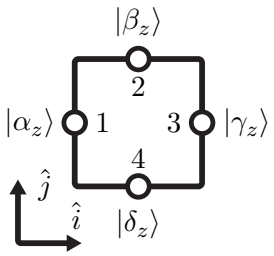
[1] Vertices are slightly modified in Sec. 2.3 with the addition of static background charges.
[2] Differs from the one deduced in Kogut's review[36] on the couplings.

and identify $|a\rangle$ as a discrete electric field E which can only take two values. Then, the unbroken local invariance —discussed below— imposes a simplified Gauss's law on this electric field. Furthermore, the h term on the Hamiltonian accounts for the energy of this electric field.

Continuing with the identification, the operator σ_z at a link on the horizontal direction $\hat{i}$ has eigenstates

$$\sigma_z |b_z\rangle = (-1)^{b_z} |b_z\rangle \ , \quad b_z \in \{0,1\} \longleftrightarrow A_i \tag{2.5}$$

which is identified as the *horizontal* component of a discrete vector potential taking only two values. Similar for the vertical direction $\hat{j}$. In this case, the plaquette operator B_p is related to the magnetic field at the direction perpendicular to the lattice, $\tilde{B}_k$. Consider its action on a lattice's plaquette which is in a product state of σ_z's eigenstates, $|\phi\rangle := |\alpha_z\rangle \otimes |\beta_z\rangle \otimes |\gamma_z\rangle \otimes |\delta_z\rangle$,



$$\begin{aligned} \sigma_z^1\sigma_z^2\sigma_z^3\sigma_z^4 |\phi\rangle &= (-1)^{\alpha_z+\beta_z+\gamma_z+\delta_z} |\phi\rangle && \overset{1)}{=} \\ &= (-1)^{(\beta_z-\delta_z)-(\gamma_z-\alpha_z)} |\phi\rangle && \overset{2)}{=} \\ &= (-1)^{\partial_j A_i - \partial_i A_j} |\phi\rangle && \overset{3)}{=} (-1)^{\tilde{B}_k} |\phi\rangle \, . \end{aligned} \tag{2.6}$$

On 1), we used $(-1)^{-\alpha} = (-1)^{\alpha}$ —in other words, exponent's summation is modulo 2—; on 2), we identified the eigenstates with the vector potential's components; and 3) follows from the electromagnetic tensor's spatial component $F_{ji} := \partial_j A_i - \partial_i A_j = \tilde{B}_k$, considering a lattice spacing of 1[3]. Therefore, the J term in the Hamiltonian of Eq. (2.3) accounts for the energy of the magnetic field.

These identifications are very simplified and far from the more complex objects from electromagnetism or other gauge theories. However, this connection still illustrates how these more complex models may work when introducing richer local systems and more complex Hamiltonians, producing more elaborated symmetries.

2.3 Symmetries and Gauss' Law

One can check that the Hamiltonian of Eq. (2.3) is locally invariant under some local gauge transformation. Consider the *star* operator[4] defined at each

[3] To make the identification more formal, the lattice spacing δ should be included in the eigenvalues ${}^{b_z}/_{\delta}$, and then the $\delta \to 0$ should be considered

[4] Not to be confused with the vector potential. The vector potential is only referred to in the previous section, while the star operator appears throughout the rest of the work.

vertex $i \in \mathcal{V}$ using the notation on Eq. (2.2),

$$A_i := \prod_{l \in i} \sigma_x^l \ . \tag{2.7}$$

Indeed, one can check that $[\mathcal{H}, A_i] = 0, \ \forall i \in \mathcal{V}$. Notice that the h term is trivial, and the J term also commutes because A_i and B_p always coincide on either zero or two sites.

Now, we can express Elitzur's theorem as follows: the only physical states of the Hilbert space are those invariant under A_i, i.e.

$$\{|\Psi\rangle \ | \ A_i |\Psi\rangle = |\Psi\rangle \ \ \forall i \in \mathcal{V}\}. \tag{2.8}$$

This means that the physical vector space is only a corner of the total Hilbert space[5]. It is precisely this realisation which offers an advantage for numeric calculations. In Ch. 4 an *ansatz* is presented which exploits this fact. By construction, it only represents physical states. The main focus of this work is precisely the efficient manipulation of the *ansatz*.

The condition on Eq. (2.8) can be identified with a simplified Gauss' law. Considering the product state $|\eta\rangle$ of the eigenstates of σ_x the invariance reads,

$$\begin{aligned} A_i |\eta\rangle &= A_i(\dots \otimes |a\rangle \otimes |b\rangle \otimes |c\rangle \otimes |d\rangle \otimes \dots) |\eta\rangle = \\ &= (-1)^{a+b+c+d} |\eta\rangle \overset{!}{=} \\ &\overset{!}{=} |\eta\rangle = (-1)^0 |\eta\rangle \quad \Longrightarrow \quad a+b+c+d = 0 \ (\text{mod } 2) \ . \end{aligned} \tag{2.9}$$

The sum modulo 2 of the variables around a vertex i is the lattice version of the divergence. Making explicit the identification of a, b, c and d with the electric field E, we write

$$\nabla \cdot E = 0, \tag{2.10}$$

which is a simplified version of Gauss' law in the vacuum. Remarkably, we recovered the familiar law of Eq. (2.10) solely from the unbroken local gauge symmetry condition.

We get a more general version of this law by slightly modifying the model. Notice that states from the model, Eq. (2.3), represent systems with no matter: we identified the links with (gauge) fields in Eq. (2.4) and Eq. (2.5). Matter is typically introduced in the vertices, which are currently empty.

We now introduce static charges on the vertices by assigning a label $g_i \in \{0, 1\}, \ \forall i \in \mathcal{V}$, related to the absence (0) or presence (1) of charge. When

[5] This corner's size still grows exponentially with the size of the lattice.

doing it rigorously[6], Elizibur's theorem selects a new Hilbert subspace as physical. Namely, the physical states are those from

$$K := \{|\Psi\rangle \ | \ A_i |\Psi\rangle = (-1)^{g_i} |\Psi\rangle \ , \ \forall i \in \mathcal{V}\} \ , \tag{2.11}$$

for a given set of static charges, $\{g_i\}_{i\in\mathcal{V}}$. We denote this subspace K as *Gauss sector* and the static charges as *background charges*[7]. In this subspace, the condition on $|\eta\rangle$, following a reasoning similar to Eq. (2.9), reads

$$A_i |\eta\rangle = (-1)^{a+b+c+d} |\eta\rangle \stackrel{!}{=} (-1)^{g_i} |\eta\rangle \implies a+b+c+d = g_i \ (\text{mod } 2) \ . \tag{2.12}$$

Identifying the electric field again, it is equivalent to the lattice version of

$$\nabla \cdot E = \rho \ , \tag{2.13}$$

Gauss' law in the presence of matter. Only states that fulfil this version of Gauss' law are physical states.

2.4 Conclusion

In summary, the model — the Hamiltonian of Eq. (2.3) and the background charges— presents a local symmetry, Eq. (2.7), and Elitzur's theorem reduces the physical Hilbert space to a corner of the total Hilbert space, Eq. (2.11). The variational *ansatz* presented in Ch. 4 was constructed by Tagliacozzo et al. in such a way that it only explores the states of physical interest (which fulfill the area law) of the physical corner of the Hilbert space [1]. That is, every state the *ansatz* represents fulfils Gauss' law, Eq. (2.12), for a fixed set of (static) background charges.

The *ansatz* is given in the formalism of Tensor Networks (TN). Such formalism is introduced in Ch. 3 as a preparation for introducing the *ansatz*. Many works have been done in the study of lattice gauge theories in the TN formalism, as reviewed in Ref. [26], although not so many in 2 spatial dimensions, like the current work. This thesis aims at contributing to this field of lattice gauge theories in the TN formalism by improving the *ansatz*, re-expressing it in a form apt for numerical optimisations.

[6] This method is reviewed in [51]. For the current model, spin 1/2 are introduced on the vertices. Eigenstates of the Pauli operator X on the vertex are identified with the presence or absence of charge. Moreover, matter-gauge interaction terms are included in the Hamiltonian. Importantly, this implies that $\mathcal{H}$ is invariant under a *new* local gauge transformation. Therefore, Elitzur's theorem selects a *new* physical subspace. To demote the charges from dynamics to (not evolving) static charges, we project the system to a subspace with fixed values of the charges, these values are the future labels g_i. Then, the resulting Hamiltonian is Eq. (2.3) plus constant terms and new parameters. Remarkably, the newly selected physical Hilbert space is unaffected by the projection, leading to a new Gauss' law, Eq. (2.13), explained in the main text.

[7] In LGT literature, the term static charges is preferred, and accordingly, it was used to introduce gauge theories in this work. To emphasise the role of background sources or sinks of the electric field, we call them differently.

Chapter 3

Tensor Networks and Symmetries

3.1 Tensor Networks

Tensors are linear maps T of the form

$$T: \; U_1 \otimes \cdots \otimes U_M \longrightarrow V_1 \otimes \cdots \otimes V_N \; , \tag{3.1}$$

where U_m and V_n are vector spaces with dimensions $D_{U,m}$ and $D_{V,n}$, respectively, for $m \in \{1, \dots \; M\}$ and $n \in \{1, \dots \; N\}$. Let $\mathcal{D}$ be the domain and $\mathcal{D}'$ be the codomain of T. We can express T using the bra-ket notation and a basis of $\mathcal{D}$ and $\mathcal{D}'$,

$$T = \sum_{u_1=1}^{D_{U,1}} \cdots \sum_{v_N=1}^{D_{V,N}} T^{v_1 \dots v_N}_{u_1 \dots u_M} \, |v_1\rangle \otimes \cdots \otimes |v_N\rangle \, \langle u_1| \otimes \cdots \otimes \langle u_M| \, , \tag{3.2}$$

where $\{|u_m\rangle\}_{u_m=1}^{D_{U,m}}$ is a basis of U_m, $m \in \{1, \dots M\}$. Similar for V_n, $n \in \{1, \dots N\}$. The components $T^{v_1 \dots v_N}_{u_1 \dots u_M}$ completely determine the tensor T.

In practical terms, one can think of these tensor components as a multidimensional array. When programming, labels $\{u_1, \dots, v_N\}$ have to be ordered in a certain way. Then, the tensor components are stored in an array A with $R := M + N$ indices, e.g.

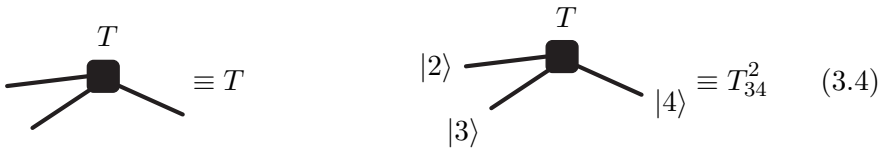
$$T^{v_1 \dots v_N}_{u_1 \dots u_M} = A(u_1, v_1, u_2, v_2, \dots v_N) \; , \tag{3.3}$$

where a certain order of the indices has been chosen. This array has dimensions $D_{U,1} \times D_{V,2} \times \ldots \times D_{V,N}$. Choosing an order of the indices is necessary to jump from formal expressions and diagrams to programming language, i.e. arrays. Therefore, keeping track of the convention used to choose this order

is important. In this text, the order used in each case is not specified, as it doesn't affect the explanations, and results can be reproduced with any convention chosen by the user.

Consider now the labels of the vectors, e.g. $u_m \in \{1, \dots D_{U,m}\}$. They enumerate the basis vectors. We use them to label the tensor's components $T^{v_1 \dots v_N}_{u_1 \dots u_M}$ following the convention "bra labels down" and "ket labels up". They distinguish domain and codomain. As shown in Sec. 3.2, we can label the basis vectors differently: with a charge a and a degeneracy label α, and write $|a;\alpha\rangle$. In this case, our notation changes to $T^{a \dots b}_{\alpha \dots \beta}$, no longer distinguishing domain and codomain. Nonetheless, they can still be seen as multidimensional arrays in a practical sense.

Now that a tensor T has been defined, we give the diagrammatic notation. Firstly, a geometric shape is associated with each tensor. Then, $R := M + N$ lines attached to it, so-called *legs*, represent its vector spaces[1] $U_1, \dots, U_M$, $V_1, \dots, V_N$. See Eq. (3.4). Because of the connection between legs and vector spaces, we use both terms interchangeably whenever there is no possibility of confusion. Additionally, tensor components are represented by writing each label in $T^{v_1 \dots v_N}_{u_1 \dots u_M}$ as a ket[2] next to the corresponding leg.



(3.4)

Usually, legs associated with the domain or codomain are distinguished. Legs have a direction —depicted with an arrowhead— depending on the vector space's type, following some convention. Because of the same distinction, legs in the diagrams of the components include bras as well as kets. Because of the $\mathbb{Z}_2$ symmetry, not making this distinction in this work results in equivalent formulations, as explained in Sec. 3.2.

The diagrammatic representation proves especially useful when considering the composition of tensors. To illustrate it, consider vector spaces $A, B, \mathcal{C}, D, E$ with dimensions $D_A, \dots, D_E$, respectively. Consider as well the tensors P and Q, which share some vector space, $\mathcal{C}$ in our example. Let R be their composition *through* the (shared) vector space $\mathcal{C}$,

$$P : A \to B \otimes \mathcal{C} \ , \qquad Q : \mathcal{C} \otimes D \to E \ , \qquad R : A \otimes D \to B \otimes E \ . \tag{3.5}$$

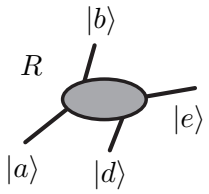
[1] This jump from the formal expression to the diagram needs a specific convention when assigning legs to vector spaces.

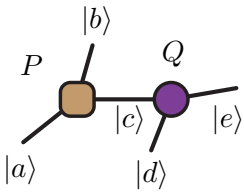
[2] In other conventions, only a label is written next to the leg. We choose kets instead to not confuse them with other labels and to remark that the label represents a state.

Their components are, respectively, P_a^{bc}, Q_{cd}^e, R_{ad}^{be}, where a, b, c, d, e are integer labels ranging from 1 to $D_A, \ldots, D_E$, respectively. The components of the composition R are given by the *contraction* of the components of P and Q:

$$R_{ac}^{be} := \sum_{c=1}^{D_c} P_a^{bc} \cdot Q_{cd}^e \ . \tag{3.6}$$

The summed label c corresponds to the vector space through which the composition is performed, i.e. $\mathcal{C}$. This is generally called "contracting tensors P and Q through $\mathcal{C}$". Diagrammatically, the contraction of P and Q through the vector space $\mathcal{C}$ is represented connecting the legs of P and Q associated with $\mathcal{C}$,

$R_{ad}^{be} =$ 

$=$ 

$\equiv \sum_{c=1}^{D_C} P_a^{bc} \cdot Q_{cd}^e \quad .$ (3.7)

Contracting (connecting) several tensors which are in a certain layout forms a network of tensors, i.e. a Tensor Network (TN). See Eq. (3.24) for an elaborated example. The whole contraction of a TN gives a single tensor or scalar, which can be associated with a state for example. Indeed, one can see a state as a map from a field to the state's Hilbert space expressed as

$$|\Psi\rangle = \sum_{i_1,\ldots,i_M} T^{i_1 \ldots i_M} \ |i_1\rangle \otimes \cdots \otimes |i_M\rangle \ , \tag{3.8}$$

and $T^{i_1 \ldots i_M}$ could be the result of the full contraction of a TN. Then we say that the TN represents that state. In the 90s and 00s, it was found that states of physical interest —such as ground state, excited states and thermal states— could be represented with TN [11–16], allowing the development of new numerical algorithms, such as the one presented in Sec. 3.3.

3.2 Symmetric Tensors and Libraries

Due to the symmetries of the system, e.g. the Hamiltonian being invariant under the action of matrices representing the $\mathbb{Z}_2$ group, the tensors of the TN representing a state of such system must fulfil certain restrictions [48, 49]. The following discussion applies to global symmetries, i.e. invariancy under global operators. However, we have local symmetries in the *ansatz*, invariance under local operators, Eq.(2.7). We consider global symmetries because these are the ones implemented in generic symmetric TN libraries. Chapter 5 shows how

the *ansatz* is not compatible with the description of this section and presents a way of overcoming the incompatibility.

Practically speaking, in symmetric tensors, only components that satisfy a specific constraint are allowed, meaning they can take values other than zero. This constraint is called *sum-rule* and shown in Eq. (3.13). Sum-rules depend on the system's symmetry and are based on the eigenvalues of the matrices representing the (symmetry) group. We introduce such representations on Sec. 3.2.1 and with its result, we give the $\mathbb{Z}_2$ sum-rule in Sec. 3.2.2.

3.2.1 Representation of a Group

A group can be represented with a set of invertible matrices fulfilling the group's multiplication table. In this work, we focus on the $\mathbb{Z}_2$ symmetry, and later, on the $\mathbb{Z}_2 \otimes \mathbb{Z}_2$ symmetry[3]. The cyclic $\mathbb{Z}_2$ group consists of two elements: the identity e and an element x such that $x \cdot x = e$. Namely, $\{x,\ x \cdot x{=}e\}$.

The $1{\times}1$ matrices (1) and (-1) are an (irreducible) representation of $\mathbb{Z}_2$. They readily fulfil the multiplication table, i.e. $(-1){\cdot}(-1) = (1)$. Another representation with $2{\times}2$ matrices is $\{\sigma_x,\ \sigma_x \cdot \sigma_x{=}\mathcal{I}\}$. Consider the Pauli matrix σ_x and its eigenvalues and eigenstates

$$\sigma_x \left|a\right\rangle = (-1)^a \left|a\right\rangle ,\ a \in \{0,1\} \ . \tag{3.9}$$

These labels $a \in \{0,1\}$ are what we call here $\mathbb{Z}_2$ *charges*[4].

Other representations can involve matrices acting on bigger vector spaces. In fact, consider any diagonal matrix with D_0 ones, expressed as $(-1)^0$, and D_1 minus ones, written as $(-1)^1$, on the diagonal:

$$X := \mathrm{diag}\left((-1)^0,\ \overset{D_0}{\ldots},\ (-1)^0, (-1)^1,\ \overset{D_1}{\ldots},\ (-1)^1\right) \ . \tag{3.10}$$

The set $\{X,\ X{\cdot}X{=}\mathcal{I}\}$ is a good $\mathbb{Z}_2$ representation. In this case, the eigenvalues are degenerated,

$$X \left|a;\alpha\right\rangle = (-1)^a \left|a;\alpha\right\rangle ,\ a \in \{0,1\},\ \alpha \in \{1,\ldots\ D_a\} \ , \tag{3.11}$$

where the Greek version of the charge label is used as a degeneracy label. Because the sum-rule uses this charges $a \in \{0,1\}$, we work on this basis and use these labels accordingly, instead of the previous ones in Eq. (3.2),

$$T^{v_1 \ldots v_N}_{u_1 \ldots u_M} \longrightarrow T^{a \ldots b}_{\alpha \ldots \beta} \ . \tag{3.12}$$

We are now in a position to introduce the sum-rule.

[3] By $\mathcal{G}$ symmetry we mean a symmetry (or invariance) under the action of the (representation of the) elements of the group $\mathcal{G}$.

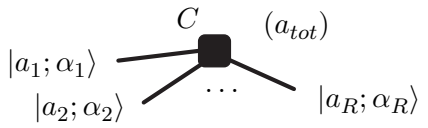
[4] Other conventions consider $\{1,-1\}$ as $\mathbb{Z}_2$ charges instead, leading to a different but equivalent sum-rule.

3.2.2 $\mathbb{Z}_2$ Sum-rule

Consider a tensor C with $R \in \mathbb{N}$ legs and its components —given by $|a_1;\alpha_1\rangle$, $\ldots$, $|a_R;\alpha_R\rangle$—, denoted as $C^{a_1\ldots a_R}_{\alpha_1\ldots\alpha_R}$. The sum-rule reads

$$a_1 + \cdots + a_R = a_{\text{tot}} \pmod 2 \,, \tag{3.13}$$

where $a_{\text{tot}} \in \{0,1\}$ is a $\mathbb{Z}_2$ *total charge* assigned to each tensor. Only components fulfilling Eq. (3.13) can be non-zero. Then, the tensor's components can be written as



$$= \begin{cases} C^{a_1\ldots a_R}_{\alpha_1\ldots\alpha_R} & \text{if } a_1{+}\ldots{+}a_R{=}a_{\text{tot}} \pmod 2 \\ 0 & \text{otherwise} \end{cases} . \tag{3.14}$$

When a distinction is made between vector spaces (legs) of the domain or the codomain, the sum-rule also distinguishes them,

$$a_1 + \cdots + a_D = a_{D+1} + \cdots + a_R + a_{\text{tot}} \pmod 2 \,, \tag{3.15}$$

where $a_1, \ldots a_D$ are the domain components and the rest the codomain components. The component's labels appear on different sides of the equality depending on the vector space's type. Because $a_i{\in}\{0,1\}$, $\forall i{\in}\{1,\ldots\ R\}$ and we add modulo 2, the sum-rules from Eqs. (3.13) and (3.15) are completely equivalent. That is why we do not distinguish between domain and codomain, which leads to not assigning direction to the legs and only using kets in diagrams. In general, for other symmetries, such a distinction is needed.

Because of the sum-rule, tensors that fulfil this constrain —*symmetric* tensors— are sparse tensors. Many of their components are zero. This offers the possibility of storing them more efficiently. Moreover, contractions of symmetric tensors include many terms of the form $0 \cdot C^{a_1\ldots a_R}_{\alpha_1\ldots\alpha_R}$ or $0 \cdot 0$. This translates into a waste of computation time and memory when programming. To avoid this waste of resources, symmetric tensor libraries can be used. These libraries are designed to optimise computations considering all these constraints. Objects that fulfil sum-rules like Eq. (3.13) can be efficiently managed. A critical time advantage in numeric computations is obtained with these libraries. In this work, we used the symmetric library QSpace [50].

As seen in Sec. 5.1, the proposed *ansatz* is remarkably sparse but does not fulfil the $\mathbb{Z}_2$ sum-rule, even though the Hamiltonian is $\mathbb{Z}_2$ symmetric. That is because there is a more restrictive symmetry in the system, the local gauge symmetry. Therefore, we cannot initially use symmetric tensor libraries to efficiently work with the *ansatz*, missing a potential numerical improvement. The first result of this work proposes another symmetry, $\mathbb{Z}_2 \otimes \mathbb{Z}_2$, in which the *ansatz* fulfils the sum-rule, i.e. its tensors are $\mathbb{Z}_2 \otimes \mathbb{Z}_2$ symmetric and

compatible with QSpace. That implies that we are successfully working with the *ansatz* efficiently.

We use the *ansatz* formulated with this newly proposed symmetry to find the ground state of the Hamiltonian of the system given in 2. We use the imaginary time evolution power method in this work [18, 21], reviewed in the next section.

3.3 Tensor Networks Algorithm

In general, the Hilbert space where the state of a system lives grows exponentially with the system's size. Even in this work, where the physical Hilbert space is a corner of the total Hilbert space (Eq. (2.11)), it still has an exponential size. Therefore, exact diagonalisation is usually ruled out as an option to find the ground state (gs). Other ways must be used for that purpose. Here, we review the imaginary time evolution power method and see how it can find the ground state.

Consider any Hamiltonian $\mathcal{H}$ and any initial state of the system $|\Psi\rangle$. Solving Schrödinger's equation, the evolution of this state by $\mathcal{H}$ to a time t is given by the exponential operator,

$$|\Psi(t)\rangle = e^{-it\mathcal{H}} |\Psi\rangle \ . \tag{3.16}$$

If we consider imaginary time $t := -i\tau$, with $\tau \in \mathbb{R}$, the evolution operator is

$$e^{-\tau\mathcal{H}} = e^{-\tau \cdot E_{\text{gs}}} |\Phi_{\text{gs}}\rangle \langle\Phi_{\text{gs}}| + e^{-\tau \cdot E_1} |\Phi_1\rangle \langle\Phi_1| + \dots \tag{3.17}$$

where we have also included its eigenvalue decomposition, with eigenvalues (energies) $E_{\text{gs}} \leq E_1 \leq E_2 \leq \dots$ and $|\Phi_{\text{gs}}\rangle, |\Phi_1\rangle, |\Phi_{\text{gs}}\rangle \dots$ the corresponding eigenvectors. The state $|\Phi_{\text{gs}}\rangle$ is the ground state with energy E_{gs}. For a positive time τ and $E \neq E_{\text{gs}}$,

$$\frac{e^{-\tau \cdot E_{\text{gs}}}}{e^{-\tau \cdot E}} = e^{-\tau(\overbrace{E_{\text{gs}} - E}^{\leq 0})} = e^{\tau|\Delta E|} \geq 1 \quad \Longleftrightarrow \quad e^{-\tau \cdot E_{\text{gs}}} \geq e^{-\tau \cdot E} \ . \tag{3.18}$$

Moreover, the longer the evolution (larger τ), the more dominant the first term becomes. In the limit $\tau \to \infty$, the evolution operator is proportional to the projector into the ground state,

$$\lim_{\tau \to \infty} e^{-\tau\mathcal{H}} = e^{-\tau \cdot E_{\text{gs}}} |\Phi\rangle_{\text{gs}} \langle\Phi|_{\text{gs}} \ . \tag{3.19}$$

The evolution removes higher energy components as time passes. In other words, an imaginary evolution evolves any state towards the ground state.

Accordingly, the energy of the *normalised*[5] evolved state decreases, reaching E_{gs} in the limit. Therefore, finding a way of implementing this evolution grants access to the ground state.

Such implementation is done with TN; both the state and the operator are represented with a TN. The state's representation is explained in Ch. 4 and in Sec. 3.3.1 the evolution operator is decomposed into a TN. The evolution is implemented in time steps. On each time step, the energy goes down while the dimensions of the TN's tensors grow. Therefore, some *truncation scheme* must be used during each time step. Truncating brings down the dimensions of the tensors by projecting each vector space. The truncation scheme is explained in Sec. 3.3.2 and technical details are given in Appendix A.

3.3.1 Evolution with Tensor Networks

The evolution during a time τ is broken down into τ/β time steps, for some time step β. In this position, the exponential operator can be decomposed with a Suzuki-Trotter decomposition [52] with an error $O(\beta^3)$, which can be manageable if β is chosen small.

We work with the Hamiltonian from Eq. (2.3). The Suzuki-Trotter decomposition is done by splitting the Hamiltonian as follows

$$\mathcal{H} = \mathcal{H}_Z + \mathcal{H}_X \qquad \text{with} \begin{cases} \mathcal{H}_Z = -J \sum_p B_p \\ \mathcal{H}_X = h \sum_l \sigma_l \end{cases} , \tag{3.20}$$

where p runs over all plaquettes of the lattice ($\mathcal{P}$) and l over all the links ($\mathcal{L}$). Notice that the terms of $\mathcal{H}_Z$ commute with each other. Similarly for $\mathcal{H}_X$. Then, the second-order expansion reads

$$e^{-\beta\mathcal{H}} = e^{-\beta/2\ \mathcal{H}_X} \cdot e^{-\beta\mathcal{H}_Z} \cdot e^{-\beta/2\ \mathcal{H}_X} + \mathcal{O}(\beta^3).$$

The evolution of a single time step is broken down into these three global gates. Due to the commutation relations of $\mathcal{H}_Z$ and $\mathcal{H}_X$, we can further decompose the global gates into local and plaquette gates,

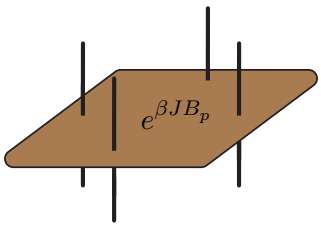
$$e^{-\beta/2\ \mathcal{H}_X} = \prod_{l\in\mathcal{L}} e^{-h\frac{\beta}{2}\ \sigma_x^l} , \qquad e^{-\beta\mathcal{H}_Z} = \prod_{p\in\mathcal{P}} e^{\beta J B_p} . \tag{3.21}$$

Furthermore, the plaquette gate can be expressed as the contraction of smaller tensors. Using the fact that $B_p^n = \mathcal{I}$ if n is even and $B_p^n = B_p$ if n is odd, we write

$$e^{\beta J B_p} = \cosh(\beta J) \cdot \mathcal{I} + \sinh(\beta J) \cdot B_p . \tag{3.22}$$

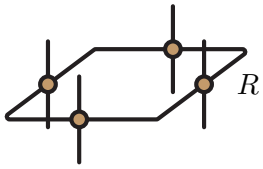
[5] The evolution is not unitary: it changes the state's norm, which must be corrected.

Using this we can express the plaquette gate as a so-called cyclic Matrix Product Operator,

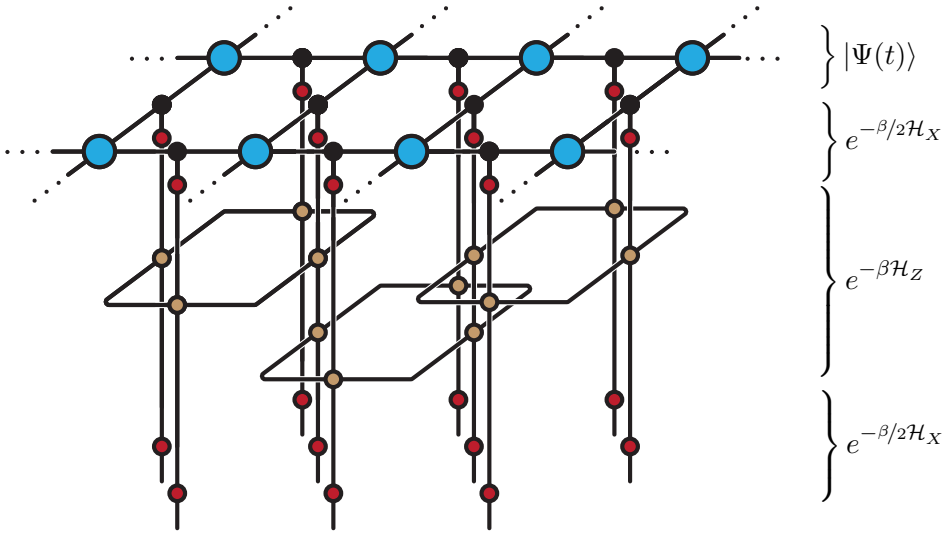


$$= \mathrm{Tr}\left[\begin{pmatrix} \cosh(\beta J)\cdot\mathcal{I} & 0 \\ 0 & \sinh(\beta J)\cdot B_p \end{pmatrix}\right] =$$

$$= \mathrm{Tr}\left[\overbrace{\begin{pmatrix} \cosh(\beta J)^{1/4}\cdot\mathcal{I} & 0 \\ 0 & \sinh(\beta J)^{1/4}\cdot\sigma_z^l \end{pmatrix}}^{R}{}^{4}\right] =$$

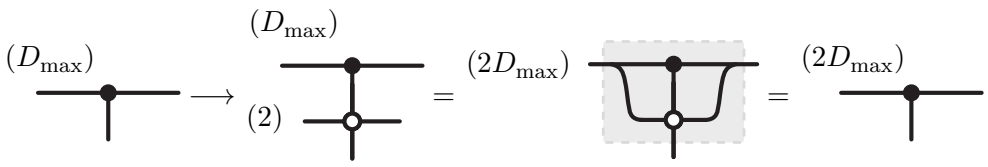
$$=$$  $$R$$ (3.23)

Bringing everything together, one time step can be expressed as a TN as in Eq. (3.24).

$|\Psi(t+\beta)\rangle =$ 

(3.24)

After contracting the gates into the *ansatz*, its original structure can be recovered back, as detailed in Ch. 6 and depicted in Eq. (3.25), where it is shown for one link. The structure is recovered but some dimensions are doubled.

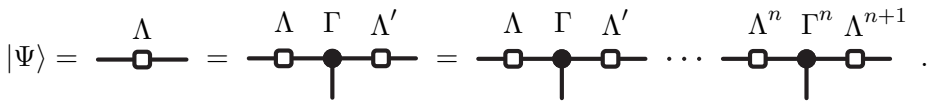


(3.25)

Equation Eq.(3.25) shows that the dimensions grow exponentially with the number of time steps, quickly reaching a size limitation. Therefore, we must project the vector space to a smaller subspace, also known as *truncating*. The method we use is explained in Sec. 3.3.2.

3.3.2 Truncation

Truncating a TN's tensor changes the represented state from $|\Psi\rangle$ to $|\tilde{\Psi}\rangle$. Ideally, the distance between these two states —measured by some norm like Frobenius norm— should be minimal. There exists a widely used optimised truncation method for (1D) matrix product states [53]. It is based on the Schmidt decomposition of the state around each bond. Consider a TN where each bond includes a diagonal matrix Λ with the singular values from the Schmidt decomposition around that bond,

$|\Psi\rangle =$  (3.26)

This form is usually referred to as *canonical form*. The truncation follows by projecting each virtual leg's vector space to the subspace where the singular values of Λ are higher.

In systems with loops, such as square lattices, a similar Schmidt decomposition is not possible anymore. The system does not separate into two parts around a bond. Despite that, a form similar to Eq. (3.26) is used for the *ansatz* of this work, as shown in Eq. (3.27).

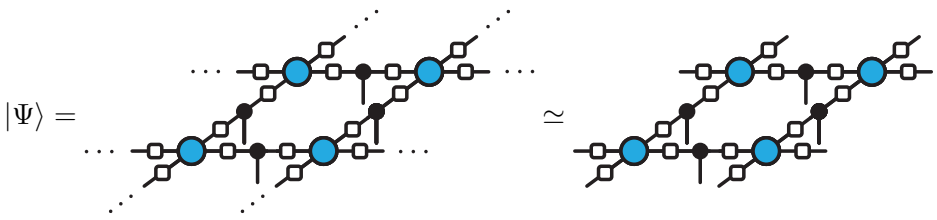
Notice that one can represent the whole state with a tensor Γ on a single physical site and two matrices around it, Λ and Λ'. In other words, the rest of the system — the *environment* of Γ— is captured by the Λ's.

Inspired by this, a similar thing is done in practice when working with the $(2D)$ *ansatz* presented[6] in Ch. 4. Each bond includes some weights as a diagonal matrix, reminiscent of the Λ's. See Eq. (3.27). Moreover, they are treated as environments for sub-networks of the TN, similar to Eq. (3.26). However, in this case, they are only an approximation to the environment[7]. This is

[6] In that chapter, the Λ matrices are ignored, but in practice, they are always included.

[7] In Eq. (3.26), the Λ matrices represented the environment exactly. In 2D, the environment can not be generally written using only these weights.

illustrated with the "$\simeq$" of Eq. (3.27).



(3.27)

For example, this approximation is used when computing a state's energy. A sub-network such as the one in Eq. (3.27) is extracted and used to calculate $\langle\Psi|\mathcal{H}|\Psi\rangle$. This environment approximation is one source of error in the results. However, we argue in Ch. 7 that the truncation error dominates.

Finally, the truncation we use is based on the weights (singular values) on the Λ matrices, similar to the 1D case. The vector space is projected on each bond to the subspace with bigger singular values. Technical details are given in Appendix A.

In the context of the imaginary evolution, applying the gates lowers the energy, while the truncation increases the energy. This increase typically occurs because truncation can populate high-energy components. In Ch. 7, the balance between these two energy changes is commented on, and the quality of the truncation method is discussed.

Chapter 4

Ansatz

We want a variational TN *ansatz* that allows us to explore the physical Hilbert space, Eq. (2.11). This is, we would like an *ansatz* that only represents states from a specific Gauss sector, i.e. states that fulfil Eq. (2.12) (Gauss' law) on all vertices, here referred to as *gauge states*.

Such an *ansatz* was introduced by L. Tagliacozzo, A. Celi and M. Lewenstein in 2014 [1]. It was formulated for arbitrary groups, but we will present only the $\mathbb{Z}_2$ version in this chapter. Remarkably, as explained in Ch. 5, we eventually need to use a $\mathbb{Z}_2 \otimes \mathbb{Z}_2$ symmetry. It has a similar structure as the $\mathbb{Z}_2$ case, and the differences are shown in detail in that chapter.

The *ansatz* we are about to present is based on two types of tensors: *copy tensors* on the links, Eq. (4.2); and *vertex tensors* on the vertices, Eq. (4.3), as shown in Fig. 4.1. The underlying idea is to move the physical information from the physical sites to the vertices. Then, the vertex tensors remove the non-gauge components of the state with that information. We show in Sec. 4.2 how every state represented by the *ansatz* is a gauge state. Additionally, in Sec. 4.3, we will discuss the inverse statement, which will be dealt with in more detail in Ch. 6.

4.1 *Ansatz*

The underlying idea is the following. We want a TN that automatically *checks* if a component fulfils Gauss' law. If it does not, the TN would evaluate to zero, removing that component from the state. Therefore, in the resulting state, only Gauss' law fulfilling component would be allowed to be different than zero. That would result in TN representing a gauge state, making the *ansatz* useful for studying that Hilbert subspace.

To check Gauss' law on each site, you need the information from the surrounding links. Then, the basic idea of the *ansatz* is to move the information from

the links to the two surrounding vertices. Then, on each vertex — which would now have enough information — evaluate to zero if Gauss' law is not fulfilled, sending the whole TN contraction (and thus the represented state's component) to zero.

The two basic blocks we need to build this *ansatz* are the *copy* and *vertex tensors*. See Fig. 4.1. There is one copy tensor on each link of the lattice, and they are in charge of moving the information towards the vertices. There is a vertex tensor on each vertex, which checks for Gauss' law.

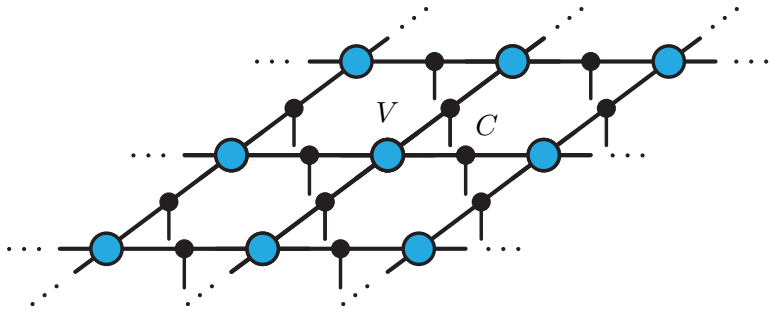
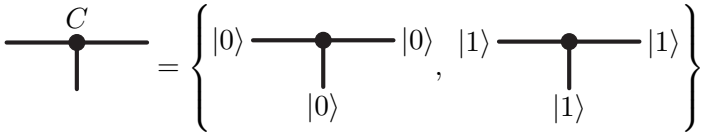


Figure 4.1: Distribution of the two types of basic tensors on the *ansatz* for an infinite square lattice. Vertex tensors, V, are placed on each vertex, represented with big blue circles. Copy tensors, C, are placed at each link, represented with small black circles.

4.1.1 Copy Tensor

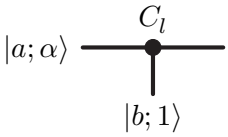
A copy tensor, C, is defined as a 3-legged tensor with only two components, shown in the set of Eq. (4.1).



(4.1)

The rest of the components are zero. Effectively, C *copies* the charge from the physical site into the virtual vector space. In other words, it *moves* the information from the physical leg to the virtual legs.

More explicitly, a copy tensor on a link l, C_l, has the following components:

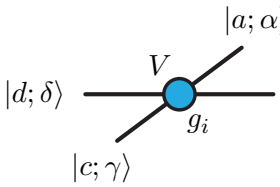


$$= \begin{cases} [C_l]^{abc}_{\alpha\gamma} & \text{if } a = b = c \\ 0 & \text{otherwise} \end{cases} , \tag{4.2}$$

where $a, b, c \in \{0, 1\}$ are $\mathbb{Z}_2$ charges; $\alpha \in \{1, \dots A_a\}$ and $\gamma \in \{1, \dots \Gamma_c\}$ are the degeneracy labels whose dimensions $A_a, \Gamma_c \in \mathbb{N}$ depend on the charge. This means that the left leg has dimension $A = A_0 + A_1$, the right leg dimension $\Gamma = \Gamma_0 + \Gamma_1$ and the physical leg dimension 2

4.1.2 Vertex Tensor

The vertex tensor, V, on a vertex i, has one leg connected to each surrounding copy tensor[1]. Its components are



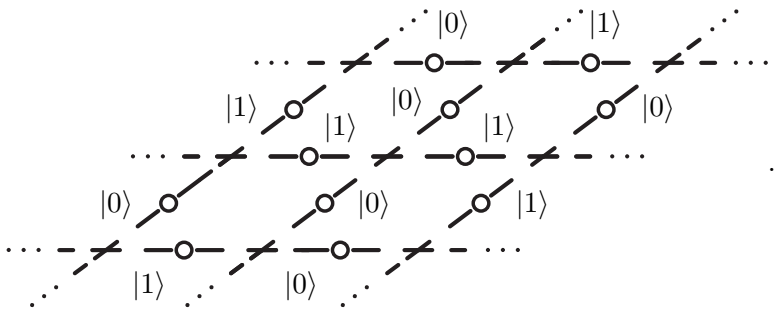
$$= \begin{cases} [V]^{abcd}_{\alpha\beta\gamma\delta} & \text{if } a+b+c+d{=}g_i \pmod 2 \\ 0 & \text{otherwise} \end{cases} , \tag{4.3}$$

where $g_i \in \{0,1\}$ is the $\mathbb{Z}_2$ background charge at site i. Labels $a, b, c, d \in \{0,1\}$ are $\mathbb{Z}_2$ charges. Consider some dimensions which depend on the charges $A_a, B_b, \Gamma_c, \Delta_d \in \mathbb{N}$. Labels α, β, γ and δ are degeneracy labels which range from 1 to A_a, B_b, Γ_c and Δ_d, respectively. Therefore, in general, vertex tensors have dimensions $A \times B \times \Gamma \times \Delta$, with $A = \sum_a A_a$, $B = \sum_b B_b$, ...

This vertex tensors evaluate to zero when Gauss' law in that vertex —$a + b + c + d = g_i \pmod 2$— is not fulfilled. Combining copy and vertex tensors, Gauss' law can be successfully imposed into the physical sites. Let's see an example.

4.2 Imposing Gauss' Law

Consider any general TN with this *ansatz* structure. Let the state represented by this TN be $|\phi\rangle$. Consider as well the simple (product) state as an example

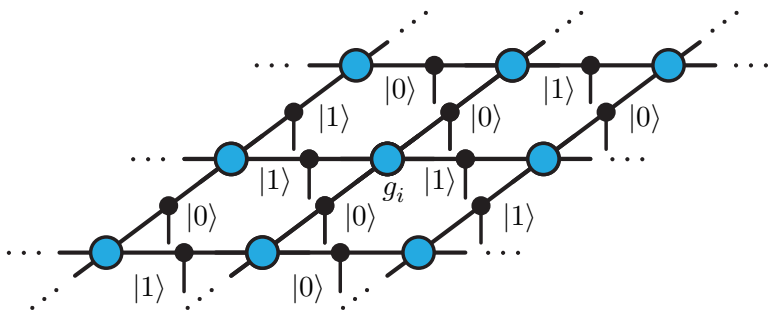
$|\eta\rangle =$ 

. (4.4)

We are interested in knowing the $|\eta\rangle$-component of the state represented by the TN, $|\phi\rangle$. This is, we ask for $\langle\eta|\phi\rangle$. In order to get it, we have to compute

[1] In a square lattice, like Fig. 4.1, the vertex tensors have 4 legs.

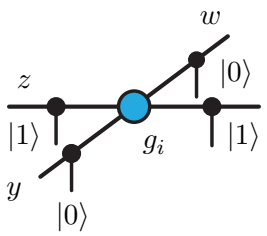
the result of the contraction of the following TN,



. (4.5)

Consider the vertex i, which has a certain background charge g_i. The contraction will look like

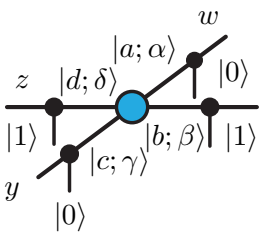
$$\langle\eta|\phi\rangle = \sum_{w,x,y,z}$$



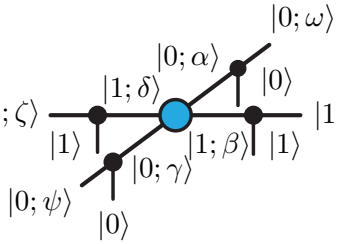
$$\cdot\ E_{wxyz}, \tag{4.6}$$

where E_{wxyz} is the environment of the vertex i and its surrounding copy tensors. Labels w, x, y, z are general labels of the virtual bond.

If we now compute the reduced contraction inside the summation, we get



x



$|1;\zeta\rangle$ $|1;\chi\rangle$ =

$$= \sum_{\alpha\beta\gamma\delta} [C_U]^{000}_{\alpha\omega}\,[C_R]^{111}_{\beta\chi}\,[C_D]^{000}_{\gamma\psi}\,[C_L]^{111}_{\delta\zeta}\cdot[V_i]^{0101}_{\alpha\beta\gamma\delta} =$$

$$= \begin{cases} \sum_{\alpha\beta\gamma\delta}\cdots & \text{if } 0{+}1{+}0{+}1{=}g_i \pmod 2 \\ 0 & \text{otherwise} \end{cases} . \tag{4.7}$$

On the first equality, the copy tensors *copied* the physical charge to the virtual bonds[2], fixing the vertex's component to the values of the physical charge. Therefore, *moving* the information successfully. The second equal is the explicit expansion of the contraction, with C_U, C_R, C_D and C_L the up, right, down

[2]That imposes a restriction on the labels w, x, y, z, but we focus on the vertex's effect here.

and left copy tensors of the vertex tensor V_i, respectively. The copy tensor factor in the summation is already in general non-zero, but V_i still imposes a restriction on the physical charges, yielding the last conditional result.

Due to the vertex tensor, the reduced contraction evaluates to zero if the *physical* charges do not fulfil Gauss' law. Finally, this result implies that, in the complete contraction of $\langle\eta|\phi\rangle$ on Eq. (4.6), the result also depends on the physical charges around vertex i:

$$\langle\eta|\phi\rangle = \sum_{w,x,y,z} \; \cdot E_{wxyz} = \begin{cases} \sum \; \cdot E_{wxyz} & \text{if } 0{+}1{+}0{+}1{=}g_i \pmod 2 \\ 0 & \text{otherwise} \end{cases} . \tag{4.8}$$

In other words, $\langle\eta|\phi\rangle = 0$ if vertex i does not fulfill Gauss' law. This can be done for every vertex. Therefore, we can write

$$\langle\eta|\phi\rangle = 0 \quad \text{if Gauss' law is not fulfilled everywhere in } |\eta\rangle \; . \tag{4.9}$$

Since we can do that for general product states $|\eta\rangle$, the *ansatz* can be made to not have any gauge-breaking component.

Summing up, for a given set of background charges $\{g_i\}$ $\forall i \in \mathcal{V}$, this *ansatz* represents states where all components fulfil Gauss' law. Meaning, *all* represented states are gauge states belonging to the Gauss sector K corresponding to the set of background charges $\{g_i\}$.

4.3 Conclusion

With copy and vertex tensors of Eqs. (4.2) and (4.3) we can build the *ansatz* of Fig. 4.1, and we have shown that it only represents gauge states, Eq. (4.8). We can write as a conclusion:

$$\text{this } \textit{ansatz} \text{ structure} \quad \Longrightarrow \quad \text{represented state is a gauge state.} \tag{4.10}$$

The inverse statement,

$$\text{gauge state} \quad \Longrightarrow \quad \text{can be represented with this } \textit{ansatz} \text{ structure,} \tag{4.11}$$

would mean that the *ansatz* can reach *all* gauge states of a Gauss sector for sufficiently large bond dimension. That would be a remarkable feature. Showing this inverse statement is not obvious, and we present it as part of this thesis results in Ch. 6.

Moreover, notice how sparse the tensors of the *ansatz* are. Specially the copy tensor of Eq. (4.2). Usually, these sparse tensors are symmetric tensors,

as defined in Ch. 3, and symmetric TN libraries can be used to optimally work with these objects in numerical methods. Because of the local gauge symmetry, the structure of this *ansatz* tensors is not straightforward. One would be tempted to treat them as $\mathbb{Z}_2$-symmetric objects, but they are not, as we will see in detail in the next chapter, Ch. 5. In that chapter, we present the main result of this work, the reformulation of this *ansatz* in terms of a $\mathbb{Z}_2 \otimes \mathbb{Z}_2$ symmetry.

Chapter 5

$\mathbb{Z}_2 \otimes \mathbb{Z}_2$ Proposal

5.1 Non $\mathbb{Z}_2$ Symmetric *Ansatz*

As introduced in Ch. 4, the copy tensors are at the core of the *ansatz*. They carry information towards both sides, from the physical site to the vertices. We give again the $\mathbb{Z}_2$ charge sectors of a copy tensor, shown in Fig. 5.1. All other sectors not shown in the figure are zero. That means C has a very sparse structure.

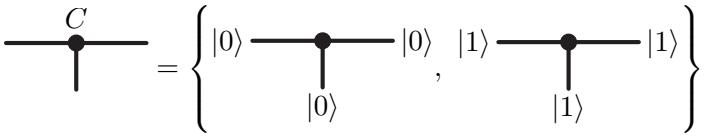
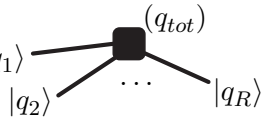


Figure 5.1: $\mathbb{Z}_2$ charge sectors of a copy tensor C of the *ansatz*. The equality between tensor C and the set means that the only possible non-zero $\mathbb{Z}_2$ charge sectors are those in the set. Since no degeneracy labels exist in the set's diagrams, they refer to the whole charge sectors (3-dimensional arrays), not only to a single element.

Accordingly, our goal is to implement these tensors on a symmetric tensor library[1]. There, symmetric tensors have to fulfil certain restrictions. Explicitly, consider the $\mathbb{Z}_2$ symmetry case and tensors with $q_{\text{tot}} \in \{0, 1\}$ and $R \in \mathbb{N}$ legs. Then, its sectors, given by $q_1, \ldots, q_R \in \{0, 1\}$, which do not fulfill the sum-rule of Eq. (5.1) must be zero.

$$q_1 + q_2 + \cdots + q_R \neq q_{\text{tot}} \pmod 2 \implies$$  $$= 0 \qquad (5.1)$$

[1] Such libraries are reviewed in Ch. 3, where we introduced some basics of TN.

Taking a look at the $\mathbb{Z}_2$ charge sectors from the copy tensor C in Fig. 5.1, we see that one of the sectors must be zero for C to be $\mathbb{Z}_2$ symmetric. The first sector fulfills the sum-rule for $q_{\text{tot}} = 0$ while the second fulfills it for $q_{\text{tot}} = 1$. However, a $\mathbb{Z}_2$ tensor can only have one q_{tot}. Therefore, both sectors can not fulfil the restriction simultaneously. This incompatibility of C's sectors and the restrictions of a $\mathbb{Z}_2$ symmetric library poses a problem: we can not use these $\mathbb{Z}_2$ libraries to work optimally with our *ansatz*.

5.2 Proposed Solution

To address this problem, we adopt the following approach: we aim to transfer information from the physical vector space (leg) to both sides of the copy tensor. We do this by *doubling* the information on the physical leg and sending one part to each side.

Our initial working idea was doubling the physical sites. Then, each site would only send information to its closest vertex. We would ensure that they send the same information by projecting the doubled sites into singlets. This idea was developed using $U(1)$ charges and, later, using $\mathbb{Z}_2$ charges. This strategy suffered from some other issues and was later modified to the presented solution in the next section, which uses another symmetry for the same *ansatz*.

5.2.1 New Symmetry

Following the strategy of *doubling* information, we *double* the symmetry: from $\mathbb{Z}_2$ to $\mathbb{Z}_2 \otimes \mathbb{Z}_2$, and then see how the *ansatz* can be adapted.

First, let's see vector spaces when such a symmetry is present. From a practical point of view, this implies that now each state has two charges. Let them be p and q, each individually a $\mathbb{Z}_2$ charge. Accordingly, they can take values $p, q \in \{0, 1\}$. One such vector $|\eta\rangle$ could be

$$|\eta\rangle = |p{=}0, q{=}1\rangle \equiv |01\rangle \equiv |01; d\rangle \,. \tag{5.2}$$

In the expression, $d \in \{1, \ldots, D_{pq}\}$ is the degeneracy label and D_{pq} the dimension of the degenerated sector, which depends on the value of the p and q charges. The label d distinguishes vectors with the same charges, i.e. in the same charge sector. Following from Eq. (5.2), we double the information of the original states, $|0\rangle$ and $|1\rangle$, and work now with the states

$$|0\rangle \longrightarrow |p{=}0, q{=}0\rangle \equiv |00\rangle \equiv |00; 1\rangle \,, \tag{5.3}$$

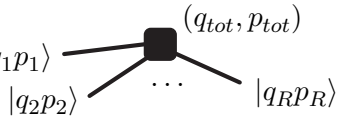
$$|1\rangle \longrightarrow |p{=}1, q{=}1\rangle \equiv |11\rangle \equiv |11; 1\rangle \,. \tag{5.4}$$

Note that there is no degeneration on this vector space, i.e. $d \in \{1\}$. On the physical vector spaces (physical legs), the vectors on the right-hand side of

Eqs. (5.3) and (5.4) will convey the same meaning as those on the left-hand side. The remaining states $|p{=}1, q{=}0\rangle$ and $|p{=}0, q{=}1\rangle$ do not have physical meaning. We do not have to worry about these components appearing on the results because they will not appear in our tensors, as seen in Ch. 6.

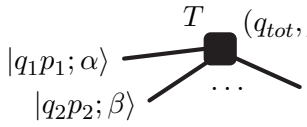
Now that we have defined the vector space, let's see the restrictions that must fulfil $\mathbb{Z}_2 \otimes \mathbb{Z}_2$ tensors. A tensor's component is given by a set of pairs of charges, as illustrated in the diagram of Eq. (5.5). $\mathbb{Z}_2 \otimes \mathbb{Z}_2$ symmetric tensors with $R \in \mathbb{N}$ legs have now two total $\mathbb{Z}_2$ charges, $p_{\text{tot}}, q_{\text{tot}} \in \{0, 1\}$. The restriction they fulfil is the following. Consider the sectors given by the set of $\mathbb{Z}_2$ charges pairs, $p_i, q_i \in \{0, 1\}$, $i \in \{1, \dots R\}$. Those which do not fulfill both sum-rules from the left-hand side of Eq. (5.5) — one for each charge — simultaneously must be zero.

$$\left.\begin{array}{c} p_1 + p_2 + \cdots + p_R \neq p_{\text{tot}} \pmod 2 \\ \text{and} \\ q_1 + q_2 + \cdots + q_R \neq q_{\text{tot}} \pmod 2 \end{array}\right\} \implies$$



$$= 0 \tag{5.5}$$

More compactly, the elements of a $\mathbb{Z}_2 \otimes \mathbb{Z}_2$ symmetric tensor T with $R \in \mathbb{N}$ legs and total charges $p_{\text{tot}}, q_{\text{tot}} \in \{0, 1\}$ are



$$= \begin{cases} [T]^{p_1 q_1 \cdots p_R q_R}_{\alpha \dots \rho} & \text{if } \begin{cases} p_1 + \cdots + p_R = p_{\text{tot}} \pmod 2 \\ \text{and} \\ q_1 + \cdots + q_R = q_{\text{tot}} \pmod 2 \end{cases} \\ 0 & \text{otherwise} \end{cases} \tag{5.6}$$

where $p_i, q_i \in \{0, 1\}$, $i \in \{1, \dots R\}$ are the pair of $\mathbb{Z}_2$ charges of each component, and the degeneracy labels $\alpha, \beta, \dots, \rho$ range from 1 to a maximum degeneracy dimension, D_{pq}, which depends on the corresponding charges of the component. See Eq. (5.2). With this symmetry, we are able to design the tensors from the *ansatz* as $\mathbb{Z}_2 \otimes \mathbb{Z}_2$ symmetric tensors.

5.2.2 Copy Tensors

Having two charges on each state, allows us to send one part of the doubled information, i.e. one charge, to one side and the other part, i.e. the other charge, to the other side. In this work, we consider two possibilities to do that, which give rise to two different copy tensors — two *types* —, which we name pq- and qp-type.

pq-type

The first physical $\mathbb{Z}_2$ charge, p, is sent *only* to the left, while the second one, q, is sent *only* to the right side. Accordingly, we define a pq-type copy tensor

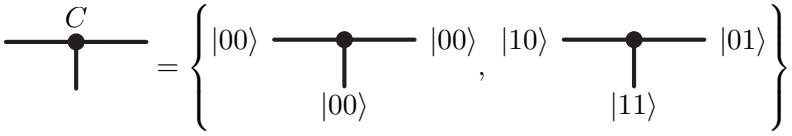


Figure 5.2: $\mathbb{Z}_2 \otimes \mathbb{Z}_2$ charge sectors of a pq-type copy tensor C. Notation is similar to that in Fig. 5.1.

C as shown in Fig. 5.2. Indeed, this copy tensor is $\mathbb{Z}_2 \otimes \mathbb{Z}_2$ symmetric since the total charges of each sector match. The first component has total charges $p_{\text{tot}} = 0 + 0 + 0 = 0$, $q_{\text{tot}} = 0 + 0 + 0 = 0$ and the second one also has $p_{\text{tot}} = 1 + 1 + 0 = 0 \pmod 2$, $q_{\text{tot}} = 0 + 1 + 1 = 0 \pmod 2$.

qp-type

In this case, the first physical $\mathbb{Z}_2$ charge p is sent only to the right, and the second one q is sent only to the left. Such a qp-type copy tensor is defined as shown in Fig. 5.3. Similar to the pq-type, one can also check that this copy tensor has total charges $p_{\text{tot}} = q_{\text{tot}} = 0$.

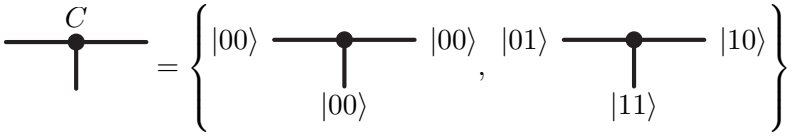
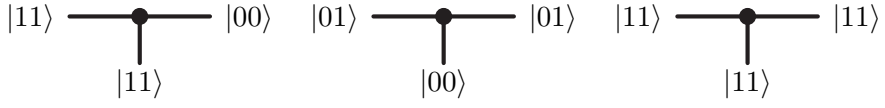


Figure 5.3: $\mathbb{Z}_2 \otimes \mathbb{Z}_2$ charge sectors of a qp-type copy tensor C. Notation is similar to that in Fig. 5.1

Comments

Sectors. Other sectors are allowed by the symmetry for a tensor with $p_{\text{tot}} = q_{\text{tot}} = 0$, like the examples shown in Fig. 5.4. However, the resulting tensor



(a) $p_{\text{tot}} = q_{\text{tot}} = 0$. (b) $p_{\text{tot}} = q_{\text{tot}} = 0$ (c) $p_{\text{tot}} = q_{\text{tot}} = 1$

Figure 5.4: Other allowed sectors in a $\mathbb{Z}_2 \otimes \mathbb{Z}_2$ symmetric tensor.

could not be a valid tensor as defined in this work. However, since the symmetry (and library) allows this sector, it can appear in the calculations. This is a problem in this work because operations that we use preserve the *ansatz*

structure, as will be shown in Ch. 6, and possible noise errors are controlled through truncation.

Type of Links. Moreover, note that in a link, *only one* charge carries information. The other charge always equals zero, meaning that components are zero where that charge is 1, as depicted in Fig. 5.5.

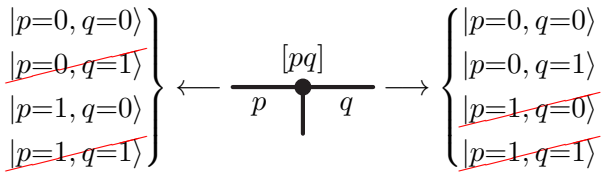


Figure 5.5: States on the virtual bonds for which the component are zero (stroked out) or can be non-zero (not stroked out) in a pq-type copy tensor.

Therefore, we can define the notation: p-type bond and q-type bond for bonds on which the information is carried through the p and q charge, respectively. In Fig. 5.5, the left bond is p-type and the right one is q-type, as the label indicates.

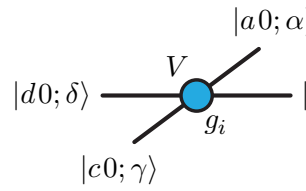
5.2.3 Vertex Tensors

We define the vertex tensors as $\mathbb{Z}_2 \otimes \mathbb{Z}_2$ symmetric tensors as well. The idea is to implement Gauss' law — which involves all four surrounding charges —– using the sum-rules of the symmetric restriction. Note that these sum-rules can only compare each $\mathbb{Z}_2$ charge *individually*: each sum-rule involves either p or q components. Therefore, we *need* the information to arrive at the vertex through the *same* charge. This is, all four bonds of a vertex *must* be of the same type: all p- or all q-type. This is a substantial restriction whose effects will be reviewed in the next section, Sec. 5.3. With this distinction, we also have two types of vertex tensors.

p-type

The p-type vertex tensor V at vertex i with background charge $g_i \in \{0, 1\}$ is defined as a 4-legged $\mathbb{Z}_2 \otimes \mathbb{Z}_2$ symmetric tensor with total charges $p_{\text{tot}} = g_i$ and $q_{\text{tot}} = 0$. The comparison happens in the $\mathbb{Z}_2$ p charge. Accordingly, its

elements are

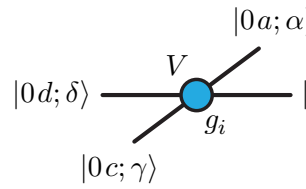


$$= \begin{cases} [V]^{abcd}_{\alpha\beta\gamma\delta} & \text{if } \; a+b+c+d = g_i \pmod 2 \\ 0 & \text{otherwise} \end{cases}, \tag{5.7}$$

where $a, b, c, d \in \{0, 1\}$ and the corresponding degeneracy labels range from 1 to a $A_a, B_b, \Gamma_c, \Delta_d$, respectively.

q-type

This type is analogous as the p-type, but with total charges $p_{\text{tot}} = 0$ and $q_{\text{tot}} = g_i$ — comparison happens in q —, which leads to the following elements:



$$= \begin{cases} [V]^{abcd}_{\alpha\beta\gamma\delta} & \text{if } \; a+b+c+d = g_i \pmod 2 \\ 0 & \text{otherwise} \end{cases}, \tag{5.8}$$

where labels are similar to Eq. (5.7).

5.2.4 Revisit *Ansatz*

Here, we illustrate how these newly defined tensors from our *ansatz* can implement Gauss' law. The example is shown around a p-type vertex tensor in Fig. 5.6.

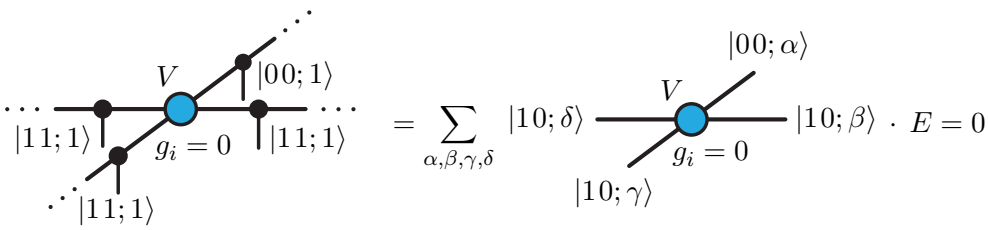


Figure 5.6: Illustration of a vertex tensor, as defined in Eq. (5.7), sending the whole contraction result to zero when the physical component does not fulfil Gauss' law. Indeed, $0 + 1 + 1 + 1 \neq 0 \pmod 2$. The contraction is explicitly expanded around the vertex, with E representing the vertex's environment.

5.3 Consequences

The need to have all four links of a vertex tensor of the same type imposes a restriction on the distribution of types of tensors in the lattice. In our case, the square lattice, this restriction enforces a checkerboard-like pattern, as depicted in Fig. 5.7.

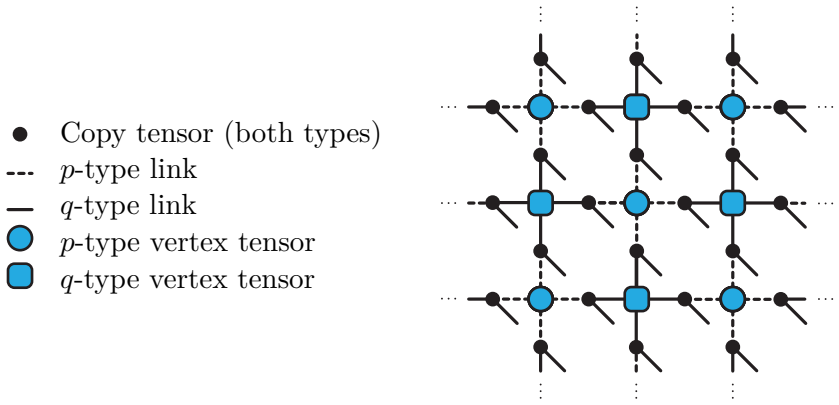


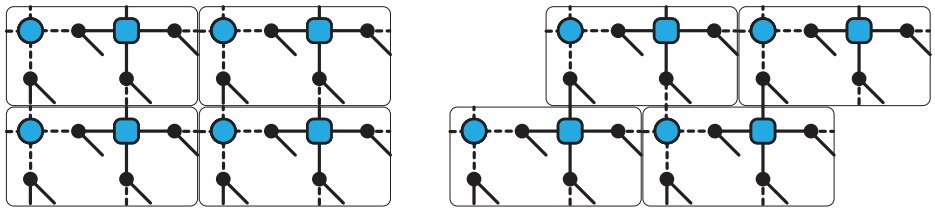
Figure 5.7: Distribution of the different types of tensors in a square infinite lattice. Notation for the different types is given in the figure. Notice the checkerboard-like pattern that the vertex tensors form and that they all have a unique type of bond.

Furthermore, the restriction extends to the size of the unit cell and how to tile the lattice with it. Let's denote the size of the unit cell $L_x \times L_y$, where L_x is the number of vertex tensors along the horizontal direction. Similar for L_y in the vertical direction. For each vertex tensor in a unit cell, the right and bottom copy tensors are included. As an example, a 1×1 and 1×2 unit cells are shown in Fig. 5.8.

Figure 5.8: Examples of unit cells. Notation follow from Fig. 5.7. **(a)** 1×1 unit cell. This forms the fundamental block for larger unit cells. **(b)** 1×2 unit cell. Used in Fig. 5.9.

Consider the 1×1 example. If we were to tile the whole lattice with it, no checkerboard pattern would appear. That would imply that the vertex's bonds are not the same type; thus, the TN can not check Gauss' law.

In fact, we would run into a similar problem for unit cells which *both sides are odd*: there is no tiling that gives rise to a checkerboard pattern. Therefore, such unit cells with both sides odd are not allowed. On the other hand, unit cells with *both sides even* can be used normally to tile the lattice, and a checkerboard pattern is recovered. The *mixed case*, where one side is odd and the other even, still allows for the checkerboard pattern, but the tiling must be shifted. In Fig. 5.9 an example is given for a unit cell with size 1×2.



(a) (b)

Figure 5.9: **(a)** Shifted and **(b)** not shifted tiling for a 1×2 unit cell — the unit cell is depicted in Fig. 5.8b. Notation follows from Fig.5.7. Note that when there is no shifting the checkerboard pattern is lost and different types of links are contracted.

In the case where the lattice is non-bipartite, this setup would not work. In a triangular lattice, for instance, there is no distribution of the current types of tensors that allows to have vertex tensors with unique types of bonds. A first solution to this issue would be adding a third $\mathbb{Z}_2$ charge. In this scenario, there would be three different kinds of links (as well as six kinds of copy tensors). This diversity would allow for the correct tiling of a triangular lattice. Acting similarly for other lattices may suffice to ensure that vertex tensors have unique types of bonds.

5.4 Practical Example and Conclusion

For practical convenience and as an example, we give the elements of the Pauli operators σ_x, σ_y and σ_z in this new symmetry. For the off-diagonal operators, we include an auxiliary leg to make them symmetric. Since there are two charges, again we consider two different auxiliary legs. Accordingly, they distinguish two types of operators. Operators σ_x, σ_y and σ_z are given in Eqs. (5.9) to (5.13).

$$\sigma_x = \left\{ \begin{array}{l} |00;1\rangle \\ \quad\vdots \\ |00;1\rangle \end{array} = 1, \quad \begin{array}{l} |11;1\rangle \\ \quad\vdots \\ |11;1\rangle \end{array} = 1 \right\} \tag{5.9}$$

$$\underline{pq}\text{-type} \qquad \sigma_y = \left\{ \begin{array}{c} |00;1\rangle \\ |10;1\rangle \text{—} \circ \text{—} |01;1\rangle = -i, \\ |11;1\rangle \end{array} \quad \begin{array}{c} |11;1\rangle \\ |10;1\rangle \text{—} \circ \text{—} |01;1\rangle = i \\ |00;1\rangle \end{array} \right\} \tag{5.10}$$

$$\underline{qp}\text{-type} \qquad \sigma_y = \left\{ \begin{array}{c} |00;1\rangle \\ |01;1\rangle \text{—} \circ \text{—} |10;1\rangle = -i, \\ |11;1\rangle \end{array} \quad \begin{array}{c} |11;1\rangle \\ |01;1\rangle \text{—} \circ \text{—} |10;1\rangle = i \\ |00;1\rangle \end{array} \right\} \tag{5.11}$$

$$\underline{pq}\text{-type} \qquad \sigma_z = \left\{ \begin{array}{c} |00;1\rangle \\ |10;1\rangle \text{—} \circ \text{—} |01;1\rangle = 1, \\ |11;1\rangle \end{array} \quad \begin{array}{c} |11;1\rangle \\ |10;1\rangle \text{—} \circ \text{—} |01;1\rangle = 1 \\ |00;1\rangle \end{array} \right\} \tag{5.12}$$

$$\underline{qp}\text{-type} \qquad \sigma_z = \left\{ \begin{array}{c} |00;1\rangle \\ |01;1\rangle \text{—} \circ \text{—} |10;1\rangle = 1, \\ |11;1\rangle \end{array} \quad \begin{array}{c} |11;1\rangle \\ |01;1\rangle \text{—} \circ \text{—} |10;1\rangle = 1 \\ |00;1\rangle \end{array} \right\} \tag{5.13}$$

Conclusion

In this chapter, we introduced the $\mathbb{Z}_2 \otimes \mathbb{Z}_2$ symmetry and successfully designed a valid *ansatz* which is $\mathbb{Z}_2 \otimes \mathbb{Z}_2$ symmetric. This will allow for the use of symmetric libraries when working with the *ansatz*.

Chapter 6

Reach of the *Ansatz*

In Ch. 4, the *ansatz* is introduced and it is shown to exclusively represent gauge states from any given Gauss sector K, as defined in Eq. (2.11). In this chapter, we take a step further and seek the answer to: does the *ansatz* represent *all* gauge states from a given Gauss sector K?

We will do so by first 1) creating both an initial gauge state from K and 2) a general gauge preserving operator θ; with those, we will be able to 3) create any general gauge state from any given Gauss sector K and we will 4) recover its TN structure. In other words, the general TN structure of all gauge states. Finally, we will 5) relate that structure with the one from the *ansatz*, i.e. copy tensors on links and vertex tensors on the sites of the lattice.

This proof can be developed using either $\mathbb{Z}_2$ and non-symmetric copy tensors or $\mathbb{Z}_2 \otimes \mathbb{Z}_2$ and symmetric copy tensors. They give equivalent results. We choose to work with $\mathbb{Z}_2 \otimes \mathbb{Z}_2$ to use this Chapter for giving practical comments on the implementation as well.

Even though this *ansatz* has been used in other works [1], we have not found an explicit answer to the question we are raising here. Therefore, this answer is another central result of the thesis.

6.1 Create Initial Gauge State

As a quick recap from Ch. 2, we define a Gauss sector K as the space of states that fulfil Gauss' law in all vertices, i.e.

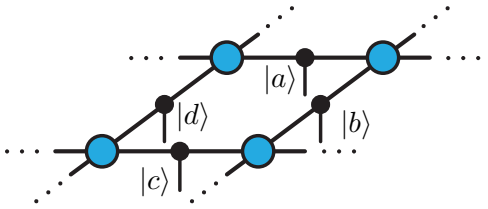
$$K := \Big\{ |\psi\rangle \;\Big|\; \underbrace{A_i |\psi\rangle = (-1)^{g_i} |\psi\rangle}_{\text{Gauss's law}},\; g_i \in \{0,1\},\; \forall i \in \mathcal{V} \Big\}, \tag{6.1}$$

for a given set of background charges $\{g_i\}$ and where

$$A_i := \prod_{l \in i} \sigma_x^l \ .$$

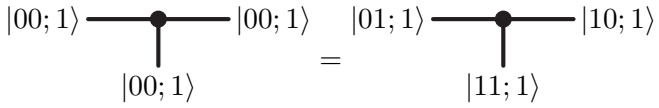
In the last equation, $l \in i$ refers to links l emanating from the vertex i.

We choose the following state as the initial state for its simple TN representation and non-zero overlapping with all states of the Gauss sector under consideration, K:

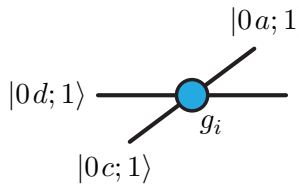


$$|\phi\rangle = \sum_{|abcd\ldots\rangle \in K} |abcd\ldots\rangle\,, \quad (6.2)$$

where $a, b, c, d, \cdots \in \{00, 11\}$ are the physical components of the state, as shown in the diagram. This state $|\phi\rangle$ is represented with the *ansatz* with the components shown in Eqs. (6.3) and (6.4).



(6.3)

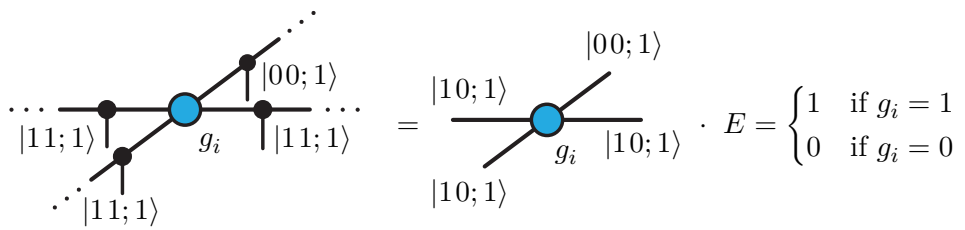


$= 1$ if $a{+}b{+}c{+}d = g_i \pmod 2$, $\forall i \in \mathcal{V}$ (6.4)

The virtual bonds have dimension 2, and their vector spaces are spanned by $\{|00; 1\rangle, |01; 1\rangle\}$. Therefore, $a, b, c, d \in 0, 1$. Examples are shown for qp- and q-type copy and vertex tensors, respectively. The other types have analogous elements.

One can check that the given TN in Eqs. (6.3) and (6.4) reproduces $|\phi\rangle$. It is illustrated in Eq. (6.5) for some background charge $g_i \in \{0, 1\}$, where E represents the environment of the vertex. Similar to how the *ansatz* works, if one were to contract the TN with the bra $\langle wxyz\ldots|$, each copy tensor would evaluate to 1 and project the virtual legs to the subspace corresponding to the physical component —span$\{|00\rangle\}$ or span$\{|01\rangle\}$, in the case of q-type link—. Then, the contraction only gives different than zero when all vertices fulfil Gauss' law, i.e. when $|wxyz\ldots\rangle \in K$, in which case it evaluates to 1. In any other case, some vertex tensor would evaluate to zero, sending the total

contraction and the overlap to zero.



(6.5)

This TN representing $|\phi\rangle$ will be the one used to generate any gauge state from K, by applying a gauge preserving operator on it, as we will see in the following sections.

6.2 Create General Gauge Preserving Operator

An operator θ is *gauge preserving* if the following two conditions are met. First, when taking a vector from a certain gauge sector K, as defined in Eq. (6.1), sends it to still the same gauge sector K. And second, every vector sent to K comes from K. In other words, θ doesn't mix sectors, as summarized in Eq. (6.6).

$$|\eta\rangle \in K \iff \theta\,|\eta\rangle \in K \tag{6.6}$$

Since sectors are defined through $A_i\ \forall i \in$ vertices, θ not mixing sectors could be translated to θ not affecting the outcome of A_i. Following this intuition one can show[1] that the above condition is equivalent to $[\theta, A_i] = 0\ \forall i$. Namely,

$$\textit{gauge preserving} \iff [\theta, A_i] = 0 \quad \forall i. \tag{6.7}$$

We want to see now which form has an operator with this restriction. For that, consider writing θ in the operator Pauli basis

$$\theta = \sum_m c_m \cdot a_m, \quad c_m \in \mathbb{C}, \tag{6.8}$$

where a_m is a tensor product of $\{\mathcal{I}, \sigma_x, \sigma_y, \sigma_z\}$, each acting in one lattice's site. The collection of all possible a_m forms a basis for general operators acting on the lattice. If we now require θ to be gauge preserving, one can see[2] that θ can only have components such that $[a_m, A_i] = 0\ \forall i$. In other words,

$$[\theta, A_i] = 0 \quad \forall i \iff c_m = 0,\ \forall m \mid \exists i, [a_m, A_i] \neq 0. \tag{6.9}$$

As we have seen in the proof (Appendix B.2), $[a_m, A_i] \neq 0$ only happens when an *odd* number of off-diagonal operators (σ_y or σ_z) meet the four σ_x of A_i.

[1] Shown in Appendix B.1.
[2] Seen in App. B.2.

And this has to be fulfilled $\forall i$, i.e. for every vertex. We can visualize this using the following diagrammatic notation:

$$\mathcal{I} \longrightarrow (nothing) \quad \sigma_x \longrightarrow \bullet \quad \sigma_y \longrightarrow \text{—} \quad \sigma_z \longrightarrow \text{—}. \tag{6.10}$$

An illustrative example of an a_m — where $\exists i \mid [a_m, A_i] \neq 0$ — is shown in Fig. 6.1.

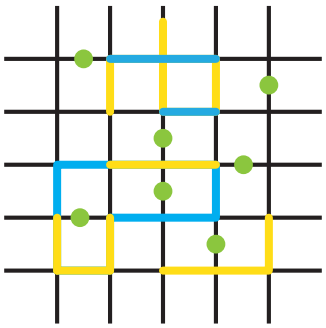


Figure 6.1: Illustration of Pauli operators that form a certain a_m from Eq. (6.8). Notation explained in the main text, Eq. (6.10).

According to the gauge preserving condition from Eq. (6.9), there is total freedom on the position of $\mathcal{I}$ and σ_x, but a_m **can only have never-ending lines**: closed lines (loops) or lines extending to infinity. Remarkably, that is precisely the condition we require in the Sec. 6.4 where we are going to apply the operator.

6.3 All Gauge States

Now that we have both an initial state and a general gauge preserving operator, we claim that for every gauge state $|\psi\rangle \in K$ there is a gauge preserving operator θ that generates it when acting on an initial state $|\phi\rangle \in K$. Indeed, we can always build

$$\theta := \frac{|\psi\rangle \langle\phi|}{\langle\phi \,|\, \phi\rangle}.$$

We can already check it is gauge invariant. Explicitly using the commutator requirement:

$$\theta A_i = \frac{|\psi\rangle \langle\phi| A_i}{\langle\phi \,|\, \phi\rangle} = \frac{|\psi\rangle \langle A_i \phi|}{\langle\phi \,|\, \phi\rangle} = (-1)^{q_i} \frac{|\psi\rangle \langle\phi|}{\langle\phi \,|\, \phi\rangle}, \qquad \forall i,$$
$$A_i \theta = \frac{A_i |\psi\rangle \langle\phi|}{\langle\phi \,|\, \phi\rangle} = (-1)^{q_i} \frac{|\psi\rangle \langle\phi|}{\langle\phi \,|\, \phi\rangle}, \qquad \forall i.$$

Hence, $[\theta, A_i] = 0, \ \forall i$; which shows that θ is gauge preserving, according to Eq. (6.7).

Therefore, all gauge states can be written as

$$\theta \left|\phi\right\rangle = \sum_{m}^{M} c_m \cdot a_m \left|\phi\right\rangle \tag{6.11}$$

Notice we haven't used any property of $|\phi\rangle$, besides that it belongs to K. In this chapter, we use the specific form of $|\phi\rangle$ defined in Eq. (6.2) for the reasons mentioned in that Section.

6.4 Recover TN Structure

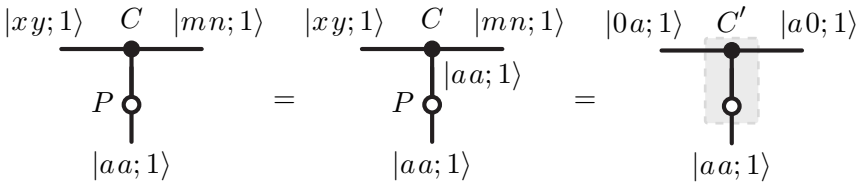
In order to recover the TN structure of Eq. (6.11) we focus first on the terms $a_m \left|\phi\right\rangle$ and then on their linear combination.

6.4.1 Individual Terms

As illustrated in Fig. 6.1, on $a_m \left|\phi\right\rangle$, a single local Pauli operator acts on each copy tensor, and vertex tensors may absorb an auxiliary leg from a_m. Let's see their effect on the local structure.

Diagonal Pauli Operators

Consider the Pauli operators $P \in \{\mathcal{I}, \sigma_x\}$. As shown in Eq. (6.12) — where $a, m, n, x, y \in \{0, 1\}$—, they do not change the charge sector of the physical leg and thus the result C' — highlighted in the equation — is simply proportional to the original copy tensor. Hence, the result is still a copy tensor.



(6.12)

This was shown for qp-type copy tensor. The pq-type result is analogous.

Off-Diagonal Pauli Operators

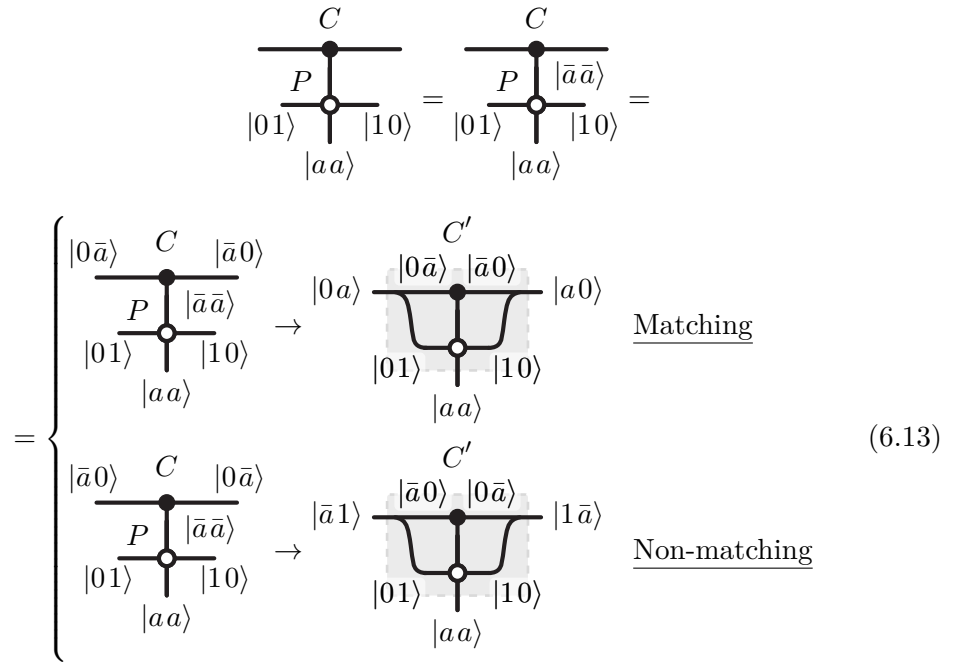
Consider the Pauli operators $P \in \{\sigma_y, \sigma_z\}$. Since these operators can also be pq- or qp-type, we distinguish two scenarios: when operator and copy tensor types match and when they do not.

Similar to before, starting from the physical vector space, we track the charge sectors[3] until we reach the final contraction, C', highlighted in the equation

[3] Degeneracy labels have been avoided in Eq. (6.13) for style reasons. The degeneracy is the same one as the diagonal operators of Eq. (6.12).

Eq. (6.13). Similarly, $a \in \{0, 1\}$ and its opposite charge is denoted as $\bar{a}$. Meaning, $\bar{a} = a + 1 \pmod 2$. It is shown for qp-type Pauli operator. For pq-type results are analogous.

In the equation, the charge sector changes through the operator and the copy tensor initially copies the opposite charge. Then, reshaping back into a single virtual leg is done with a symmetric isometry[4]. Accordingly, if the tensor types match, the copied charge will reverse back into the original, thus preserving the copy tensor structure. Otherwise, it will not.



(6.13)

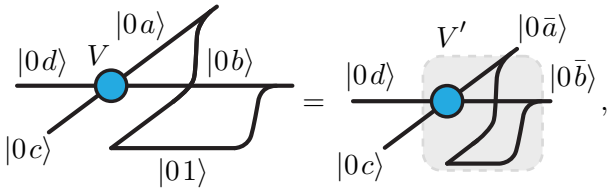
We can see how matching types are needed in order to preserve the copy tensor structure. This need can always be satisfied. Namely, when programming the gates in the code, we must (and *can*) ensure this matching by using the appropriate type of operator on each link.

Auxiliary Leg on Vertex

These auxiliary legs are not identities, but rather a projector into the subspace $\text{span}\{|10; 1\rangle\}$ or $\text{span}\{|01; 1\rangle\}$, if the leg is type p or q type, respectively. This absorption is an outer product followed by a symmetric isometry (similar as

[4]This is also a symmetric tensor, and thus charges must be conserved and the sum-rule fulfilled.

before). The absorption is illustrated in the following equation



(6.14)

where charge notation is similar to Eq. (6.13) and the case for p -type vertex tensor is analogous. In order for the structure to be preserved we must check if $\bar{a}+\bar{b}+c+d = g_i \pmod 2$, where g_i is the background charge at the considered vertex, i. And indeed that is the case since:

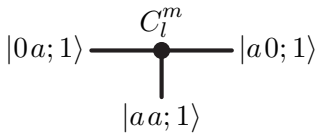
$$\bar{a}+\bar{b}+c+d = a+1+b+1+c+d = g_i \pmod 2. \tag{6.15}$$

Again, here the type of link and vertex have to match. Otherwise, sectors like $\text{span}\{|1a\rangle\}$ would appear, which is not part of the vertex tensor as we have defined it.

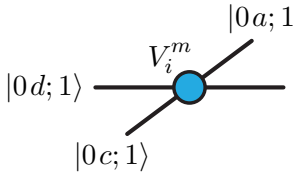
In conclusion, we have seen how $a_m |\phi\rangle$ locally preserves[5] the structure, which implies that $a_m |\phi\rangle$ retains the *ansatz*'s structure. Therefore $\theta |\phi\rangle$ is a linear combination of bond-2-TN with the *ansatz*'s structure.

6.4.2 Linear Combination

In this section, we will show examples of qp- and q-types tensors, for the other types the result is analogous however. Consider that, for a particular θ, the linear combination has $M \in \mathbb{Z}$ terms. Consider then the term $m \in \{1, \dots, M\}$: $c_m \cdot a_m |\phi\rangle$, a TN with the *ansatz*'s structure. Let the copy tensor in link l and the vertex tensor in vertex i from that m term be C_l^m and V_i^m, respectively. The vector space in their virtual legs has dimension 2, e.g., the vector space of the right leg of a qp-type C_m^l is $\text{span}\{|00;1\rangle, |10;1\rangle\}$. The components of these tensors are shown in Eqs. (6.16) and (6.17), where $a, b, c, d \in \{0, 1\}$.



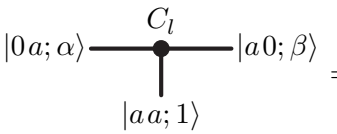
$$= [C_l^m]_{11}^{aa} \tag{6.16}$$



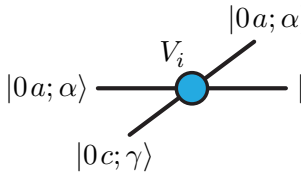
$$= \begin{cases} [V_i^m]_{1111}^{abcd} & \text{if } a+b+c+d = g_i \pmod 2 \\ 0 & \text{otherwise} \end{cases} \tag{6.17}$$

[5] If matching types of virtual and auxiliary links.

Then, one can get a TN representation of $\theta\,|\phi\rangle$ with tensors C_l and V_i at link l and vertex i, respectively, with the following elements:



$$= \delta_{\alpha,\beta}\cdot[C_l^{\alpha}]_{11}^{aa} = \alpha \begin{pmatrix} \overset{\beta}{\longrightarrow} & & & \\ \left[C_l^1\right]_{11}^{aa} & 0 & \dots & 0 \\ 0 & \left[C_l^2\right]_{11}^{aa} & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & \left[C_l^M\right]_{11}^{aa} \end{pmatrix}, \tag{6.18}$$

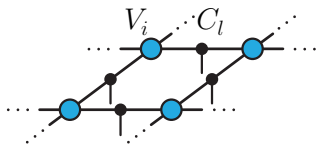


$$= \delta_{\alpha,\beta}\delta_{\beta,\gamma}\delta_{\gamma,\delta}\cdot[V_i^{\alpha}]_{1111}^{abcd}\,, \tag{6.19}$$

where now the virtual leg's vector space is $2M$-dimensional, e.g.:

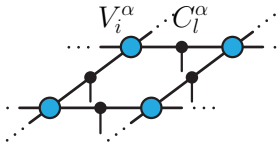
$$\mathrm{span}\{|00;1\rangle\,,\dots,|00;M\rangle\,,|01;1\rangle\,,\dots,|01;M\rangle\}. \tag{6.20}$$

Accordingly, $\alpha,\beta,\gamma,\delta \in \{1,\dots,M\}$. One can check that Kronecker's deltas cancel all sums in the contraction but one, as shown in Eq. (6.21), recovering the original linear combination.



$$= \overbrace{\sum_{a,\alpha}\cdots\sum_{b,\beta}}^{\text{for each virtual bond}}\Big(\underbrace{\delta_{\alpha\beta}\cdot\dots}_{\text{across each tensor}}\Big)\cdot\left([C_l^{\alpha}]_{11}^{aa}\cdot\dots\right) =$$

$$= \sum_{\alpha=1}^{M}\left(\sum_{a=0}^{1}\cdots\sum_{b=0}^{1}[C_l^{\alpha}]_{11}^{aa}\cdot\dots\right) =$$

$$= \sum_{\alpha=1}^{M}$$  $$= \sum_{m=1}^{M} c_m\cdot a_m\,|\phi\rangle =$$

$$= \theta\,|\phi\rangle \tag{6.21}$$

Therefore, this shows that all gauge states from given K can be written in this diagonal structure[6].

[6] We note that this representation might be optimised, such that changing the gauge and truncating would yield the same structure with dense tensors (but still copy tensors) and lower bond dimension.

6.5 Relate To the *Ansatz* Structure

So far, we have been working with three (presumably) different sets:

- K: all gauge states,
- P: all states with diagonal structure,
- Q: all states represented by the *ansatz*.

We can claim

$$Q \overset{(A)}{\subset} K \overset{(B)}{\subset} P. \tag{6.22}$$

As explained in Ch. 4, all states represented by the *ansatz* are gauge states. That is, all elements from Q are in K, proving (A). Furthermore, up to this point in the chapter, we have shown that all gauge states have diagonal structure, i.e. (B).

Moving on, now it is straightforward to see that

$$P \subset Q, \tag{6.23}$$

since elements from P have a copy structure with the peculiarity of being diagonal. That is still included in *all* TN with the *ansatz* structure, i.e. Q.

Finally, bringing it all together, we write

$$Q \subset K \subset P \subset Q \tag{6.24}$$

which implies

$$Q = K = P. \tag{6.25}$$

Therefore, the *ansatz* represents *all* gauge states from a given Gauss sector K. As well as the introduction of $\mathbb{Z}_2 \otimes \mathbb{Z}_2$ symmetry in the Gauge Theory context, the result from Eq. (6.25) is part of the main results.

Chapter 7

Benchmark Results

We study the system of an infinite lattice to test the *ansatz* using this $\mathbb{Z}_2 \otimes \mathbb{Z}_2$ symmetry. Moreover, we are also going to look at the finite case with a 2×2 lattice. We do so with the aim of better understanding behaviours observed in the infinite case, which we explain in Sec. 7.1.

In both study cases, we are interested in finding out the ground state of the Hamiltonian $\mathcal{H}$ that governs the system, Eq. (2.3). As seen in Ch. 3 and Appendix. A, the method for finding the ground state uses an evolution on complex time, which ideally evolves any state towards the ground state, steadily decreasing the state's energy.

We investigate the energy of the evolved state as time passes, i.e. versus the number of time steps performed. We consider scenarios with bigger or smaller time steps, β, various parameters of the Hamiltonian, J and h, and different maximum bond dimension $D_{\max}$.

Features of the results are commented on based on these possible sources of error. First, we consider the *Trotter error* due to the decomposition of the evolution operator. Second, the approximation of the environment can also impact the results. And last, we take into account the *truncation error* which originates from discarding components of the tensors

7.1 Infinite Case

Results are produced for an infinite lattice tilled with a unit cell of size 2×2. See Sec. 5.3 for details on the unit cell. Moreover, we work without background charges. They can be easily introduced back in the code. Qualitatively we get similar values, but we have not focused on studying the code's performance for different charges in detail. That is because, as shown in this chapter, we already found convergence issues in the simple case without background

charges. Motivated by solving this issue, we fixed the background charges and later studied the finite 2×2 case.

First, we show the convergence of our algorithm for individual time steps and then argue about the accuracy of the result by comparing different time steps. After that, we show the unusual dependence of the accuracy with β and finally, we mention an odd aspect of the evolution when reaching absolute minimums.

7.1.1 Convergence

When running the algorithm with different random initial states, we find the results shown in Fig. 7.1. The main features are discussed below, with a more detailed explanation of potential causes provided in the following sections.

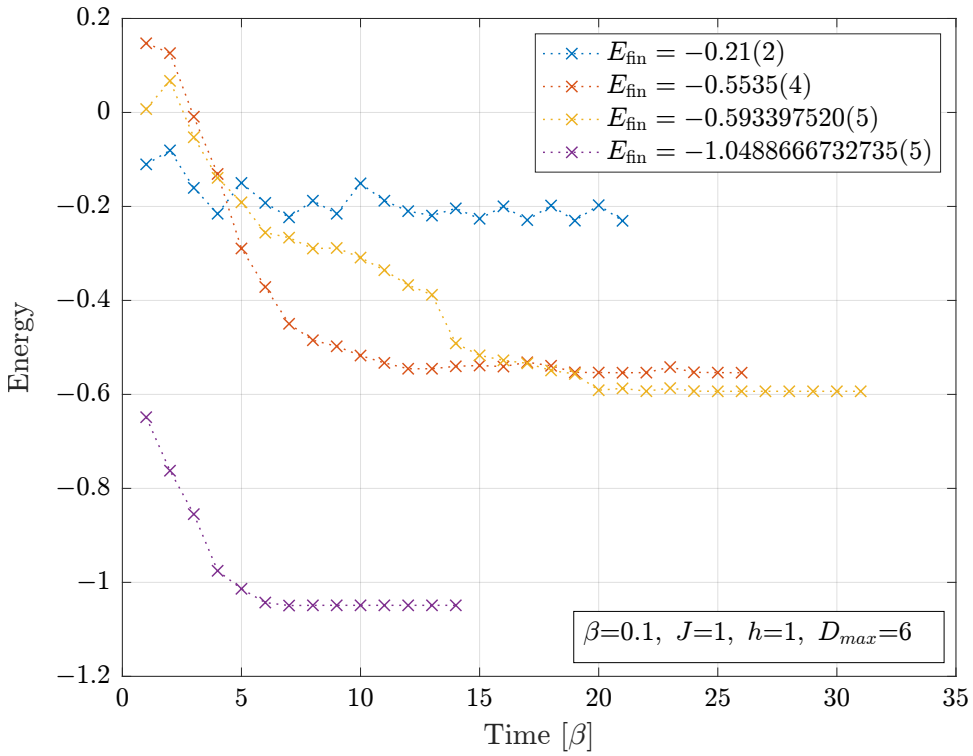


Figure 7.1: Energy of the evolved state for different random initial states. The same time step, $\beta = 0.1$, is used for all lines. The algorithm gets stuck along the way, at the energies shown in the legend. These energies are given up to the last stable decimal when the evolution converges. Notice, for instance, how the *convergence* of the blue line oscillates around a central value, $E_{\text{fin}} = -0.21$.

The energy of the evolutions in the figure always converges[1]. However, it does so at different energies. In some cases, the energy it converges to is

[1] In practice, it is also found that the energy always converges.

incredibly precise: for long times the energy changes on the 15^{th} decimal or it doesn't change at all. See the purple line of Fig. 7.1, $E_{\text{fin}} \approx -1.048$. For unstable convergences, the algorithm gets stuck between certain energy values and jumps between them. This case can be seen in the blue line of Fig. 7.1, $E_{\text{fin}} = -0.21(2)$.

Moreover, repeated executions of the algorithm — with different random initial states — get stuck in the same vicinity of values. That indicates that the converged energy is not random. It could be related to the spectrum of the Hamiltonian (excited states). In the figure, we can see two such convergences, the ($E_{\text{fin}} = -0.55\ldots$) yellow and ($E_{\text{fin}} = -0.59\ldots$) orange lines.

7.1.2 Accuracy

Even though the algorithm gets stuck, we study the best convergence value for different time steps, hoping they converge to similar physically meaningful values. The comparison is shown in Fig. 7.2.

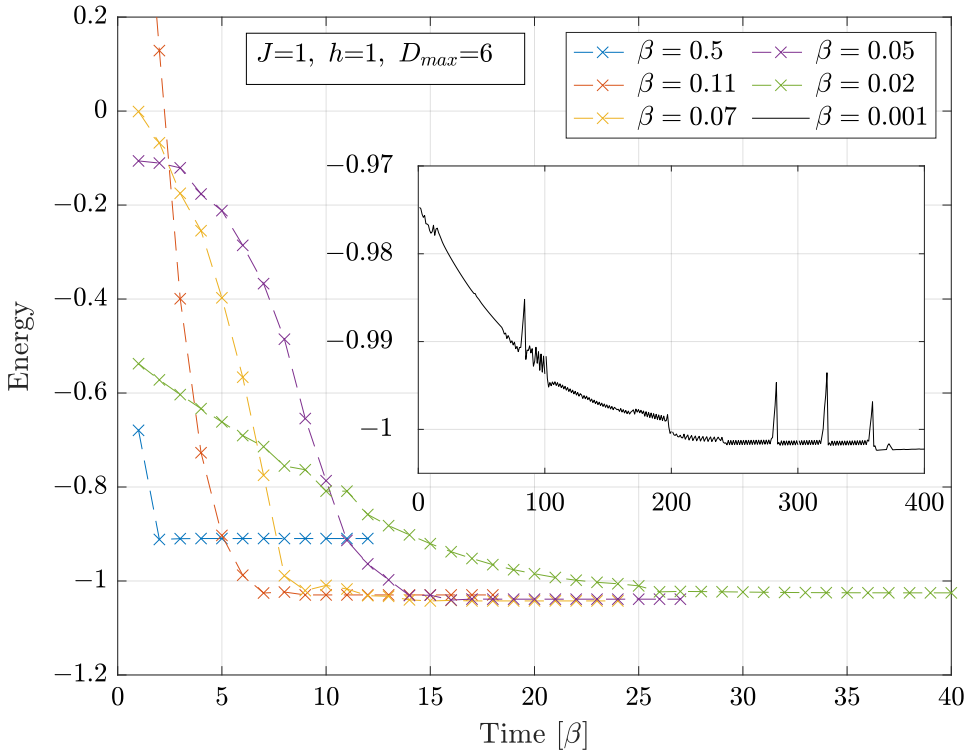


Figure 7.2: Energy of the evolved state for different random initial states and time steps, β. For each β, only the lowest convergence is shown. Notice how the bigger the time step, the faster the convergence. Convergence energies are shown in Fig. 7.3. They are similar for all β's, except for $\beta = 0.5$.

The figure shows that the algorithm converges at similar low values for different time steps and initial states. This leads us to assume that the converged energies approach the ground state energy and it is not another stuck point. In other words, this favorable results shows that the algorithm can successfully reach the ground state despite getting stuck in some cases.

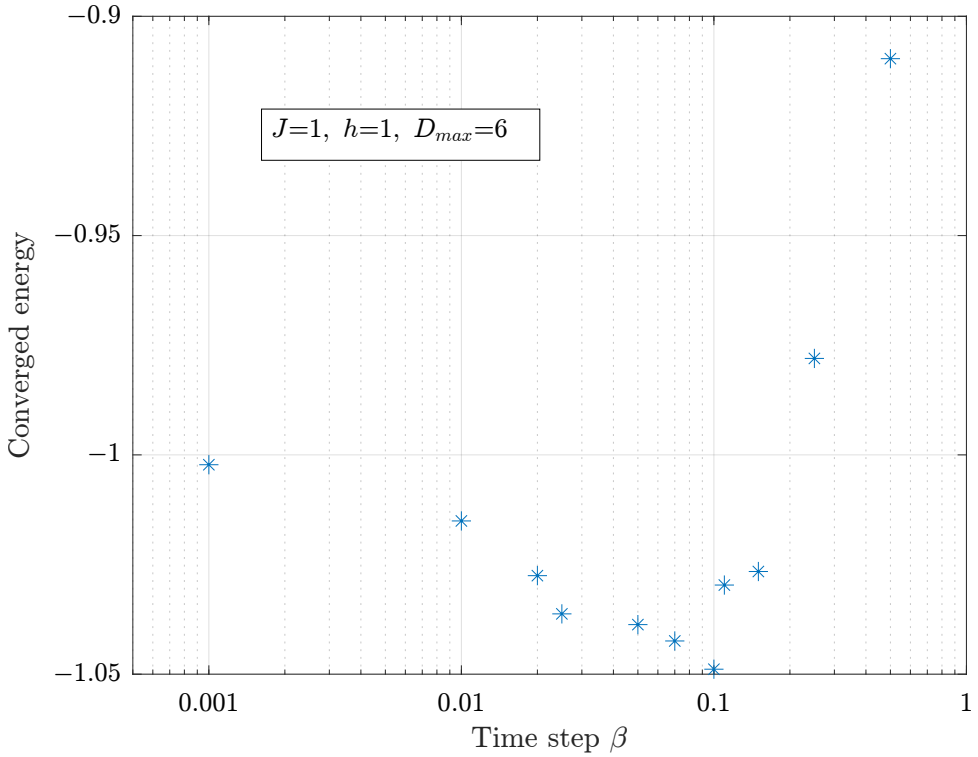


Figure 7.3: Lowest (non-stuck) converged energies for multiple time steps, β. Note the logarithmic scale on the x axis. A singular time step $\beta^\star$ at 0.1 gives the best converged energy. For β's different from $\beta^\star$ the final converged energy seems to grow. This can be expected for higher time steps due to Trotter error since the error is $\mathcal{O}(\beta^3)$. However, the lower regime behaviour is unexpected and focused on during the following sections.

The converged energies are shown in Fig. 7.3. The figure shows how above a certain time step threshold, around $\beta^\star$=0.1, the converged energy quickly rises. We can attribute this to dominating Trotter error which goes as $\mathcal{O}(\beta^3)$. Under the threshold, however, the converged energy goes slowly upwards, indicating that the other sources of error come into play.

7.1.3 Adaptive Evolution

We tune the time step during the evolution to further illustrate the unusual dependence of the non-Trotter error. In this simulation, after an evolution

with some β has converged, we start another evolution with a different time step, starting from the previous final state. We hope that even if lower time steps converge to higher energies on their own, as in Fig. 7.3, if they start past their convergence points —reached by the previous β— they might be able to converge to lower energies. Moreover, this adaptive evolution is a typical strategy used to increase the accuracy of the results in Suzuki-Trotter-based time evolutions. Thus, it beneficial on its own to study this strategy's performance. Results are shown in Fig. 7.4.

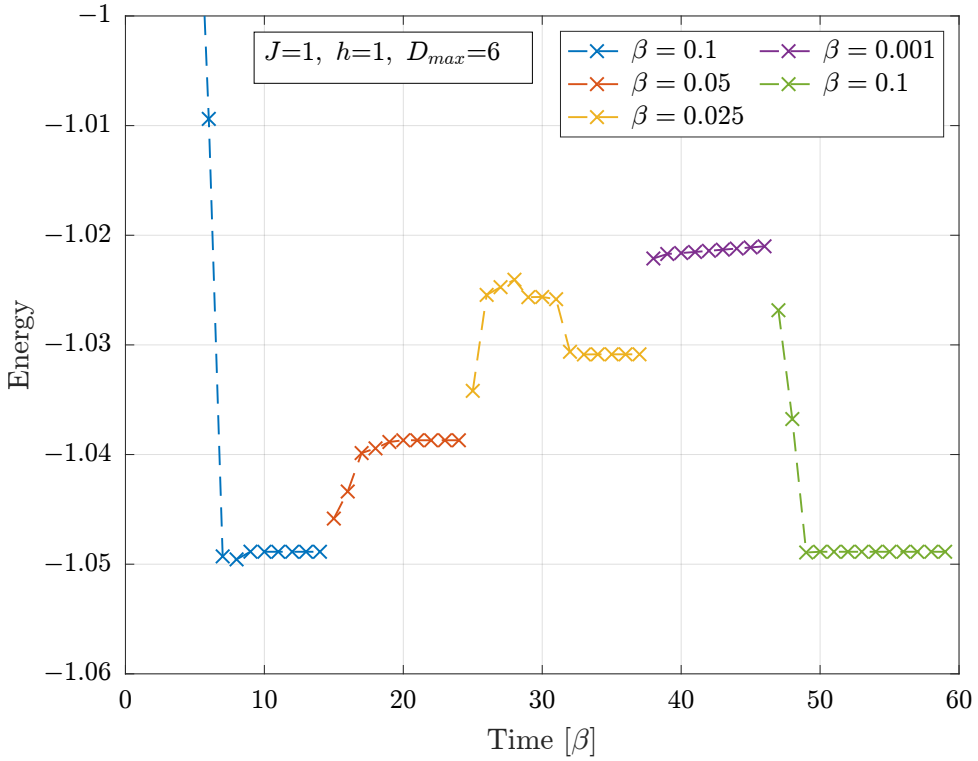


Figure 7.4: Energy of the evolved state as the time step is changed, as explained in the main text. Contrary to what was expected, even starting from low energy, small β makes the energy converge upwards. After the smallest time step, $\beta = 0.01$, the larger initial time step $\beta = 0.1$ can still recover the lowest convergence energy.

Results show that the adaptive strategy can not be used to increase accuracy in the current situation. In fact, starting from a low energy, upon decreasing the time step, an upwards convergence is seen. One explanation we consider here, is that the decrease of energy caused by applying one gate, is overcompensated by the error in the algorithm, overall increasing the energy. The second β=0.1 evolution shows how the energy can be made to converge once more to low energy by using bigger time step, i.e. bigger energy decrease per gate applied. In ideal cases, converged energies are product of the fixed point of the Suzuki-Trotter-decomposed evolution operator, i.e. the state yielded after several

applications of the operator. Results show that this is not the case here, that the convergence here is due to the balance of a energy-decreasing mechanism (the operator) and a energy-increasing mechanism (the error).

7.1.4 *Bouncing* Off the Ground State

In general energy evolutions, we observe what we denote here as a *bouncing* effect: the energy reaches an absolute minimum before immediately increasing again to its final convergence value, see the inset of Fig. 7.5. On the main plot of Fig. 7.5 the *bouncing* effect is seen to be a general behaviour.

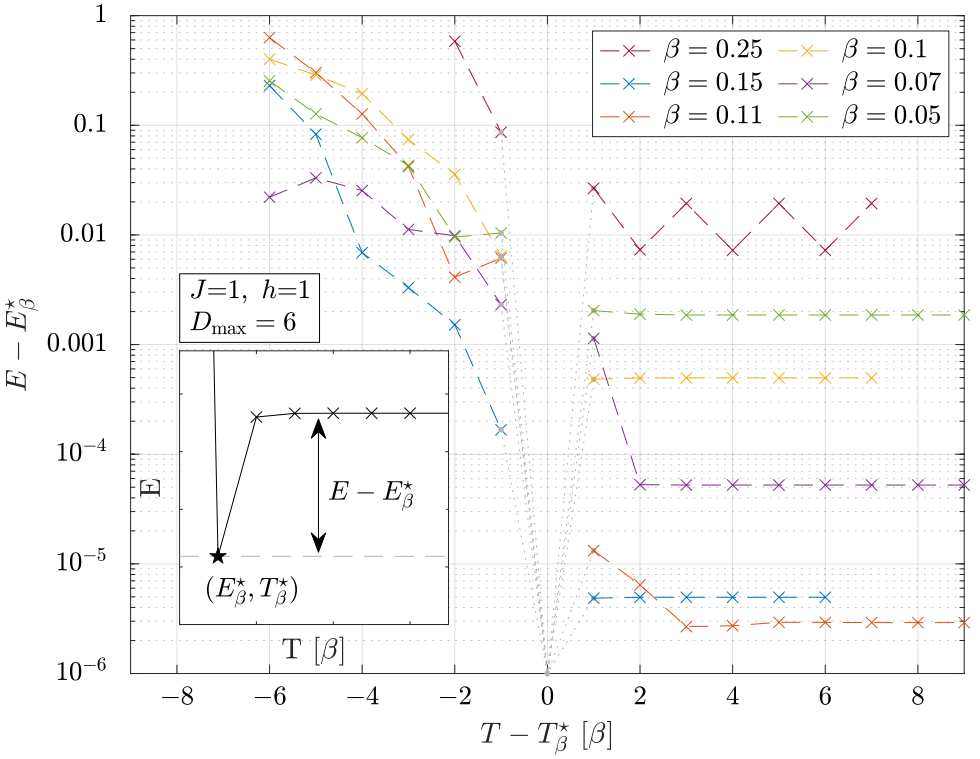


Figure 7.5: **Inset.** β-evolved state's energy zoomed in around the absolute minimum. This minimum is at $(E^\star_\beta, T^\star_\beta)$. After this point, a general trend is observed: the energy *bounces up* and converges at a higher value. Bounce energies $E^\star_\beta$ are given in Table 7.1. **Main plot.** For evolutions with the legend's βs, the energy difference $E-E^\star_\beta$, depicted in the in the inset, is plotted. The bounce is represented at $T-T^\star_\beta=0$ with dotted grey lines. Consider the yellow line β=0.1 as a reading example. Its final energy difference converges around 5×10^{-4} and from the table we read $E^\star_{0.1}=-1.0494$. The plot shows how the energy *bounces up* from $E^\star_{0.1}$ to $-1.0494+0.0005=-1.0499$. Moreover, the plot illustrates how this *bouncing* behaviour is generally observed for *multiple* evolutions, even if it occurs at different scales.

β	$E^\star_\beta$
0.25	-0.99
0.15	-1.026597
0.11	-1.029681
0.10	-1.0494
0.07	-1.04248
0.05	-1.041

Table 7.1: Bounce energies from Fig. 7.5 for each β-line. The decimals are given according to the magnitude of the final energy difference $E - E^\star_\beta$ in the plot.

Considering only the Trotter error, the *bouncing* behaviour can not be explained. In the absence of other errors, the energy steadily decays. Therefore, the effect should be explained considering other sources of errors. Moreover, the fact that the effect is observed for general evolutions points at the fact that it is not a simple anomaly. Thus, this behaviour gives direct insight on the nature of the energy-increasing mechanism.

As previously mentioned, one possible explanation for this behaviour is the competition of two energy-changing mechanisms in the evolution. On the one hand, the application of the evolution operator (gate) *wihtout* truncation decreases the energy of the state, i.e. projects out high-energy components. This has been seen in the first steps of evolutions with large $D_{\max}$[2], where truncation is not yet performed. On the other hand, we now consider the truncation as the mechanism raising the energy. The error introduced by the truncation populates low-lying excited components.

These considerations lead to the following explanation of the *bouncing* effect. Initially, there are many high energy components to project out. Therefore, the gate severely decreases the energy, unaffected by the truncation populating low-lying excited components. At the minimum, the evolved state only has low-energy components, weakening gate's effect: it now decreases the energy slower. However, we *assume* the truncation still populates excited components, even for a low energy state. After the minimum, the rate at which the truncation populates excited states is greater than the energy-decrease rate of the gate. Therefore, the energy grows. It does so until the rates of both mechanism are matched, at which point the energy converges. In this way the explanation covered all features of the observed behaviour.

[2]For better testing, one should set $D_{\max}$ large only a few time steps before the bounce, since large $D_{\max}$ only allows for a few steps before saturating the computer. This has not been prioritized in the numerical study.

Another possible explanation would be that the environment approximation makes the algorithm moderately non-variational. Meaning, optimising the truncation of a subnetwork of the TN causes the rest of the

Recall that we extract a subnetwork of the full TN when we apply the gate and truncate afterwards, as seen in Eq.(3.27) and in in Appendeix A. The truncation should produce a subnetwork of bond dimension $D_{\max}$ as close as possible of the contraction of the *one* plaquette. However, the ultimate goal is instead to produce a full TN optimally close to the contraction of the *full* gate. We are simply using a strategy in which we optimise the tensors locally. It can happen that the optimal local strategy —with the considered extraction of the subnetwork[3]— does not produce an optimal TN. Meaning, the energy is locally lowered but at the expense of global energy increase. We refer to this scenario as a non-variational algorithm. The *bouncing* effect can be seen as the point where the local strategy is no longer variational, thus converging upwards.

For easier numerical implementations and, importantly, to rule out the effect of the environment, we consider a finite lattice. We hope that similar behaviour is observed in a scenario where there is no environment truncation. That would allow us assert the presence of truncation error causing this particular effect.

7.2 Finite 2×2 Lattice

The finite lattice we work with in this section is a simple square lattice with two vertices per side. Meaning that there will be one copy tensor and two vertex tensors per side in the TN we will use. Such TN is shown in Fig. 7.6. This system has only four physical sites —diagonal legs in the figure—. Considering the physical meaningful subspace at each site, $\{|00;1\rangle, |11;1\rangle\}$, as defined in Eqs. (5.3) and (5.4), the total Hilbert space has dimension 16, which allows for exact diagonalisation and easier analysis of the results. Moreover, fixed boundary conditions are imposed by projecting the virutal legs.

7.2.1 *Bouncing* Off the Ground State

The *bouncing* effect observed in the infinite case explained in the previous section, is still seen and with more accuracy in this finite scenario, as shown in Fig. 7.7. In this case, the energy is compared with the exact ground state energy, instead of the bouncing energy of Fig. 7.5. Since we also observe this

[3] A local strategy can give good results if the effect of the rest of the TN is adequately considered. That means, setting more robust schemes for representing the environment instead of the current diagonal matrices. Ideally, the local subnetwork should be related to the reduced density matrix result of the partial trace of the environment.

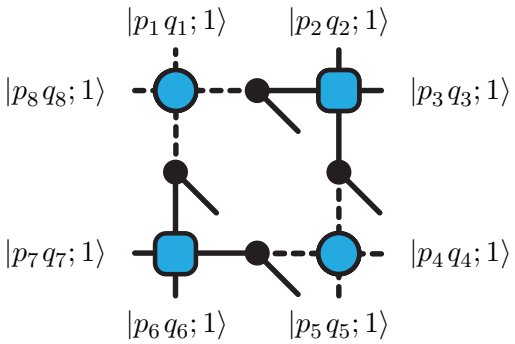


Figure 7.6: Finite 2×2 lattice studied in Sec. 7.2. Four random initial vertex and copy tensors are initialized with initial bond dimension 2. Tensors' types follow the notation: dashed line are p-type bonds, straight lines are q-type bonds and the rest of the tensor types follow accordingly. Fixed boundary conditions are imposed projecting the open virtual legs into a 1-dimensional subspace with the indicated charges. In this work, we focused on boundary conditions $p_q = q_i = 0,\ \forall i \in \{1, \ldots, 8\}$.

effect in the finite case, we conclude that the mechanism responsible for it does not come from environment approximation.

7.2.2 Resetting Evolution

Remarkable accuracy can be reached in the *bouncing* point, which leads us to approach that point slower in the hopes of getting better accuracy. Moreover, we realised that after the *bounce*, the convergence can not be consistently improved, even fine-tuning the time step. Accordingly, we look to avoid the *bouncing* point by resetting the evolution to a starting state before the *bounce*—see the inset from Fig. 7.8—. We start the evolution again from there with a new time step. Then, in general, the new evolution keeps on decreasing the energy and reaching closer to the ground state energy before *bouncing* again. An evolution with multiple resets is shown in Fig. 7.8.

In this finite lattice and with this resetting method, the ground state energy can be accurately described up to very high precision. However, something clearly disturbs the evolution in the *bouncing* point, which we study further in the next section.

7.2.3 Known Fix-point

Consider the state

$$|\eta\rangle := \bigotimes_{l=1}^{4} |+\rangle_l \quad \text{with} \quad |+\rangle_l := |00; 1\rangle_l + |11; 1\rangle\,, \tag{7.1}$$

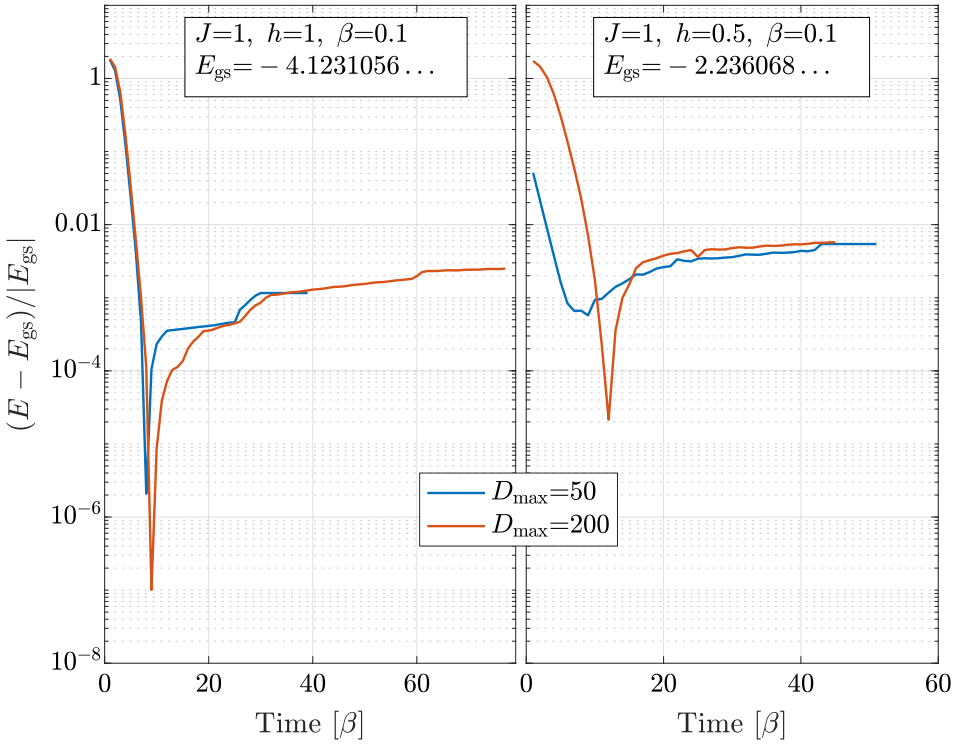


Figure 7.7: Relative energy difference between exact ground state energy E_{gs} and the evolved state's energy E, for $D_{max} \in \{50, 200\}$ and β=0.1. The left and right-hand side plots use different parameters for the Hamiltonian, shown in the corresponding annotations. In both cases, the *bouncing* behaviour from Fig. 7.5 is still observed[†]. The results are more accurate for higher maximum bond dimension D_{max}. Finally, the accuracy of the right plot is worse than the left plot's. It seems that the ground state for J=1, h=0.5 is more complex than the J=1, h=1 one, signaling

[†] This time however, the comparison is more accurate, since we use the exact ground state energy.

where l labels the physical sites. This state is the ground state of the Hamiltonian for $J = 1$ and $h = 0$. We can represent $|\eta\rangle$ with our *ansatz* with a bond dimension 2. Since it is the ground state — an eigenvector of the Hamiltonian — it shouldn't change during the evolution of the algorithm. We start the evolution with this state and give the results in Tables 7.2a-7.2c for different maximum virtual bond dimension.

Results from the tables have a fundamental issue: the bond dimension grows. As we said, the state remains the same, $|\eta\rangle$, and it can be represented with a TN of bond dimension 2. Everything more than that results in an inefficient way of representing the state. The fact that the bond dimension still grows in the evolution shows that the algorithm cannot truncate away the inefficient

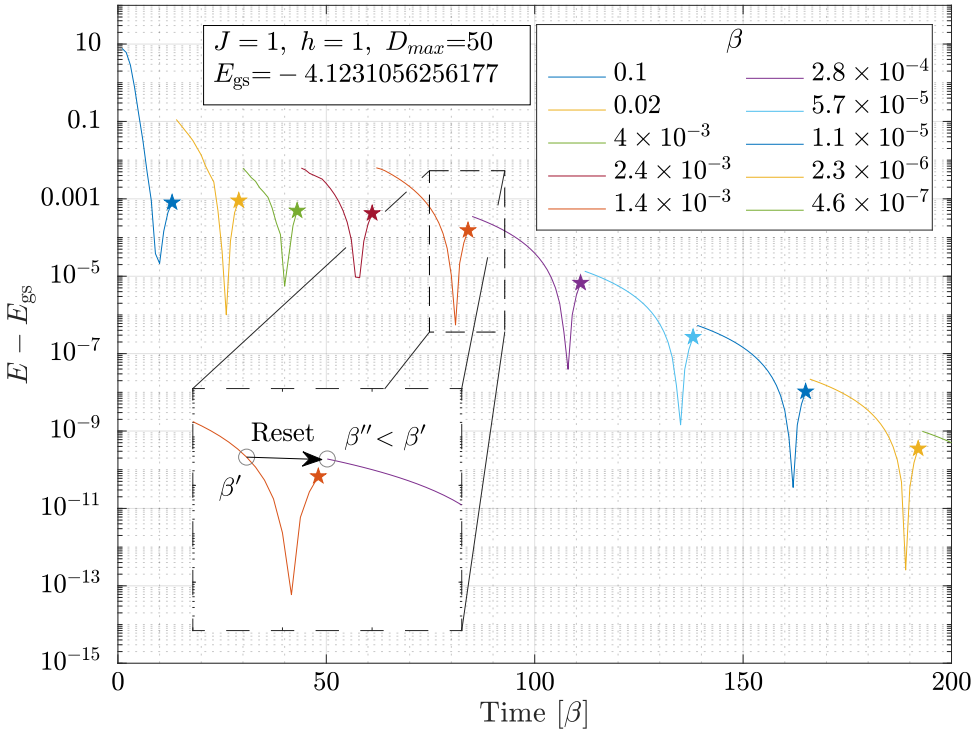


Figure 7.8: Energy difference between the evolved state, E, and the ground state, E_{gs}, with parameters shown in the plot's annotation. After the energy *bounces*, the evolution is reset: a new evolution starts with lower β and a starting state before the bounce point. Depicted in the inset. With this method, the energy difference is systematically reduced up to 10^{-13}, before the evolution is manually stopped.

components. This is made more clear the moment truncation starts. Not only does the energy increase, as expected with any approximation such as truncating, but it ends up giving imprecise results as the evolution continues, similar to the final convergence after the *bouncing* effect.

7.3 Candidate Problem

The behaviour of the errors does not match the expected behaviour from considering only Trotter errors. We have argued that there should be another mechanism influencing our results. The other central approximation in this work is working with a finite bond dimension, which we maintain under some threshold by truncating away specific components. Therefore, the truncation method is a probable candidate for the source of error. We considered the environment approximation as a source of error, however in the last section is shown that the *bouncing* effect can be reproduced only with truncation error.

Table 7.2: The TN's bond dimension, D, and energy difference between the evolved state's energy, E, and the exact ground state, $E_{gs} = -1$, as the state evolves (with $\beta = 0.1$ time steps). Three cases are presented with different maximum bond dimension, given in the corresponding captions, all with $J = 1$, $h = 0$. Time is not given for the **(b)** and **(c)** tables, but it is similar to **(a)**. The time at which truncation starts is indicated with the dashed line (notice that after that $D = D_{max}$). Only the magnitude of the energy difference is given. Vertical dots on the third row mean that the evolution kept the energy difference at zero while doubling the current bond dimension. The last vertical points indicate that the evolution continued without ever converging back to E_{gs}.

(a) $D_{max} = 50$

Time	$E - E_{gs}$	D
0β	0	2
1β	0	4
2β	0	8
3β	0	16
4β	0	32
5β	10^{-7}	50
6β	10^{-7}	50
7β	10^{-4}	50
⋮	⋮	⋮

(b) $D_{max} = 500$

$E - E_{gs}$	D
0	2
0	4
⋮	⋮
0	64
0	128
10^{-14}	200
10^{-11}	200
10^{-6}	200
⋮	⋮

(c) $D_{max} = 2000$

$E - E_{gs}$	D
0	2
0	4
⋮	⋮
0	512
0	1024
10^{-16}	2000
10^{-11}	2000
⋮	⋮

In 1D systems, it is known how to minimize truncation errors [53]. One can bring the TN into the so-called *canonical form*. In such form, the TN on each side of the link represents an orthonormal basis, and the link includes some positive weights, so-called *singular values*. Crucially, in 1D systems, one side of the link is not connected to the other side — except through the considered link—. Discarding the components with lower singular values achieves an optimal truncation. That means: the truncated state with reduced bond dimension is the closest to the original state with higher bond dimension —where the distance (or error) is measured w.r.t. the Frobenius norm—.

And not only that, but such truncation can also remove *internal correlations*. These are components of the tensors which do not contribute at all to physically meaningful results. That implies that they are inefficient components of the TN and lead to inefficient representations of the target states. When truncating, detecting and removing these components before removing meaningful ones is essential. These internal correlations will be explained in more detail in the next chapter.

For 2D systems, or in general, systems with loops[4], a TN representation which allows for optimal truncation, a *canonical form*, is still an active research field. Accordingly, the truncation method of this work is not optimal. The truncated state is not maximally close to the original one, and more importantly, the internal correlations are not distinguished and, therefore not adequately removed.

We argue that in our case, the algorithm suffers especially from internal correlations due to the loop operator of the Hamiltonian (the plaquette term). It stresses the key feature of (non) separability of the TN around a link. We are led to this conclusion, especially after the results on Table 7.2, where for even such a simple case the bond grows systematically. We study it in more detail in the next chapter.

[4]The characteristic mentioned before for 1D systems — that around a link the TN is split into two disconnected and independent parts — is so crucial because it distinguished them from systems with loops, like square lattices.

Chapter 8

Internal Correlations

As explained previously, in order to do numeric calculations we need to keep our objects of study manageable, i.e. a finite size. For that, we limit the bond dimension the TN can have by a certain D_{max} and make sure it doesn't grow past that truncating when needed.

For a given virtual link of the TN, with its corresponding vector space, the general idea of truncating a link means discarding some vectors of its space. Only keep a certain subspace of that vector space, and accordingly, only keep the sectors of the surrounding tensors that act on the remaining subspace. Effectively, this means only keeping certain components of those tensors. With that, we create a new TN with a reduced bond dimension that represents a new state.

The optimal truncation is that in which the new state is as close to the old one as possible, with respect to some measure which gives a sense of how close two states are. Typically, the Frobenius norm is chosen for such a measure.

In the formalism of TN, we have a certain freedom that we can use in order to improve truncation methods: the gauge freedom, as reviewed in Appendix A. With that, we can change the link's vector space such that the discarded vectors are the ones contributing the least to the contraction result. Thus, making the truncated state closer to the original one.

Luckily, there may be some vectors in the TN representation which are known that contribute exactly nothing to the contraction result. They have no physical meaning. These are called *internal correlations*, to be explained in more detail below, and are the main focus of this chapter.

Knowing these vectors exist, we could find them, and remove them, which won't change the state. Only after would we apply our general truncation method. In this way, the truncation has a better starting point and, since the

dimension has already been reduced, it has to truncate away fewer physically meaningful vectors. Not so luckily, dealing with internal correlations turns out not to be so easy in 2D, as we will also see.

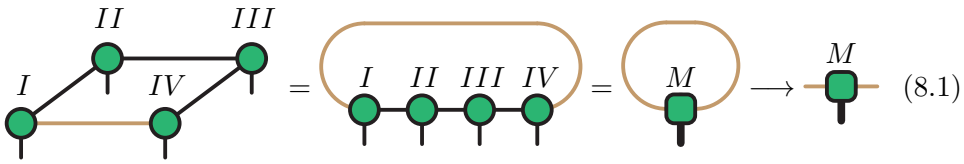
In 1D or TN without loops, these two steps — removing internal correlations and truncating optimally — are done at the same time, thanks to the *canonical form*. When the TN is in this form, there is a known optimal truncation method [53]. Therefore internal correlations are always removed before physically meaningful vectors. For systems with loops, such as the case of this thesis, finding such optimal form — where internal correlations and truncation are optimally dealt with at the same time — is still an active field of research.

Differently from what we have found in the literature, we focus solely on the internal correlation — finding and removing them — as a pre-optimization for the truncation, instead of finding a canonical form [54, 55].

8.1 Internal Correlations

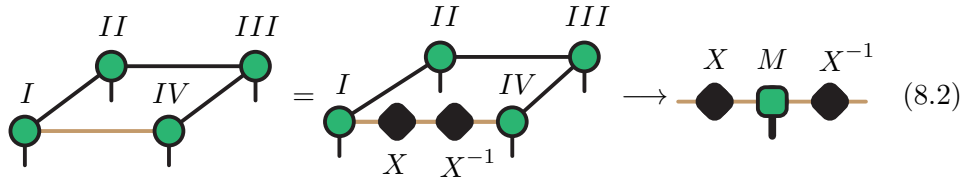
We will present the concept of internal correlations with an example where some component of a link will be shown to have zero impact on the result of the truncation.

Consider the whole contraction of a TN around a certain link, *without* contraction the link. That results in a single three-leg tensor, with two virtual links contracted to each other — the link under study — and a (reshaped) physical leg containing all physical legs of the TN. This is shown in the diagrammatic Eq. (8.1).



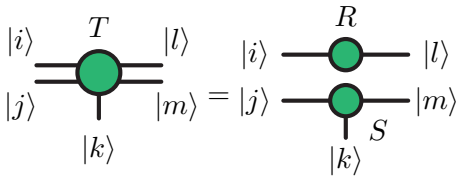
(8.1)

When the gauge freedom is utilized and a resolution of the identity $XX^{-1} = \mathcal{I}$ for invertible X is introduced, the same contraction would look instead like Eq. (8.2)



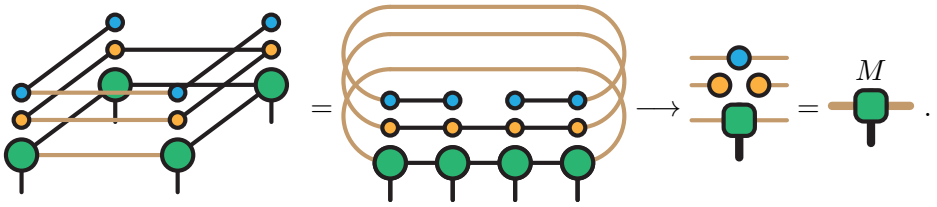
(8.2)

Let's review how a tensor product would look like in the TN diagrammatic notation.



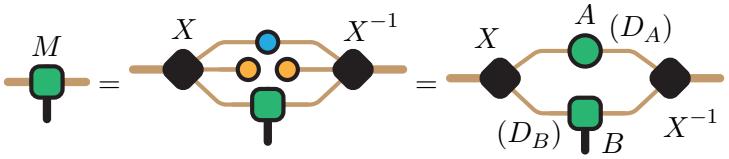
$$\longleftrightarrow \quad T_{ijklm} = R_{il} \cdot S_{jkm} \tag{8.3}$$

Consider now the alternative possibility of the TN in Eq. (8.4). Each tensor at each site of this TN is composed of the tensor product of three other tensors, depicted in Eq. (8.3). The tensors *disconnected* from the physical leg can also form loops or single lines, as shown in Eq. (8.4) with orange and blue tensors, respectively. Consider its contraction around the same link, analogous as before, resulting in M, where all virtual bonds have been reshaped on the last equality,



(8.4)

And similar to Eq. (8.2), consider the case where the gauge is changed. In general, such gauge change reshapes the three vector spaces — three legs — into a single one,



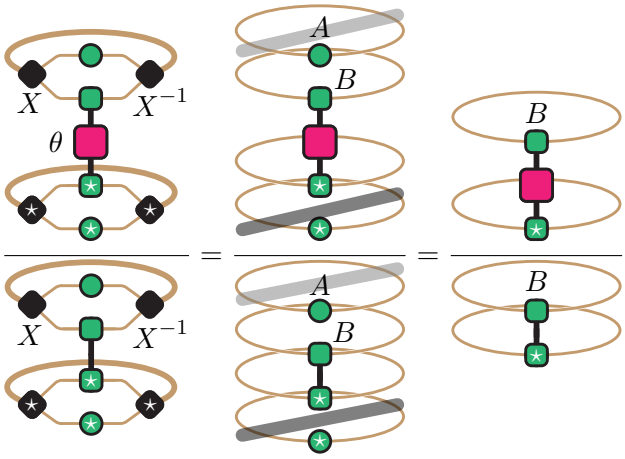
(8.5)

All tensors disconnected from the physical legs — orange and blue tensors in Eq. (8.4) — have been brought together into tensor A. Let's denote the virtual bond's dimension of A and B, D_A and D_B, respectively.

Let's consider a physically meaningful contraction, e.g. an expectation value,

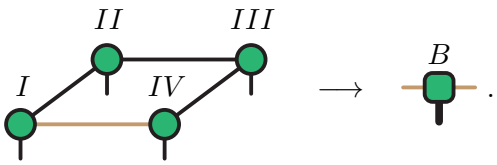
E_θ, of some general operator[1] θ for some state $|\Psi\rangle$,

$$E_\theta = \frac{\langle\Psi|\theta|\Psi\rangle}{\langle\Psi|\Psi\rangle} =$$



, (8.6)

where the star, $\star$, inside the tensors denotes complex conjugation. Notice how the factors related to tensor A all cancel each other. The result is the same one as if we had considered a TN with only B,

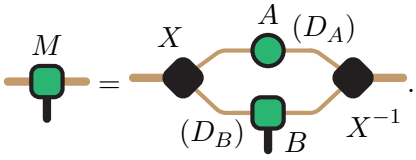


. (8.7)

In this TN, the considered link only has dimension D_B, while in the total TN of Eq. (8.4) the link had $D_A \cdot D_B$. There are $(D_A \cdot D_B - D_B)$ *less* vectors in the link from Eq. (8.7), while the result is *exactly* the same. All these vectors are the ones related to tensor A. If we would discard them — discard the whole tensor A — the contraction result would not be affected at all. Precisely these vectors associated with tensors *disconnected* from all physical legs are what we call *internal correlations*.

8.2 Finding Internal Correlations

Our goal is the following. Given a TN which contracts around a certain link to give tensor M, find the invertible matrix X such that



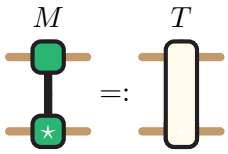
. (8.8)

[1] Even if it is acting on the single physical leg of B, recall that leg is the result of reshaping all physical legs into one. Therefore, the operator θ is completely general, including non-local operators.

For nomenclature purposes: we say that such a matrix X *separates* M into A and B. Moreover, in general, such an X may not exist. When there exists one, we say M is *separable*.

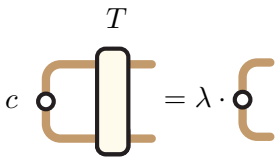
8.2.1 Rephrase the Problem

Motivated by strategies used by G. Evenbly [54], we rephrase the problem in terms of transfer matrices and dominant eigenvectors. Consider the transfer matrix, T,



, (8.9)

whose dominant eigenvector we call c and fulfils

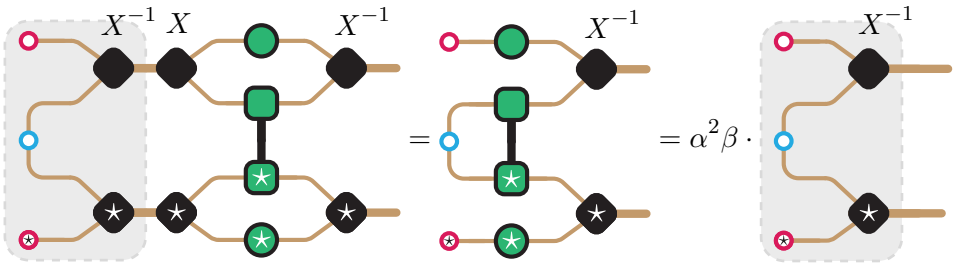


. (8.10)

Discussions in this Section will assume we are working with the left dominant eigenvector, as shown in Eq. (8.10), but results are analogous for the right dominant eigenvector. If M is separable using X into A and B, then the separated tensors' (left) dominant eigenvectors are

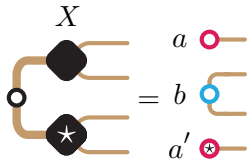
. (8.11)

With them, one can build the vector highlighted in Eq. (8.12) and see that it is also an eigenvector of T.



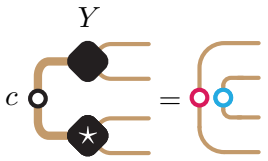
(8.12)

Then, one can say[2]

M separable with $X \implies$ 

. (8.13)

The reverse statement *might* not be true, but despite that, we shift the aim of our problem into the following: finding a Y such that[3]



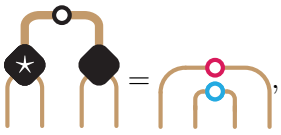
. (8.14)

If we find Y, which separates the eigenvector, it will be a good candidate for X, which separates the contracted TN, M, and therefore would give us direct access to the internal correlations, tensor A.

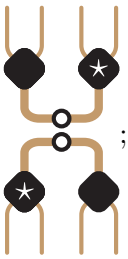
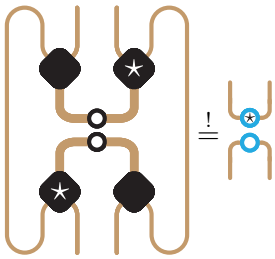
8.3 Proposed Solutions

8.3.1 Disentangle Problem

We could view this problem as the separated eigenvector representing a pure state of 4 sites. For familiarity with the vertical notation when dealing with physical sites, we could write it in the form

$|\Psi\rangle =$  , (8.15)

and ask which Y is the one that disentangles the two subsystems (inner sites versus outer sites). This is equivalent to asking that the state of one of the subsystems, P, (after tracing out the other) is a pure state, as shown in Eq. (8.16).

$\rho_Y =$  ; $\rho_P = \mathrm{Tr}_O[\rho_Y] =$ 

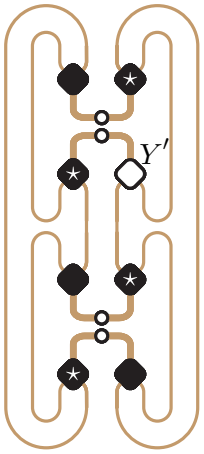
. (8.16)

[2]When T is degenerated, the vector obtained in Eq. (8.10) might not fulfil Eq. (8.13), but a linear combination of the degenerated dominant eigenvectors would fulfil it.

[3]We relax the problem to finding a general eigenvector of $AA^\dagger$, instead of the individual eigenstates of Eq. (8.13). That can be checked later.

For normalized states ρ_P, if and only if ρ_P is pure then $\mathrm{Tr}[\rho_P \cdot \rho_P]$ is maximized. Then, finding Y can be seen as an optimization problem, depicted in Eq. (8.17), and solved iteratively by replacing one Y at a time for an improved Y' that maximizes the trace.

$$\mathrm{Tr}[\rho_P \cdot \rho_P] = \quad = \mathrm{Tr}[Y' \cdot E] \tag{8.17}$$



If Y' is unitary[4]

$$\mathrm{Tr}[|Y'\cdot E|] \leq \mathrm{Tr}[|E|] = \sum_n s_n \implies \max\left(\mathrm{Tr}[Y'\cdot E\right) = \sum_n s_n, \tag{8.18}$$

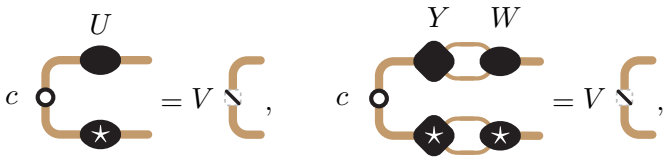
where s_n are the singular values of the environment E. The inequality can be saturated, and therefore the maximum reached, with

$$E = U \cdot S \cdot V, \quad Y' := V^\dagger U^\dagger \implies \mathrm{Tr}[Y'\cdot E] = \mathrm{Tr}[V^\dagger U^\dagger U S V] = \mathrm{Tr}[S] = \sum_n s_n. \tag{8.19}$$

The iterative algorithm would work by doing one of these optimization steps for one Y, then replacing all four Y's for the improved one — altering the environment E —, and iterating until convergence is reached. Such convergence could be measured by the purity of the subsystem ρ_P.

8.3.2 Find Y Manually

Consider the dominant eigenvector, which is in fact a matrix, and consider its eigenvalues V. These eigenvalues do not change if Y is applied, i.e.

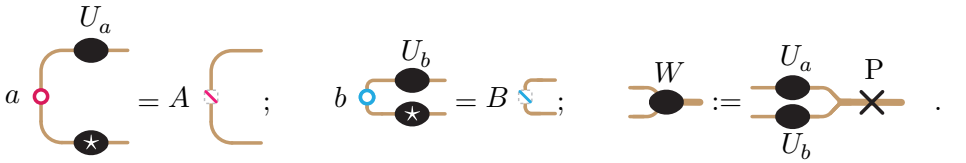


(8.20)

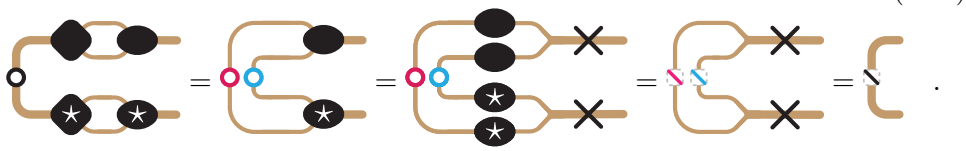
[4]The case where Y is a general invertible matrix has not been considered so far. This restricts to change the gauge only with unitary matrices, i.e. changing the basis of the links.

where here U and W are the change of basis into the diagonal form. These results have not been proven in this work but confirmed positively with several randomised tests. In the worst-case scenario, the result certainly holds if Y is unitary.

When Y is the one that separates the eigenvector c into the split vectors a and b — see Eq. (8.14)—, we can manually build the unitary W using the unitaries that would diagonalise the split vectors, U_a and U_b respectively. The designed W is given in Eq. (8.21) and it is shown to successfully produce the eigenvalues V in Eq. (8.22),



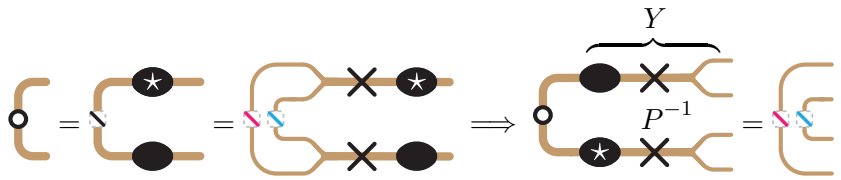
(8.21)



(8.22)

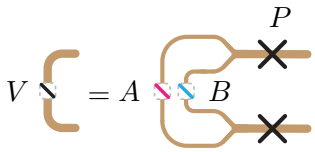
where fusing the legs is done with a basic reshape — like the one implemented by Matlab's reshape function — and P is a permutation on one leg. This permutation accounts for the more or less arbitrary ordering of the eigenvalues obtained numerically — usually sorted ascendant-wise —. When the separated eigenvalues are known, P is the permutation that sorts them, but these separated eigenvalues are precisely what we do not know.

Using this reshape, permutation and change of basis U, one can build Y in the following way:



(8.23)

Moreover, these relations can be written in terms of the eigenvalues as vectors V, A and B,



$$\iff \quad V = P(A \otimes B) \quad . \tag{8.24}$$

Therefore, the only missing object is the permutation P. If we have it, we can build Y as shown in Eq. (8.23), and consequently have a good candidate for separating the tensor M as in Eq. (8.8), which would solve our issues with internal correlations.

The problem as stated in Eq. (8.24) is formulated independently of the physics. It is an isolated problem, which we will now try to solve. The solution we provide is a non-iterative solution and therefore does not depend on the convergence of an algorithm, which can cause problems. Moreover, it has not been found in the literature and its solution is another one of the independent results of this work.

8.4 Find Permutation

Let's formulate the problem. For some dimensions D_a, D_b and $D_v = D_a \cdot D_b$, and vector $V = [v_1, \dots, v_{D_v}]$, find the permutation P such that

$$P(V) = A \otimes B = [a_1 b_1, a_1 b_2, \dots, a_{D_a} b_{D_b}], \tag{8.25}$$

for some vectors $A = [a_1, \dots, a_{D_a}]$ and $B = [b_1, \dots, d_{D_b}]$. The permutation $P = [p_1, \dots, p_{D_v}]$ acts on V, denoted with $P(V)$, as follows[5]

$$P(V) = [v_{p_1}, v_{p_2}, \dots, v_{p_{D_v}}]. \tag{8.26}$$

Notice this permutation P is the inverse of the P appearing in the previous section. Before, P permuted the assumed A and B, and here it permutes V instead.

We will first solve the problem under assumptions that avoid certain extreme cases that would impact our strategy, and treat those extreme cases separately when possible.

We start assuming that the given V is exactly separable. This is, there exists the permutation P that we are looking for which solves Eq. (8.25). In other words, V has components, for instance, like these

$$V = [a_3 b_1, a_2 b_1, a_2 b_3, \dots]. \tag{8.27}$$

Cases where we also allow for approximate results are compatible with the solution by relaxing some of the conditions that will be asked.

The second working assumption can be phrased as assuming that *numbers do not play against us*. The meaning and need of this will be seen in more detail further ahead. In cases with randomness in the numbers, this assumption is valid, since only very specific values are the ones that would break the solution and the probability of randomly hitting them is vanishing.

[5] In Matlab, the syntax is instead V(P).

8.4.1 Strategy

The solution will be based on the so-called *quotients matrix*. It is a square $D_v \times D_v$ matrix, whose i-th row is the horizontal vector V/v_i, as seen in Eq. (8.28). From its definition, one can see that in the diagonal there are always ones and that elements from the upper triangle matrix, are the inverses of the lower diagonal matrix, mirrored around the diagonal.

$$V = [v_1, v_2, \ldots, v_{D_v}] \quad \Longleftrightarrow \quad M = \begin{pmatrix} v_1/v_1 & v_1/v_2 & \ldots & v_1/v_{D_v} \\ v_2/v_1 & v_2/v_2 & \ldots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ v_1/v_{D_v} & \ldots & \ldots & v_{D_v}/v_{D_v} \end{pmatrix} \tag{8.28}$$

As practical examples, consider the case with $D_a = 2, D_b = 3$ and two examples. The first, vector V_{corr} which does not need any permutation, i.e. $V_{\text{corr}} := A \otimes B$. The second example, a general permuted vector $V_{\text{inc}} := P^{-1}(A \otimes B)$, $P = [3, 1, 4, 5, 2, 6]$. Let their quotient matrices be called *correct* and *incorrect*[6] quotient matrices, respectively.

$$V = [a_1b_1, a_1b_2, a_1b_3, a_2b_1, a_2b_2, a_2b_3]$$

$$M_{cor} = \begin{pmatrix} 1 & b_2/b_1 & b_3/b_1 & a_2/a_1 & a_2b_2/a_1b_1 & a_2b_3/a_1b_1 \\ b_1/b_2 & 1 & b_3/b_2 & a_2b_1/a_1b_2 & a_2/a_1 & a_2b_3/a_1b_2 \\ b_1/b_3 & b_2/b_3 & 1 & a_2b_1/a_1b_3 & a_2b_2/a_1b_3 & a_2/a_1 \\ a_1/a_2 & a_1b_2/a_2b_1 & a_1b_3/a_2b_1 & 1 & b_2/b_1 & b_3/b_1 \\ a_1b_1/a_2b_2 & a_1/a_2 & a_1b_3/a_2b_2 & b_1/b_2 & 1 & b_3/b_2 \\ a_1b_1/a_2b_3 & a_1b_2/a_2b_3 & a_1/a_2 & b_1/b_3 & b_2/b_3 & 1 \end{pmatrix} \tag{8.29}$$

$$V = [a_1b_3, a_1b_1, a_2b_1, a_2b_2, a_1b_2, a_2b_3]$$

$$M_{\text{inc}} = \begin{pmatrix} 1 & b_2/b_1 & a_2/a_1 & a_2b_3/a_1b_1 & b_3/b_1 & a_2b_2/a_1b_1 \\ b_1/b_2 & 1 & a_2b_1/a_1b_2 & a_2b_3/a_1b_2 & b_3/b_2 & a_2/a_1 \\ a_1/a_2 & a_1b_2/a_2b_1 & 1 & b_3/b_1 & a_1b_3/a_2b_1 & b_2/b_1 \\ a_1b_1/a_2b_3 & a_1b_2/a_2b_3 & b_1/b_3 & 1 & a_1/a_2 & b_2/b_3 \\ b_1/b_3 & b_2/b_3 & a_2b_1/a_1b_3 & a_2/a_1 & 1 & a_2b_2/a_1b_3 \\ a_1b_1/a_2b_2 & a_1/a_2 & b_1/b_2 & b_3/b_2 & a_1b_3/a_2b_2 & 1 \end{pmatrix} \tag{8.30}$$

[6] Any quotient matrix which is not the correct one is called incorrect.

Notice the structure of M_{corr}. It has some repeated values (a_2/a_1, b_2/b_1, ...) and at very specific positions. In the matrices, the repeated terms which include A and B elements have been highlighted in red and blue, respectively. In contrast, M_{inc} has the *same* repeated values but in *incorrect* positions.

The existence of repeated values is a feature of separability of the vector, i.e. of the tensor product structure, $A \otimes B$. Quotient matrices of general vectors do not have these repeated values. Therefore, we aim to use these values to argue about the separability of the considered vector and to find P.

However, relying too much on the existence of repeated values can be problematic. There can be specific cases where repeated values and their positions are not given by the tensor product structure but by finely tuned values of V's components. This would lead to the existence of repeated values even when V is not separable. Additionally, it could happen that the separated vectors themselves, A or B, have repeated values, giving wrong positions of the repeated values. All these cases are what we considered as extreme cases, and assumed that would not happen in the first solution of the problem, and in practice would be treated separately.

When conditions of the problem are favourable, one can find the permutation P in the following way. Permute M_{inc} such that the repeated values are in the correct positions, using M_{corr} as reference. Then $P(V)$ will create a *correct* quotients matrix, and that implies that $P(V) = A \otimes B$, solving the problem. The concrete steps are introduced in the following section.

8.4.2 Steps

Consider a given $V = [v_1, \ldots, v_{D_v}]$. In order to separate it into A and B vectors with dimensions D_a and D_b respectively, one should follow these steps.

1. Create quotients matrix M_{inc}, as described in Eq. (8.28).
2. Find elements of the first row which are repeated[7].
3. Classify them depending if they come from A (a_i/a_j) or B (b_i/b_j), denoted as $A \to a_i/a_j$ or $B \to b_i/b_j$, respectively. Do it by:
   - assuming first repeated element, x, to come from A,
   - and for the rest of the repeated elements, such as y, if $x \cdot y \in M_{\text{inc}} \implies B \to y$, otherwise $A \to y$[8].

[7] When allowing for approximate results, $P(V) \simeq A \otimes B$, we can set a tolerance for considering two elements as equal. It is also helpful for avoiding noise errors.

[8] If $A \to y$ then $xy = a_i a_k / a_j a_l$, which is not in M_{inc}, see Eq. (8.29). This is another weak point of the solution since elements of V can be such that $a_i a_k / a_j a_l$ appears in M_{inc}. This would have other consequences on M_{inc} that could be noticed and used as warning flags. Moreover, to reinforce this classification, one could do it over and over using different initial reference values, not only using the first repeated element.

4. Consider ordering the first row such that the repeated elements are in the *correct* positions, given by M_{cor} of Eq. (8.29). Let the permutation used to order the first row be P_1. Permute the initial M_{inc} with it, $P_1(M_{\text{inc}})$.
5. Repeat steps 2 to 4 for rows 2 to $D_x = \min(D_a, D_b)$[9]. If V is separable and not an extreme case, ordering a new row will not disorder any repeated elements from the previous ones.
6. The final P is the composition of all row-ordering permutations, P_i, $i \in \{1, \dots, D_x\}$, as it brings M_{inc} into $M_{\text{corr}} = P(M_{\text{inc}})$.

With this method, and implementing all extreme cases, one can find permutation P to solve the problem in Eq. (8.24), which later may allow to identify the internal correlations.

Due to time limitations — this problem was raised towards the end of the thesis and is the last problem we were working with — not all extreme cases have been implemented. Allowing for approximate results has been successfully implemented, as well as cases with zeros in V. Unfortunately, cases with repeated values have not been faced. Lastly, the implementation of the currently working solutions into the TN has not been made, and therefore its effect on the physical results has not been tested.

[9] Keeping the repeated elements' positions of the first D_x rows fixes the positions of all D_v components. Any new permutation would disorder them. One can check this statement by looking at Eq. (8.29). Therefore, if the problem is well conditioned and a solution P exists, it will be already found at row D_x — up to internal permutations of the separated vectors A and B—.

Chapter 9

Summary and Outlook

Summary. In Ch. 4, we presented the *ansatz* of Ref. [1], which is expressed in the TN formalism (Ch. 3) and only represents states from the physical corner of the Hilbert space (Ch. 2). Later, in Ch. 5, we showed how the *ansatz* is not straightforwardly compatible with generic symmetric libraries. In that same chapter, the first result of the thesis is presented, which consists of expressing the *ansatz* with a new symmetry such that the previous incompatibility is overcome. Moreover, we demonstrated in Ch. 6 that, not only does the *ansatz* represent physical states, but *all* physical states can be represented by the *ansatz*.

In order to test the new *ansatz*, we searched for the ground state of the model Eq. (2.3) in an infinite lattice, with an imaginary evolution, Ch. 7. We encountered some limitations —instabilities and unexpected energy behaviours of the algorithm— and studied them on a finite lattice. That led us to consider internal correlations as the main error source in the algorithm. Then, in Ch. 8, we reformulated the problem for a general TN with a loop and proposed a solution to it. The solution is based on the permuted tensor product of two vectors, $V = P(A \otimes B)$, and finding P given V. This is the last independent result of the thesis.

Outlook. In this work, we considered a simple $\mathbb{Z}_2$ lattice gauge theory with only gauge fields on the links: no (matter) systems on the sites. The present work is expected to apply to more complex systems with minimal modifications.

Matter on the links would result in vertex tensors with 5 legs. Then, the background charge g_i should be replaced with the charge of the physical leg's component. That would produce still $\mathbb{Z}_2 \otimes \mathbb{Z}_2$ vertex tensors, while the copy tensors would remain untouched., i.e. an overall $\mathbb{Z}_2 \otimes \mathbb{Z}_2$ *ansatz* still.

Moreover, other systems on the sites could be considered, which could imply other local symmetries in the Hamiltonian, like $Z(N)$, $U(1)$ or $SU(2)$. In that case, the generalisation to these new symmetries of the *doubling* strategy should be checked. Furthermore, the proof of Ch. 6 could be extended to these symmetries.

The proposed solution to internal correlations should be studied further and tested in the $\mathbb{Z}_2$ model. If it is robustly successful, that would imply pleasant progress in TN with loops and progress in our simulations.

With such progress, a more extensive search of the ground state (for several model parameters) could be performed. As a result, the phase diagram of this model (and possibly others) could be drawn using imaginary time evolution. That would allow us to better check the algorithm with known physics, and use it to pursue new physics.

Appendix A

Advanced Concepts in TN

On Ch. 3 we introduced the definition of a tensor, Eq. (3.2), how to contract them, Eq. (3.6), and their diagrammatic notations, Eqs. (3.4) and (3.7). Later on in Sec. 3.3, we explained how to find the ground state using these concepts, that is: applying the gates in Eq. (3.21) to the symmetric *ansatz* of Ch. 5 and truncating. Additionally, the environment approximation is introduces in Eq. (3.27) which is used for calculating expectation values, such as the energy. The exact details of the truncation and the energy calculations are shown in Sec. A.1 and Sec. A.2, respectively.

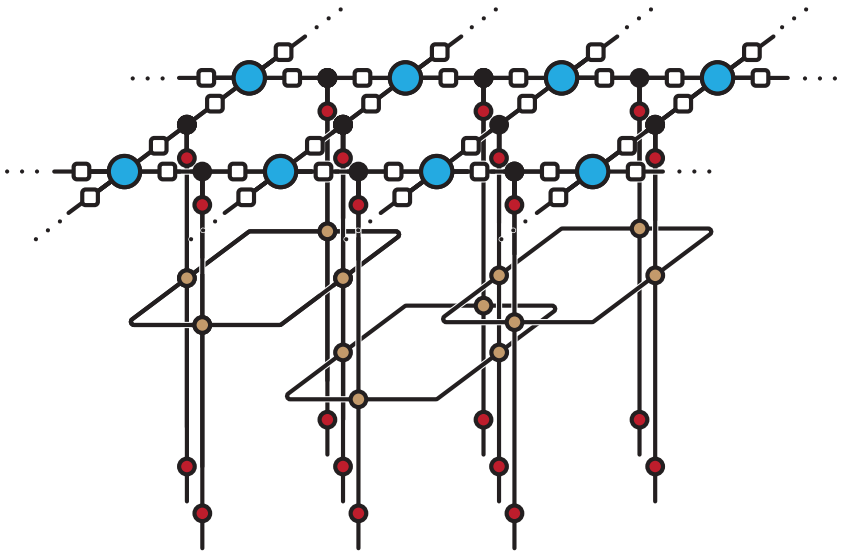
Moreover, in Ch. 8 we use the so-called *gauge freedom* on a TN, which allows to "change the gauge" to our convenience[1].This is explained in Sec. A.3.

[1]Not to be confused with the gauge freedom on high energy physics. Even if they are a similar kind of freedom, in the TN context it simply means introducing an identity resolution in some contraction.

A.1 Applying the Plaquette Term

Evolving for a time step β means: first computing the contraction introduced in Eq. (3.24) which we present again here with the Λ matrices:

$|\Psi(t+\beta\rangle =$



(A.1)

and then recovering the *ansatz*'s structure while keeping the bond's dimensions under some threshold $D_{\max}$. Applying the local gates —contracting one copy tensor with one local gate through the d-dimensional physical leg— does not grow the bond. Thus, applying the local terms involves a single contraction per link — with cost $\mathcal{O}(d^2 D_{\max}^2)$.

Here, we focus on applying the (cMPO) plaquette terms of Eq. (3.23). As seen in Eq. (A.1), we must apply one plaquette operator for each plaquette on the lattice. Since we work on an infinite lattice tiled by a unit cell, see Fig. (5.9), we only apply the operator to the unit cell. We do so one plaquette at a time[2], as explained in the following

We start the process by extracting a plaquette from the TN, as in Eq. (3.27). Then, these steps are followed:

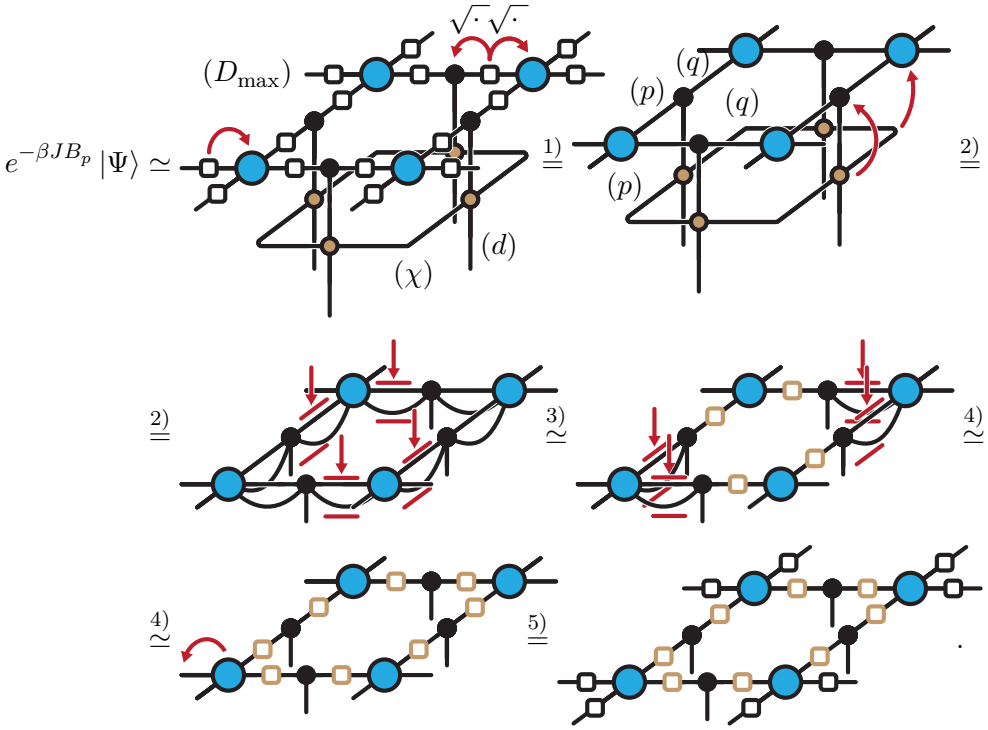
1) the environment —including the inner Λ matrices— is absorbed; cost $\mathcal{O}(D_{\max}^5)$,
2) the MPO's tensors and χ-dimensional auxiliary legs are absorbed into the copy and vertex tensors, respectively; cost $\mathcal{O}(\chi^2 D_{\max}^4)$,
3) the diagonal vertices are truncated —details are given in Eq. (A.3)— producing new Λ matrices; cost $\mathcal{O}(\chi^3 D_{\max}^5)$—,

[2] In practice, we iterated through the unit cell following a checkerboard pattern: first "one color" and then the opposite one. We do so to avoid the overlapping of plaquettes on a first run. The effect of this method has not been studied in detail.

4) the off-diagonal vertices are similarly truncated; cost $\mathcal{O}(\chi^3 D_{\max}^5)$; and finally,

5) the environment is detached; cost $\mathcal{O}(D_{\max}^5)$.

The final cost is: $\mathcal{O}(\chi^3 D_{\max}^5)$. See Eq. (A.2).



(A.2)

On the absorption of the environment in 1), the inner Λ's are absorbed to the neighbouring tensors, $\sqrt{\Lambda}$ to each side. And on the final step 5), detaching the environment means that the original environment Λ_0 —depicted with the original black strokes— is recovered by introducing $\mathcal{I} = \Lambda_0 \cdot \Lambda_0^{-1}$ in the bond and absorbing the inverse in the vertex tensor. Finally, as seen in Ch. 6, the type of the MPO's auxiliary leg has to match the virtual leg's type in order to recover the $\mathbb{Z}_2 \otimes \mathbb{Z}_2$ symmetric structure, as depicted in Eq. (6.12) with (p) and (q).

The truncation on the vertices is done by:

1) *cutting*[3] the vertex into big (4+1 legs) and small (2+1 legs) contracted tensors; cost $\mathcal{O}(\chi^3 D_{\max}^5)$,

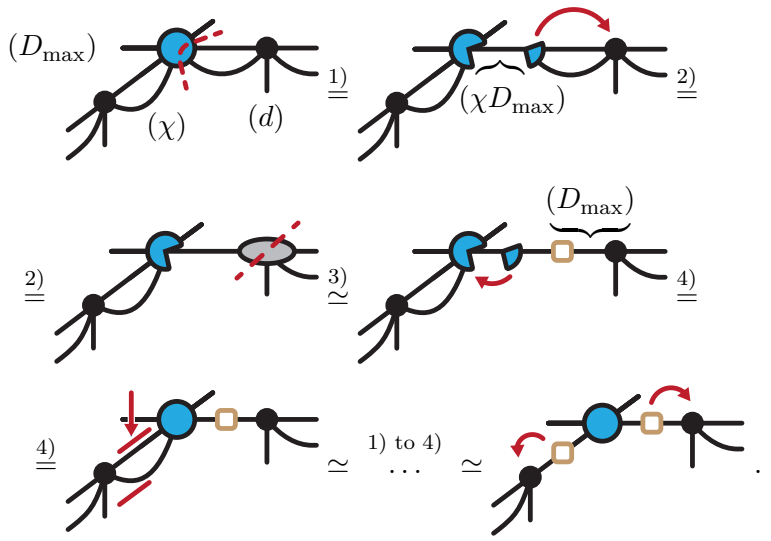
2) absorbing the small tensor towards the nearest copy tensor; cost $\mathcal{O}(\chi^3 d D_{\max}^3)$,

[3] Cutting here means doing a non.truncated singular value decomposition and absorbing the singular values to the smaller tensor, resulting in two final tensors.

3) performing a truncated singular value decomposition; cost $\mathcal{O}(\chi^2 d D_{\max}^3)$; and finally,

4) absorbing back the small tensor; cost $\mathcal{O}(\chi D_{\max}^5)$.

This is done twice, once for each neighbouring copy tensor, as shown in Eq. (A.3) with dots. The total cost amounts to $\mathcal{O}(\chi^3 D_{\max}^5)$.



(A.3)

In the off-diagonal case, the only difference is that the auxiliary legs of the copy tensors are no longer there. Additionally, to account for the environment of the off-diagonal corners, before their truncation happens, the copy tensors absorb the new Λ's, as depicted at the end of Eq. (A.3) and with a cost of $\mathcal{O}(\chi d D_{\max}^3)$, which later are detached back.

With this, the plaquette is successfully applied and the bonds truncated down to $D_{\max}$.

A.2 Energy Calculation

Considering the Hamiltonian $\mathcal{H}$ from Eq. (2.3), the energy of a state $|\Psi\rangle$ is

$$\begin{aligned}\frac{1}{\mathcal{N}}\langle\Psi|\mathcal{H}|Psi\rangle &= \frac{1}{\mathcal{N}}\langle\Psi| - J\sum_p B_p + h\sum_l \sigma_x^l|\Psi\rangle = \\ &= \frac{1}{\mathcal{N}}\langle\Psi| - J\sum_p B_p + \frac{h}{2}\sum_p\sum_{l\in p}\sigma_x^l|\Psi\rangle = \\ &= \sum_p \frac{1}{\mathcal{N}}\langle\Psi| - JB_p + \frac{h}{2}\sum_{l\in p}\sigma_x^l|\Psi\rangle \end{aligned} \tag{A.4}$$

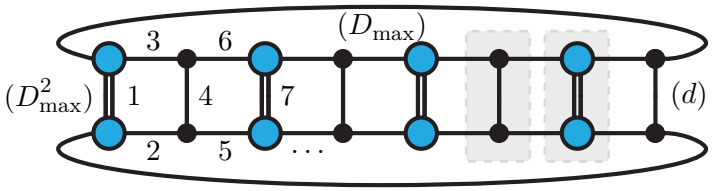
where $\mathcal{N} = \langle\Psi|\Psi\rangle$ is the norm of the state.

Since we work with an infinite lattice, we compute instead the energy of a unit cell, where the summation runs over a finite amount of plaquettes. Finally, we compute the energy per site E on a $L_x \times L_y$ unit cell (u.c.), which includes a normalisation factor of $2 \cdot L_x L_y$ — because there are two sites in a 1×1 unit cell, Fig. 5.8a—

$$E = \frac{1}{2L_xL_y} \sum_p \in \text{u.c.} \; \frac{1}{\mathcal{N}} \langle \Psi | - JB_p + \frac{h}{2} \sum_{l \in p} \sigma_x^l | \Psi \rangle \,. \tag{A.5}$$

For computing Eq. (A.5) we only need $\mathcal{N} = \langle \Psi | \Psi \rangle$, $\langle \Psi | B_p | \Psi \rangle$ and $\langle \Psi | \sigma_x^l | \Psi \rangle$. They are all very similar contractions once the operators are absorbed by the state's tensors. The local operator σ_x^l is directly absorbed in the copy tensors. The plaquette B_p is absorbed as well but the auxiliary legs have to be treated, similar to Eq. (A.2). The remaining contraction is simply the overlap of $|\Psi\rangle$ and the absorbed state $|\tilde{\Psi}\rangle$, $\langle \Psi | \tilde{\Psi} \rangle$. Again, a whole state is approximated by extracting a plaquette from the TN together with the Λ matrices, as in Eq. (3.27).

Once the plaquette is extracted, the Λ's are all absorbed, similar to Eq. (A.2), and the remaining contraction can be seen as



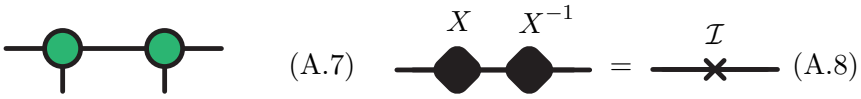
(A.6)

Contracting this TN can be efficiently done with a cost of $\mathcal{O}(D^7_{\max})$ if contracting through the bonds in the indicated order, $1, 2, 3 \ldots$ in Eq. (A.6). Due to lack of time, only the more straightforward way of contracting the TN was implemented, which involved the transfer matrices highlighted in Eq. (A.6). This second way of contracting gives instead a cost of $\mathcal{O}(D^8_{\max}$. The cost difference can be easily fixed and was not a problem in this work because time limitations where never an issue.

A.3 Gauge Freedom in TN

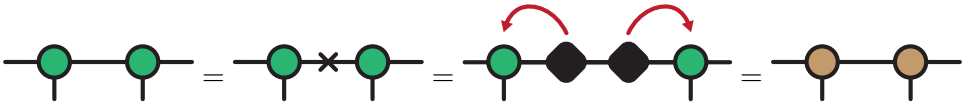
We explain the concept of gauge freedom with an example. Consider the TN of Eq. (A.7) for some general tensors, and the identity[4] resolution $X \cdot X^{-1} = \mathcal{I}$ of Eq. (A.8), for some invertible matrix X.

[4] Usually, the identity is depicted diagrammatically with a single line, as nothing happens across an identity. We explicitly draw it here for illustrative purposes.



(A.7) (A.8)

The contracted links in Eq. (A.7) can be seen as an $\mathcal{I}$. Introducing an identity in any leg does not change the result. We can then use Eq. (A.8) to introduce the matrices X and X^{-1} and absorb them into the TN's tensors. This creates a TN with new tensor which is exactly equivalent to the original one. See Eq. (A.9)



(A.9)

Therefore, one can freely introduce invertible matrices in the bonds without changing the state represented by the TN. This is the *gauge freedom* in the context of TN. It is used to set up TN for better truncation, or in general, to impose a certain condition on the tensors of the TN. In Ch. (8) we look for an invertible matrix X —a gauge change— that makes a tensor have a tensor product structure under a basic isometry (reshape).

Appendix **B**

Derivations of the Proof

B.1 First Derivation

Show that

$$[\theta, A_i] = 0 \ \forall i \iff \theta \text{ is gauge preserving,}$$

where gauge preserving means

$$|\eta\rangle \in K \iff \theta\,|\eta\rangle \in K.$$

B.1.1 Assume $[\theta, A_i] = 0 \ \forall i$

Again, we have to show the left and right implications independently using the assumption.

Right implication, $\Longrightarrow$

Given that $|\eta\rangle \in K$,

$$A_i\,(\theta\,|\eta\rangle) \overset{(1)}{=} \theta\,(A_i\,|\eta\rangle) = \theta\,((-1)^{q_i}\,|\eta\rangle) = (-1)^{q_i}\,(\theta\,|\eta\rangle) \ \forall i \implies \theta\,|\eta\rangle \in K,$$

where in (1) we have used the commutator assumption.

Left implication, $\Longleftarrow$

By negating the implication, we show instead

$$|\eta\rangle \notin K \implies \theta\,|\eta\rangle \notin \ K.$$

The case where $|\eta\rangle \in K' \neq K$, we have showed already that $\theta\,|\eta\rangle \in K' \neq K$, which agrees with the above statement. We have to show the case where $|\eta\rangle$ doesn't belong to any gauge sector. In other words, when $|\eta\rangle$ is a sum of vectors from different gauge sectors. For example, in the case of two such

sectors, $|\eta\rangle \in K \oplus K'$, for $K \neq K'$. In this case, assuming $|\alpha\rangle \in K, |\beta\rangle \in K'$, we have to show

$$A_i\theta\,|\eta\rangle \overset{(1)}{=} \theta A_i(|\alpha\rangle+|\beta\rangle) = \theta\left((-1)^{q_i}\,|\alpha\rangle + (-1)^{q_i'}\,|\beta\rangle\right) = (-1)^{q_i}\theta\,|\alpha\rangle+(-1)^{q_i'}\theta\,|\beta\rangle \;\forall i$$

where in (1) we used the commutator assumption. The only way for $\theta\,|\eta\rangle$ to belong to K is if $|\beta\rangle \in K$, which we assumed not to be the case. This finishes the proof for both ways of the gauge preserving definition. Thus proves the implication where we assume the commutator result.

B.1.2 Assume θ is gauge preserving

We have to show that $\theta A_i = A_i\theta\ \forall i$. For states on a gauge sector $|\eta\rangle \in K$,

$$\theta A_i\,|\eta\rangle = (-1)^{q_i}\underbrace{\theta\,|\eta\rangle}_{\in K} = A_i\theta\,|\eta\rangle\,,$$

where the under brace result comes from θ being gauge preserving. On the other hand, when the state is not in any gauge state, $|\eta\rangle \in K \oplus K'$, assuming $|\alpha\rangle \in K, |\beta\rangle \in K'$,

$$\theta A_i\,|\eta\rangle = \theta A_i(|\alpha\rangle+|\beta\rangle) = (-1)^{q_i}\theta\,|\alpha\rangle+(-1)^{q_i'}\theta\,|\beta\rangle \overset{(1)}{=} A_i\theta\,|\alpha\rangle+A_i\theta\,|\beta\rangle = A_i\theta\,|\eta\rangle\,.$$

In (1) we have used again the fact that θ is gauge preserving.

With this we finish the whole proof. The last expressions prove that a gauge preserving operator commutes with $A_i\ \forall i$. And with it, we have both ways of the original claim.

B.2 Second Derivation

Let a general operator be written as $\theta = \sum_m c_m \cdot a_m,\ c_m \in \mathbb{C}$. Show that the gauge preserving condition implies,

$$[\theta, A_i] = 0\ \forall i \iff c_m = 0,\ \forall m \mid \exists i, [a_m, A_i] \neq 0.$$

Left implication, $\Longleftarrow$

In this case it is straight forward to see that

$$[\theta, A_i] = \sum_m c_m \cdot [a_m, A_i] = \sum 0 \cdot [a_m, A_i] + \sum c_m \cdot 0 = 0.$$

Right implication, $\Longrightarrow$

First of all,

$$[\theta, A_i] = 0 \implies \sum_m c_m \cdot [a_m, A_i] = 0.$$

Consider the outcome of $[a_m, A_i]$. There are two possible outcomes. When a_m is such that the four σ_x of A_i meet an odd number of off-diagonal operators (σ_y or σ_z) the commutator is non-zero (we then say $m \in odd$). Otherwise it yields zero ($m \in even$). A couple of examples, considering the first four sites those meeting with A_i:

$$\begin{aligned} m \in even: \qquad [a_m, A_i] &= [\sigma_x \otimes \sigma_y \otimes \sigma_z \otimes \mathcal{I} \otimes ..., \sigma_x \otimes \sigma_x \otimes \sigma_x \otimes \sigma_x \otimes ...] = \\ &= \mathcal{I} \otimes \left\{\sigma_y\sigma_x \otimes \sigma_z\sigma_x - \sigma_x\sigma_y \otimes \sigma_x\sigma_z\right\} \otimes \sigma_x \otimes ... = \\ &= \mathcal{I} \otimes \left\{-i\sigma_z \otimes i\sigma_y - i\sigma_z \otimes (-i\sigma_y)\right\} \otimes \sigma_x \otimes ... = \\ &= 0 \end{aligned}$$

$$\begin{aligned} m \in odd: \qquad [a_m, A_i] &= [\sigma_y \otimes \sigma_y \otimes \sigma_z \otimes \mathcal{I} \otimes ..., \sigma_x \otimes \sigma_x \otimes \sigma_x \otimes \sigma_x \otimes ...] = \\ &= \left\{\sigma_y\sigma_x \otimes \sigma_y\sigma_x \otimes \sigma_z\sigma_x - \sigma_x\sigma_y \otimes \sigma_x\sigma_y \otimes \sigma_x\sigma_z\right\} \otimes \sigma_x \otimes ... = \\ &= \left\{-i\sigma_z \otimes -i\sigma_z \otimes i\sigma_y - i\sigma_z \otimes i\sigma_z \otimes (-i\sigma_y)\right\} \otimes \sigma_x \otimes ... = \\ &= -2i \cdot \sigma_z \otimes \sigma_z \otimes \sigma_y \otimes \sigma_x \otimes ... \end{aligned}$$

Notice how the different result comes from the amount of i factors that meets up, which makes a distinction between odd and even number of off-diagonal operators.

$$[a_m, A_i] =: \begin{cases} 0 & m \in even \\ \tilde{a}_m & m \in odd \end{cases} \implies \sum_m c_m \cdot [a_m, A_i] = \sum_{m \in odd} c_m \cdot \tilde{a}_m$$

Moreover, notice how the non-zero result $\tilde{a}_m$ is proportional to another basis vector, i.e. still a basis vector. And more importantly, for $m \in odd$, this map from a_m to another $\tilde{a}_m$ through the commutator is a one to one map. This means that the collection of results of $\{\tilde{a}_m\}$ doesn't have repeated vectors. Thus, such collection is still a good basis (of reduced dimension).

Finally, we can use the linear independence of vectors from a basis to write

$$\sum_{m \in odd} c_m \cdot \tilde{a}_m = 0 \iff c_m = 0, \ \forall m \in odd.$$

This concludes the proof since what we wrote in the original claim as "$\forall m \mid \exists i, [a_m, A_i] \neq 0$" is what we defined in the proof as *odd*.

Bibliography

[1] L. Tagliacozzo, A. Celi, and M. Lewenstein, "Tensor Networks for Lattice Gauge Theories with Continuous Groups", en, Phys. Rev. X **4**, 041024 (2014).

[2] A. Altland and B. D. Simons, *Condensed Matter Field Theory*, 2nd ed. (Cambridge University Press, Cambridge, 2010).

[3] J. Bardeen, L. N. Cooper, and J. R. Schrieffer, "Theory of Superconductivity", Phys. Rev. **108**, Publisher: American Physical Society, 1175 (1957).

[4] J. I. Cirac and F. Verstraete, "Renormalization and tensor product states in spin chains and lattices", en, J. Phys. A: Math. Theor. **42**, 504004 (2009).

[5] J. Eisert, M. Cramer, and M. B. Plenio, "Colloquium: Area laws for the entanglement entropy", Rev. Mod. Phys. **82**, Publisher: American Physical Society, 277 (2010).

[6] P. Calabrese and J. Cardy, "Entanglement entropy and quantum field theory", en, J. Stat. Mech. **2004**, P06002 (2004).

[7] M. M. Wolf, "Violation of the Entropic Area Law for Fermions", Phys. Rev. Lett. **96**, Publisher: American Physical Society, 010404 (2006).

[8] S. R. White, "Density-matrix algorithms for quantum renormalization groups", Phys. Rev. B **48**, Publisher: American Physical Society, 10345 (1993).

[9] I. Peschel, M. Kaulke, X. Wang, and K. Hallberg, eds., *Density-Matrix Renormalization*, en, Vol. 528, Lecture Notes in Physics (Springer Berlin Heidelberg, 1999).

[10] U. Schollwöck, "The density-matrix renormalization group", Rev. Mod. Phys. **77**, Publisher: American Physical Society, 259 (2005).

[11] J. Dukelsky, M. A. Martín-Delgado, T. Nishino, and G. Sierra, "Equivalence of the variational matrix product method and the density matrix renormalization group applied to spin chains", en, EPL **43**, Publisher: IOP Publishing, 457 (1998).

[12] S. Östlund and S. Rommer, "Thermodynamic Limit of Density Matrix Renormalization", Phys. Rev. Lett. **75**, Publisher: American Physical Society, 3537 (1995).

[13] A. Klümper, A. Schadschneider, and J. Zittartz, "Matrix Product Ground States for One-Dimensional Spin-1 Quantum Antiferromagnets", en, EPL **24**, 293 (1993).

[14] F. Verstraete and J. I. Cirac, *Renormalization algorithms for Quantum-Many Body Systems in two and higher dimensions*, arXiv:cond-mat/0407066, July 2004.

[15] F. Verstraete and J. I. Cirac, "Valence-bond states for quantum computation", Phys. Rev. A **70**, Publisher: American Physical Society, 060302 (2004).

[16] J. I. Cirac, D. Pérez-García, N. Schuch, and F. Verstraete, "Matrix product states and projected entangled pair states: Concepts, symmetries, theorems", Rev. Mod. Phys. **93**, Publisher: American Physical Society, 045003 (2021).

[17] Y.-Y. Shi, L.-M. Duan, and G. Vidal, "Classical simulation of quantum many-body systems with a tree tensor network", Phys. Rev. A **74**, Publisher: American Physical Society, 022320 (2006).

[18] G. Vidal, "Classical Simulation of Infinite-Size Quantum Lattice Systems in One Spatial Dimension", en, Phys. Rev. Lett. **98**, 070201 (2007).

[19] G. Vidal, "Class of Quantum Many-Body States That Can Be Efficiently Simulated", Phys. Rev. Lett. **101**, Publisher: American Physical Society, 110501 (2008).

[20] R. Orús and G. Vidal, "Infinite time-evolving block decimation algorithm beyond unitary evolution", Phys. Rev. B **78**, Publisher: American Physical Society, 155117 (2008).

[21] G. Vidal, "Efficient simulation of one-dimensional quantum many-body systems", Phys. Rev. Lett. **93**, arXiv:quant-ph/0310089, 040502 (2004).

[22] G. Evenbly and G. Vidal, "Tensor Network Renormalization", Phys. Rev. Lett. **115**, Publisher: American Physical Society, 180405 (2015).

[23] J. Haegeman, T. J. Osborne, and F. Verstraete, "Post-matrix product state methods: To tangent space and beyond", Phys. Rev. B **88**, Publisher: American Physical Society, 075133 (2013).

[24] L. Vanderstraeten, J. Haegeman, and F. Verstraete, "Tangent-space methods for uniform matrix product states", en, SciPost Physics Lecture Notes, 007 (2019).

[25] R. Orús, "Tensor networks for complex quantum systems", en, Nat Rev Phys **1**, Number: 9 Publisher: Nature Publishing Group, 538 (2019).

[26] M. C. Bañuls and K. Cichy, "Review on novel methods for lattice gauge theories", en, Rep. Prog. Phys. **83**, arXiv:1910.00257 [hep-lat, physics:hep-th, physics:quant-ph], 024401 (2020).

[27] M. K. Gaillard, P. D. Grannis, and F. J. Sciulli, "The standard model of particle physics", Rev. Mod. Phys. **71**, Publisher: American Physical Society, S96 (1999).

[28] K. Aoki, K. Sakakibara, I. Ichinose, and T. Matsui, "Magnetic order, Bose-Einstein condensation, and superfluidity in a bosonic $t\text{\ensuremath{-}}J$ model of ${\text{CP}}^{1}$ spinons and doped Higgs holons", Phys. Rev. B **80**, Publisher: American Physical Society, 144510 (2009).

[29] G. Baskaran and P. W. Anderson, "Gauge theory of high-temperature superconductors and strongly correlated Fermi systems", Phys. Rev. B **37**, Publisher: American Physical Society, 580 (1988).

[30] I. Ichinose and T. Matsui, "Lattice gauge theory for condensed matter physics: ferromagnetic superconductivity as its example", Mod. Phys. Lett. B **28**, Publisher: World Scientific Publishing Co., 1430012 (2014).

[31] P. A. Lee and N. Nagaosa, "Gauge theory of the normal state of high-${\mathit{T}}_{\mathit{c}}$ superconductors", Phys. Rev. B **46**, Publisher: American Physical Society, 5621 (1992).

[32] K. G. Wilson, "Confinement of quarks", Phys. Rev. D **10**, Publisher: American Physical Society, 2445 (1974).

[33] F. J. Wegner, "Duality in Generalized Ising Models and Phase Transitions without Local Order Parameters", en, Journal of Mathematical Physics **12**, 2259 (1971).

[34] J. Kogut and L. Susskind, "Hamiltonian formulation of Wilson's lattice gauge theories", en, Phys. Rev. D **11**, 395 (1975).

[35] M. Creutz, "Gauge fixing, the transfer matrix, and confinement on a lattice", Phys. Rev. D **15**, Publisher: American Physical Society, 1128 (1977).

[36] J. B. Kogut, "An introduction to lattice gauge theory and spin systems", en, Rev. Mod. Phys. **51**, 659 (1979).

[37] S. Elitzur, "Impossibility of spontaneously breaking local symmetries", Phys. Rev. D **12**, Publisher: American Physical Society, 3978 (1975).

[38] A. Y. Kitaev, "Fault-tolerant quantum computation by anyons", Annals of Physics **303**, 2 (2003).

[39] T. M. R. Byrnes, P. Sriganesh, R. J. Bursill, and C. J. Hamer, "Density matrix renormalisation group approach to the massive Schwinger model", Nuclear Physics B - Proceedings Supplements **109**, 202 (2002).

[40] M. Bañuls, K. Cichy, J. Cirac, and K. Jansen, "The mass spectrum of the Schwinger model with matrix product states", en, J. High Energ. Phys. **2013**, 158 (2013).

[41] L. Tagliacozzo and G. Vidal, "Entanglement renormalization and gauge symmetry", en, Phys. Rev. B **83**, arXiv:1007.4145 [cond-mat, physics:hep-lat, physics:hep-th, physics:quant-ph], 115127 (2011).

[42] J. Haegeman, K. Van Acoleyen, N. Schuch, J. I. Cirac, and F. Verstraete, "Gauging Quantum States: From Global to Local Symmetries in Many-Body Systems", Phys. Rev. X **5**, Publisher: American Physical Society, 011024 (2015).

[43] E. Zohar and M. Burrello, "Building projected entangled pair states with a local gauge symmetry", en, New J. Phys. **18**, Publisher: IOP Publishing, 043008 (2016).

[44] E. Zohar, M. Burrello, T. B. Wahl, and J. I. Cirac, "Fermionic projected entangled pair states and local U(1) gauge theories", Annals of Physics **363**, 385 (2015).

[45] P. Emonts, A. Kelman, U. Borla, S. Moroz, S. Gazit, and E. Zohar, "Finding the ground state of a lattice gauge theory with fermionic tensor networks: a $2+1d$ $\mathbb{Z}_2$ demonstration", Phys. Rev. D **107**, arXiv:2211.00023 [cond-mat, physics:hep-lat, physics:quant-ph], 014505 (2023).

[46] E. Zohar and J. I. Cirac, "Combining tensor networks with Monte Carlo methods for lattice gauge theories", Phys. Rev. D **97**, Publisher: American Physical Society, 034510 (2018).

[47] Y. Kuramashi and Y. Yoshimura, "Three-dimensional finite temperature Z2 gauge theory with tensor network scheme", en, J. High Energ. Phys. **2019**, 23 (2019).

[48] I. P. McCulloch, "From density-matrix renormalization group to matrix product states", en, J. Stat. Mech. **2007**, P10014 (2007).

[49] S. Singh and G. Vidal, "Tensor network states and algorithms in the presence of a global SU(2) symmetry", Phys. Rev. B **86**, Publisher: American Physical Society, 195114 (2012).

[50] A. Weichselbaum, "Non-abelian symmetries in tensor networks: A quantum symmetry space approach", Annals of Physics **327**, 2972 (2012).

[51] M. Dalmonte and S. Montangero, "Lattice gauge theories simulations in the quantum information era", en, Contemporary Physics **57**, arXiv:1602.03776 [cond-mat, physics:hep-lat, physics:quant-ph], 388 (2016).

[52] A. T. Sornborger and E. D. Stewart, "Higher-order methods for simulations on quantum computers", en, Phys. Rev. A **60**, 1956 (1999).

[53] F. Verstraete and J. I. Cirac, "Matrix product states represent ground states faithfully", Phys. Rev. B **73**, arXiv:cond-mat/0505140, 094423 (2006).

[54] G. Evenbly, "Gauge fixing, canonical forms, and optimal truncations in tensor networks with closed loops", Phys. Rev. B **98**, Publisher: American Physical Society, 085155 (2018).

[55] J. Tindall and M. Fishman, *Gauging tensor networks with belief propagation*, arXiv:2306.17837 [quant-ph], Aug. 2023.