NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF PHYSICS

AFONSO DAVID DOS REIS CALDEIRA DE CORDEIRO CAPELA

Bachelor in Engineering Physics

APPLICATION OF TENSOR NETWORKS ON TEXT UNIVERSES

MASTER IN PHYSICS ENGINEERING

NOVA University Lisbon
⟨March⟩, ⟨2022⟩

NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF PHYSICS

APPLICATION OF TENSOR NETWORKS ON TEXT UNIVERSES

AFONSO DAVID DOS REIS CALDEIRA DE CORDEIRO CAPELA

Bachelor in Engineering Physics

Advisers: João Pires da Cruz
Invited Aux. Professor, University of Lisbon
André Wemans
Assistant Professor, NOVA University Lisbon

MASTER IN PHYSICS ENGINEERING

NOVA University Lisbon
⟨March⟩, ⟨2022⟩

Application of Tensor Networks On Text Universes

To my parents,
Miguel and Carolina.

Agradecimentos

Quero começar por agradecer aos meus orientadores: Professor João Pires da Cruz e Professor André Wemans, pela sua paciência e pelo seu apoio naquele que foi o maior desafio do meu longo percurso académico.

Aos meus pais, a quem esta tese é dedicada. Esta tese é para eles, por eles e graças a eles, devido à sua inacreditável capacidade de me dar sempre o apoio necessário na altura certa. Jamais chegaria onde estou hoje senão fosse pela vossa incrível dedição e por me ensinarem tudo aquilo que sei hoje. Esta tese tem a vossa co-autoria.

Aos meus avós, António, Francisco, Judite e Maria que sempre me motivaram e acarinharam. Aos que já cá não estão resta-me apenas o conforto de saber que estariam orgulhosos de mim. Ao Francisco e ao Paulo por fomentarem em mim todos os mais diversos conhecimentos de música, literatura e cultura e por todo o companheirismo dos últimos anos. Aos meus primos Cristina, Luis, João, Maria João, Leonor, Matilde, Sara, Carolina, Sofia e Tio Alberto. Ao meu irmão Migas, obrigado por tudo e vemo-nos amanhã em Babilónia.

Aos meus segundos familiares Nuno, Catarina, Filipa Faria, Filipa Palha, Pita, Lourenço, Martim, Zé Maria, Matilde, Gonçalo e Duarte. Obrigada por uma infância inesquecível.

À Bia, o Gonçalo, a Leo e o Pedro, amigos com quem iniciei esta louca jornada de aventuras. Obrigada pela cumplicidade e companheirismo nas memórias que guardarei carinhosamente. Ao Newton, querido afilhado, por me aturares. À Bruna por toda a amizade e ajuda. À Carol por toda a paciência. À Ana Raquel, à Patrícia, ao Marcelo e ao Paim.

À minha companheira Carolina.

Às amizades que forjei no curso e na faculdade, caloiros, colegas e veteranos, com os quais tive o prazer de trabalhar, conviver e conhecer. Para a história fica a amizade, aquele amor que nunca se perde.

Ao grupo dentro do grupo de introvertidos e extrovertidos, todos com o mesmo amor pela música e pelas calúnias da vida de estudante, obrigado por todas as histórias que jamais poderemos contar aos nossos netos e, acima de tudo, obrigado por existirem.

Por último, obrigado ao Professor Santana pela faculdade na qual tive o prazer de

entrar e viver, e na qual, com muita tristeza minha, não pude acabar o curso.

Abstract

According to Dirac, the main challenge is that the application of these fundamental laws results in equations that are too difficult to solve for the mathematical treatment of a significant portion of physics and the entirety of chemistry. Even with today's most cutting-edge technological resources, simulating a quantum mechanical system is typically a very difficult task; one of the primary reasons is related to the number of parameters required to represent a quantum state. This type of perspective when approaching complex mathematical situations in the field of physics still holds true today. Tensor Networks can provide an efficient approximation to certain classes of quantum states, and the associated graphical language makes it easy to describe. The Density Matrix Renormalization Group (DMRG), a tensor network method, is a numerical algorithm for the efficient truncation of the Hilbert space of low-dimensional strongly correlated quantum systems based on a rather general decimation prescription. This work aims to analyze the way in which Tensor Network Methods like MPS (Matrix Product State) and DMRG can be applied in order to achieve a more efficient examination of abstract universes with correlation between objects, specifically a text universe, and give a result which can be use to describe the system without the risk of high dimensionality and the use of great computational power. The possibility of applying these different methods for analysis can have an impact, not only in the field of condensed matter theory but on the study of black holes, quantum computing and the development of the holographic universe theory.

Keywords: Tensor Networks, Matrix Product State, Density Matrix Renormalization Group, Quantum Mechanical System, Quantum Computing

Resumo

Dirac disse em tempos que as leis fundamentais necessárias para o tratamento matemático de uma grande parte da física e de toda a química são já completamente conhecidas, e a dificuldade reside apenas no facto de que a aplicação dessas leis leva a equações que são demasiado complexas para serem resolvidas. Este ponto de vista ao abordar matematicamente situações complexas no campo da física manteve-se até aos dias de hoje onde, até com o uso dos recursos tecnológicos mais avançados, a simulação de um sistema mecânico quântico é geralmente uma tarefa de extrema dificuldade; um dos principais motivos está relacionado com o número de parâmetros necessários para representar um estado quântico. As redes de tensores (*Tensor Networks*) podem fornecer uma aproximação eficiente para certas classes de estados quânticos, e a linguagem gráfica associada a eles torna fácil a sua descrição. O *Density Matrix Renormalization Group* (DMRG), um método de *Tensor Networks*, é um algoritmo numérico para o truncamento eficiente do espaço de Hilbert de sistemas quânticos de baixa dimensão fortemente correlacionados. Este trabalho tem como objetivo analisar a maneira como os métodos de *Tensor Networks* como MPS (*Matrix Product State*) e DMRG podem ser aplicados para alcançar um estudo eficiente de universos abstratos com correlação entre objetos, especificamente um universo de texto, e dar um resultado que pode ser usado para descrever o sistema sem o risco de alta dimensionalidade e o uso de grande poder computacional. A possibilidade de aplicar estes diferentes métodos de análise pode ter um impacto não apenas no campo da teoria da matéria condensada, mas também no estudo dos buracos negros, computação quântica e no desenvolvimento da teoria holográfica do universo.

Palavras-chave: Redes de Tensores, *Matrix Product State*, *Density Matrix Renormalization Group*, Sistemas Quânticos Mecânicos, Computação Quântica

Contents

List of Figures

List of Listings

1

Introduction

The challenge of analysing a complex system has always been a difficult task across the history of science. The ability to understand a system's behaviour, its context and the interaction that exists between its components is, and was, one of the most essential tools in some of science's most groundbreaking discoveries like Darwin's theory of evolution or Einstein's Theory of Relativity. Two very distinct fields and yet two very similar approaches in terms of trying to make sense of specific phenomenons by analyzing the whole environment that surrounded them.

The arrival of technology, and its rapid evolution, has been an incredible aid in helping researchers tackle the overwhelming task of understanding some of our most complex systems, but even with all the tools that technology has given, there are still limitations that slow down the rhythm of scientific research, such as dimensionality and the demand for higher computational power.

A fundamental challenge in some areas of physics, chief among them condensed matter, is the classical simulation of quantum many-body systems. This difficulty originates in the Hilbert space, where if the size of a system is increasing the Hilbert space will be growing exponentially so that in the polynomial space it becomes impossible to represent a generic many-body state. Thankfully, only a small portion of the Hilbert space is occupied by Hamiltonians (and other physically interesting states) and ground states. It may be possible to avoid the "curse of dimensionality" for some of these states.

Tensor Networks are a class of variational wave functions used in the study of quantum many-body systems [62], they extend one-dimensional matrix product states to higher dimensions while preserving some of their useful mathematical properties. Tensor networks have been useful in studying the dynamics of strongly correlated many-body systems and have also been adapted for supervised learning [77]. Essentially, Tensor Networks have provided the possibility of taking a very complex grid and projecting it to a more basic geometry without losing a significant amount of information. This is why some of the most complex areas of physics, such as the search for a holographic theory of

the universe, have been using tensor network methods to try and understand the interactions of bodies in the universe, while also considering difficult concepts such as gravity, entanglement and black holes.[57][53]

In one-dimensional quantum systems (such as spin chains), the Density Matrix Renormalization Group (DMRG) is a popular and very powerful numerical method for computing ground states. It is essentially a variational algorithm that minimizes the energy over Matrix Product States (MPS). As the name suggests, an MPS is a data structure representing a many-body state by products of matrices. Both MPS and DMRG are examples of Tensor Network Methods.

Thesis Structure

The work described in this thesis is a study of how MPS and DMRG, two Tensor Network Methods, can be applied to a text universe and how they are successful in tackling problems such as the "dimensionality curse" and the demand for high computational power. The text universe serves as an example of a complex system that has a lot of similarities with other complex systems encountered in problems across many areas of physics.

Theoretical Concepts

The first chapter aims to introduce the theoretical concepts necessary to understand the world of tensor networks starting by the mathematical definition of a tensor in the most basic form and its properties. The second section in this chapter shows the theory behind the ability of joining tensors into networks and how tensor methods can be defined, as well as their applications.

The third section of this chapter explains the Quantum N-Body Problem, which has an important conclusion that relates to the application of DMRG, concerning the importance of knowing the Ground State Energy of a system, in order to describe it. The sections titled Introduction to MPS and Matrix Product Operator present the most basic configurations possible (of linear geometry) of a tensor network: the Matrix Product State (MPS), a data structure representing a many-body state by products of matrices, and the Matrix Product Operator (MPO), a tensor network where each tensor has two external, uncontracted indices as well as two internal indices contracted with neighboring tensors in a chain-like fashion.

The Density Matrix Renormalization Group is presented in the sixth section of the theoretical concepts as a more advanced method, one that is very popular and the most powerful numerical method for computing ground states. It is essentially a variational algorithm that minimizes the energy over matrix product states (MPS).

The Tensor Renormalization Group section serves to introduce how RG methods can be applied for coarse-graining. This is an important concept to understand the DMRG

and its application. Finally the Language Model is introduced in order to define a text universe as a possible type of tensor network which validates it as an object of study for DMRG application.

Matrix Product State in Action

In this chapter, it is shown how to apply the Matrix Product State in Python and how it is a useful tool to work around the "dimensionality curse" and help diminish the need for high computational power. The "curse of dimensionality", as coined by Bellman [6], refers to various phenomena that arise when analyzing and organizing data in high-dimensional spaces that do not occur in low-dimensional settings such as the three-dimensional physical space of everyday experience.

Density Matrix Renormalization Group in Action

This chapter starts by expanding on what is introduced in the Theoretical Concepts and shows a general review of the whole sweep process of an infinite DMRG as well as its application on a text. This application is created through a program in C that was implemented in order to get a Ground State Energy as well as other information about the system.

2

Theoretical Concepts

2.1 Tensors

The first important thing to understand is the concept of tensors, and for that it is necessary to understand scalars and vectors. In the case where we want to specify a value for a certain point, for example, the temperature in the city of Lisbon, one could simply use a number with a corresponding unit like 293 Kelvin which is equivalent to 20 degrees Celsius. In other words, a quantity that has a magnitude is called a scalar [79]. To specify the temperature in Lisbon only single number is needed (Figure 2.1).

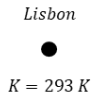


Figure 2.1: Scalar: a single component with no basis vectors

In a different case where you want to know the displacement between points A and B, which can be for example the JFK airport and the top of the Empire State Building respectively, the magnitude of the displacement ($\|\vec{d}\| = 22.5$ km) would not be enough. A direction is also necessary, which is specified by the vector $\vec{AB}$. A vector has both a magnitude and a direction to specify the displacement and can be broken down into three components because it exists in a three dimensional space. So, considering a coordinate system xyz, the displacement is specified using three components (Figure 4.6).

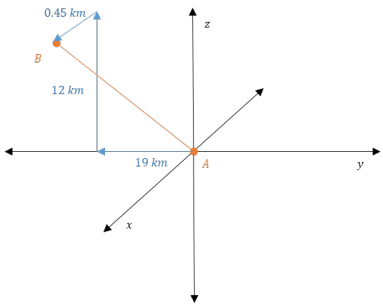


Figure 2.2: Vector: Tensor of rank 1.

The displacement $\vec{d}$ can be written as the sum of the components in the coordinate

system using the unit vectors ijk:

$$\vec{d} = -19\hat{i} + 12\hat{j} + 0.45\hat{k}. \tag{2.1}$$

Therefore, since the displacement is given by the sum of three component vectors, the vector that describes the displacement between A and B has three components and one basis vector for each component.

For a next level case, let's consider a point O inside a steel beam of a bridge (Figure 2.3). To specify the stresses that are acting on point O, the first thing to do would be to make three cross sections of the beam through this point.

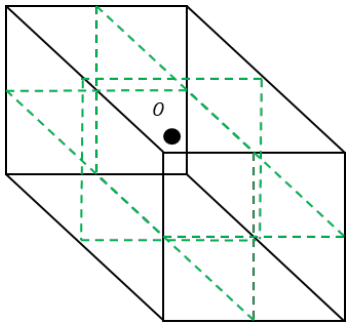


Figure 2.3: Point O in a steel beam and the three cross sections

These cross sections would be made according to the orientation of a coordinate system xyz. The first cross section would be perpendicular to the x direction (Figure 2.4(a)), the second is perpendicular to y (Figure 2.4(b)) and the third to z (Figure 2.4(c)). Because the beam is under stress, the point O is experiencing a force per unit area, and this force can be broken up into three components in each cross section (Figure 2.4). Each component is represented by P and two subscripts: the first denotes the direction that the area is perpendicular to, and the second denotes the direction that the force is acting on.

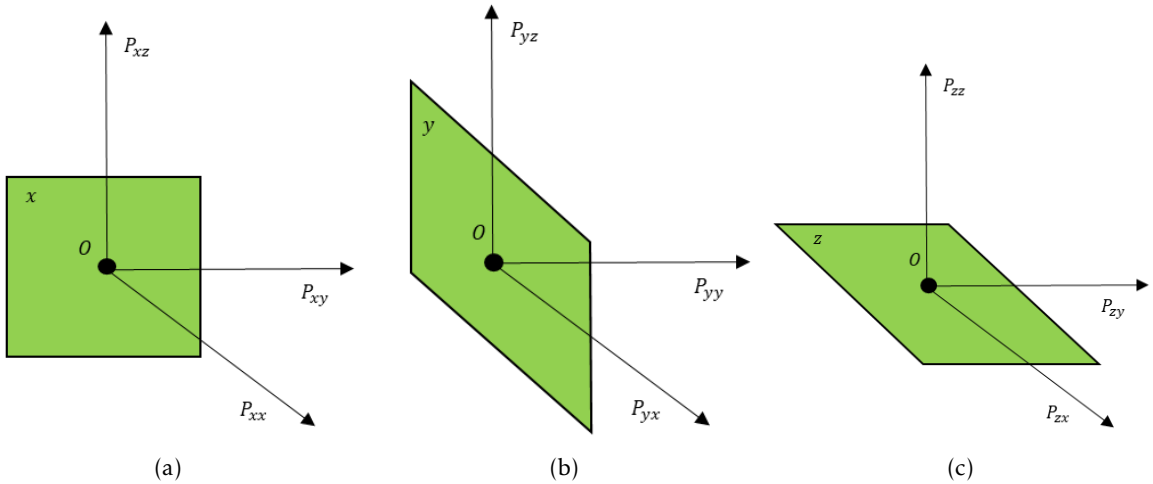


Figure 2.4: Cross Sections of the steel beam and the representation of the force per unit are that acts in each one.

All these force per unit area components can be combined into a 3×3 matrix[22]:

$$P = \begin{bmatrix} P_{xx} & P_{xy} & P_{xz} \\ P_{yx} & P_{yy} & P_{yz} \\ P_{zx} & P_{zy} & P_{zz} \end{bmatrix}. \tag{2.2}$$

The forces in the same direction cannot be added: P_{xx} and P_{yx} are acting on the same x direction but the nature of these forces is different, one is pulling and another is shearing, which causes the steel beam to deform in different ways, this is why it is important to specify the surface that the force acts on. The matrix P specifies the stress that acts on point O, where each component corresponds to a force per unit area on a particular surface that the point O inhabits.

The matrix P has nine components and each is specified by a magnitude and two basis vectors, one for the cross-sectional area that is being acted on and another for the force acting on that area.

All three of these mathematical objects (the scalar, the vector and the matrix) have something in common: they are all called tensors[45][13]. If one exists in an M dimensional space, a tensor of rank n is a mathematical object that has n indices, m^n components, and obeys certain transformation rules. The rank of a tensor can be thought of as the number of basis vectors needed to fully specify a component of the tensor[4]. Considering the first case where 0 basis vectors were needed to specify the scalar component, it is then possible to consider that the scalar is a tensor of rank 0. Using this same train of thought, the vector in the second case (4.6) is a tensor of rank 1 and the matrix P is a tensor of rank 2. A tensor of rank 3 can be represented through a 3-D array and in this case each component is specified by three basis vectors, and there are a total of $3^3 = 27$ components [45][19].

Two important properties of a tensor are the covariance and the contravariance. A tensor is invariant, meaning that, if, for instance, the coordinate system in the second case is transformed, the vector components change and the basis vectors will change as well, but the displacement vector still has the same magnitude and it's still pointing from point A to point B. In general, vector components change under a change of coordinate system in one of two ways: in a covariant fashion and in a contravariant fashion [14].

A contravariant vector is one where the basis vector and the component are transformed in a opposite manner, the components of the vector are said to be *contravariant*. Contravariant components are specified using superscripts opposite of basis vectors which are specified with subscripts. Examples of vectors with contravariant components include the position of an object relative to an observer, or any derivative of position with respect to time, including velocity, acceleration, and jolt [74].

For the case of the covariant vector, a covariant component transforms in the same manner as the basis vector and it is specified using subscripts. Examples of covariant vectors generally appear when taking the gradient of a function. Covariance and contravariance are particularly important for understanding how the coordinate description

of a vector changes by passing from one coordinate system to another. They are also important to understand tensor contraction [33].

Supposing that S is a *(p,q)* tensor, meaning, that the contravariant rank is p and the covariant rank is q, such that:

$$S = (S^{i_1,i_2...i_p}_{j_1,j_2,...j_q}). \tag{2.3}$$

For the contraction of S with respect to a contravariant index i_f and covariant index j_g where f and g are any number from 1 to p and 1 to q respectively, each of these indices are replaced by the same index u

$$i_f = j_g = u. \tag{2.4}$$

Since now there is a repeated index, a summation is conducted over u from 1 to n to end up with the contracted matrix S':

$$S' \equiv (S^{i_1...u...i_p}_{j_1...u...j_q}). \tag{2.5}$$

The contraction S' of S is a tensor with a contravariant rank reduced by 1 and a covariant rank reduced by 1: *(p-1,q-1)*. Tensor contraction can be seen as a generalization of the trace, which is an important tool in many areas of study in physics [74][33].

2.2 Tensor Network Theory

Tensor Network Theory has been growing in recent years in its application in various fields of quantum information processing. It can be placed as a language which is transversal to many fields such as the quantum realm of physics, computer science and mathematics. [8].

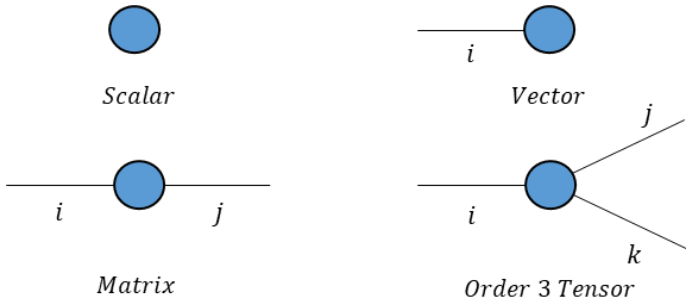


Figure 2.5: Examples of diagram representation. Considering the simple representation of a dot as a scalar (Tensor of Rank 0) and each "arm" representing a basis vector (or an index). This simplified way of representing tensors is important to better understand tensor calculus and is used throughout the work present in this thesis.

In mathematics, the idea of multi-linear maps, or functions with multiple parameters that are linear with respect to each parameter, is encapsulated and generalized by the term tensors. A collection of tensors connected to one another by contractions can be used to observe and depict a tensor network relatively simply. [11].

In terms of representation, a tensor is a labelled shape having zero or more open input legs pointing down and zero or more output legs pointing up, such as a box, oval, or triangle. The upper and lower indices are corresponding to specific arms (Figure 2.5)[65]. The relevant indices are contracted (summed over) when two tensor arms are connected by a wire, some examples of this are represented in Figure 2.6. A *tensor network* is made up of two or more tensors in a diagram [9].

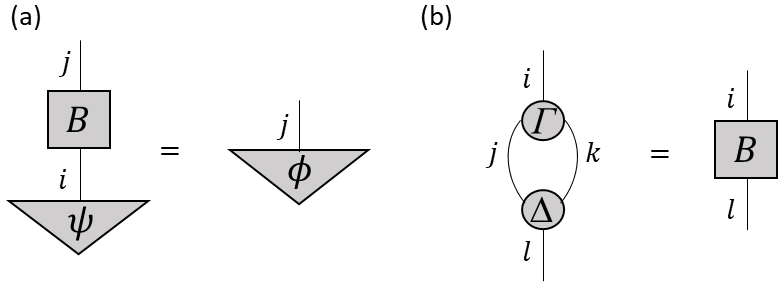


Figure 2.6: Two examples of wire connecting between tensors. The first one (a) is a diagram equivalent to $A_i^j(\psi)^i = (\phi)^j$, which is equivalent to a matrix multiplying a vector resulting in another vector. The wire i is fully connected and therefore the corresponding index is summed over. The diagram (b) is a contraction of two indices between two rank-3 tensors and is equivalent to $(\Gamma)_i^{jk}(\triangle)_l^j k = B_l^i$.[3].

Pines and Lauglin said in 2000 regarding the Schrödinger Equation that it "cannot be solved accurately when the number of particles exceeds about 10. No computer existing, or that will ever exist, can break this barrier because it is a catastrophe of dimension." [48]. This is a very pessimistic view of Quantum Mechanics and it is based on how many particle systems are perceived. One can argue that this negative perspective is due to the illusion of complexity, meaning that the complexity that appears in core mechanics isn't always there. Another, more elegant way, to state this is that "Nature does not explore all possibilities", therefore once one recognizes that since physical quantum states have some special structure then perhaps the exponential complexity isn't there so long as one knows what the special structure of physical quantum states is.

One of the most important characteristics of physical quantum states is the concept of locality [32], meaning that the response of systems is relatively local. In mathematical terms, if we look at the lower energy sector of the physical world the states have very little entanglement so if one does a measurement at one point you do not observe correlations very far away.

Tensor Networks are the key to study these approaches, in other words, they provide a language for low entanglement states because they provide a natural way to pick out only

the physically relevant portion of quantum mechanics[20][11][55]. There are very different kinds of tensor networks that reflect different geometries of entanglement (Figure 2.7).

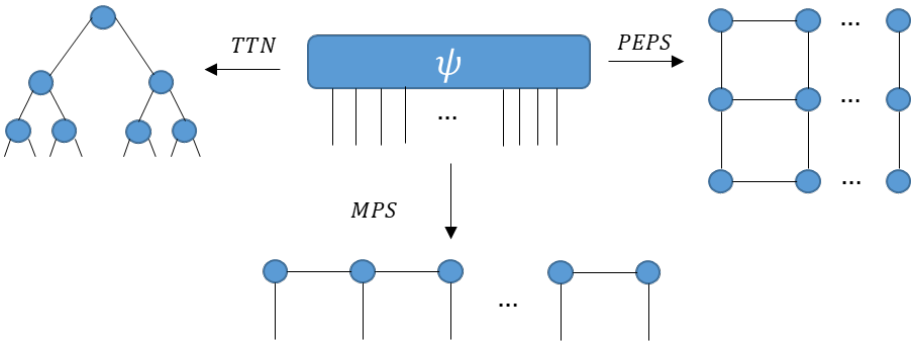


Figure 2.7: The factorization of a quantum state *psi* into several network architectures is the idea that is being expressed. Matrix Product States (MPS), Projected Entangled Pair States (PEPS), and Tree Tensor Networks (TTN) are the applications. These techniques simulate a complex quantum state using a straightforward, regular structure, effectively using lossy data compression that protects the key characteristics of the quantum state. [20].

Let us consider the following graphical language where a general wave function is written as in (2.6).

$$|\psi\rangle = \sum_{n_1 n_2 n_3} \psi^{n_1 n_2 n_3} |n_1 n_2 n_3\rangle. \tag{2.6}$$

In this equation there is a quantum state $|\psi\rangle$ which is expanded with a set of amplitudes $\psi^{n_1 n_2 n_3}$ and a set of basis states $|n_1 n_2 n_3\rangle$. The basis states will be different depending on the system it is representing (for example for spins n would have the state "up" or "down")[30]. So a wave function can be represented by a set of numbers $\psi^{n_1 n_2 n_3}$ which can also be called a tensor [55] and can be drawn as represented in Figure 2.8.

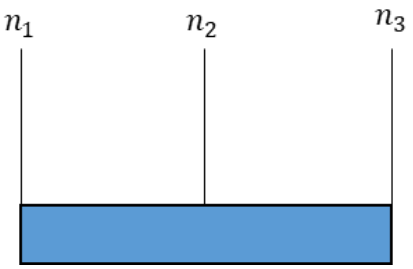


Figure 2.8: General State of a tensor in graphical form where each leg corresponds to one of the sites and the thick line to a single tensor.

Returning to the concept of low entanglement states, the best way to have a basic grasp of the concept of entanglement is to first look at the the limit scenario where a state has no entanglement[46], considering a system with only 2 parts, i.e. two spins:

$$No\ entanglement:\ \psi^{n_1 n_2} = A^{n_1} A^{n_2}. \tag{2.7}$$

In physical terms, no entanglement means that local measurements on one of the systems are completely uncorrelated or can be done independently of each other [36]. This is actually a local criteria for local realism as was defined by Einstein [47].

For the scenario where there is some entanglement and considering ψ a matrix (that is just an object with two indices), then one can break up a matrix into simpler components, so the wave function amplitude $\psi^{n_1 n_2}$ can be written not as a simple product of amplitudes (like in (2.7)) but as a sum of products of amplitudes:

$$Entangled:\ \psi^{n_1 n_2} = \sum_i A_i^{n_1} A_i^{n_2}. \tag{2.8}$$

This is essentially a matrix factorization. The reason it is called low entanglement is because there is a small number of terms in the sum, the summation index i is what is generating the entanglement.

Because some types of quantum systems may now be replicated more effectively, there has been an increase in interest in numerical methods that perform tensor contractions [20]. This has created new opportunities for a deeper understanding of specific physical systems (Figure 2.7). The term "tensor network methods"refers to the complete group of related tools that are frequently used in contemporary condensed matter physics, mathematics, and computer science as well as quantum information science [24].

2.3 Quantum N-Body Problem

Several theories related to the search for a "theory of everything" [48], such as quantum gravity [27][44] and string theory [5][80], are inclined to the existence of a minimal observable length in nature. As a result, research into quantum theories has gained traction in theoretical physics. Modifying the quantum mechanics' standard commutation between the position and momentum operators is a cost-effective technique to introduce such a minimal length.

A modified Heisenberg algebra with a limited length has been used to study the one-dimensional quantum N-body problem [60][39]. It has been demonstrated that the ground-state energy of the system under study, which consists of N relativistic particles with mass m interacting via a pairwise potential, can be bounded from below by a simple formula that only requires knowledge of the ground-state energy of a correlating two-body system [12].

The starting point would be to consider the relative positions $r_{jk} = x_j - x_k$ naturally used in standard quantum mechanics when dealing with systems of potential $V(x_j - x_k)$. The N-body Hamiltonian can be written as:

$$\hat{H}^{(N)} = \sum_{j=1}^{N} \frac{\hat{p}_j^2}{2m} + \sum_{j<k=1}^{N} V(\hat{x}_j - \hat{x}_k) \tag{2.9}$$

which describes a one-dimensional system of N-particles with mass m interacting via the pairwise potential V. The shape of this Hamiltonian suggests the introduction of "modified" relative positions:

$$\hat{r}_{jk} = \hat{x}_j - \hat{x}_k. \tag{2.10}$$

In analogy with the standard case of the relative momenta $\pi_{jk} = (p_j - p_k)/2$ the modified Heisenberg can be obtained as:

$$[\hat{r}_{jk}, \hat{\pi}_{jk}] = i(1 + \beta\hat{\pi}_{jk}^2). \tag{2.11}$$

To recover the standard Heisenberg algebra at the lowest order, one has at order $\hat{p}^2$

$$\Theta(\hat{p}) = 1 + \beta\hat{p}^2 \tag{2.12}$$

where the parameter β should be such that $\beta\langle\hat{p}^2\rangle \ll 1$. Considering the ansatz (2.12), to find the explicit form of $\hat{\pi}_{jk}$ the Heisenberg algebra considered comes from an expansion of $\hat{p}$: It is thus sufficient for the formulas to be valid at order $\hat{p}^2$. Assuming this, the commutation relation in (2.11) can be written as:

$$\hat{\pi}_{jk} = \left(\frac{\hat{p}_j - \hat{p}_j}{2}\right)\left(1 - \frac{\beta}{4}(\hat{p}_j + \hat{p}_k)^2\right). \tag{2.13}$$

This is anti-symmetric in j, k and reduces to the standard relative momentum for $\beta = 0$. Still at the second order in the momenta $\hat{p}_j$, one has

$$\frac{4}{N}\sum_{j<k=1}^{N} \hat{\pi}_{jk}^2 = \sum_{j=1}^{N} \hat{p}_j^2 - \frac{1}{N}\left(\sum_{j=1}^{N} \hat{p}_j\right)^2 \leq \sum_{j=1}^{N} \hat{p}_j^2. \tag{2.14}$$

The above inequality yields the following lower bound of the Hamiltonian 2.9:

$$\hat{H}^{(N)} \geq \sum_{j<k=1}^{N}\left[\frac{\hat{\pi}_{jk}^2}{2\mu} + V(\hat{r}_{jk})\right] \tag{2.15}$$

where $\mu = \frac{mN}{4}$. Since the lower bound Hamiltonian (2.15) is separable, it can be shown that a lower bound on the ground-state energy ε^N of $\hat{H}^{(N)}$ is given by:

$$\varepsilon^{(N)} \geq E^{(N)} = \frac{N(N-1)}{2}E^{(2)} \tag{2.16}$$

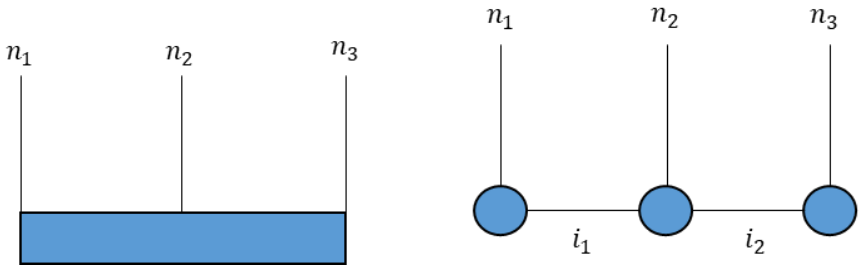
Where $E^{(2)}$ is the ground-state energy of the two-body Hamiltonian

$$\hat{H}^{(2)} = \frac{\hat{\pi}^2}{2\mu} + V(\hat{r}). \tag{2.17}$$

In this equation $\hat{r}$ and $\hat{\pi}$ satisfy $[\hat{r}, \hat{\pi}] = i(1 + \beta\hat{\pi}^2)$. The lower bound (2.16) implicitly assumes that the spatial wave function of the bound state is totally symmetric. Following (2.16), any two-body problem with modified Heisenberg algebra whose ground-state energy is known can be used to bound from below the ground-state energy of a corresponding N-body problem.

2.4 Introduction to Matrix Product State

The Matrix Product State (MPS) or Tensor Train (TT) tensor network is a factorization of a tensor with N indices in to a chain-like product of three index tensors [55][18]. It is one of the best understood tensor networks and can be expressed in tensor diagram notation where one considers the tensor ψ:



(a) General State (b) Matrix Product State

Figure 2.9: General State and equivalent MPS

In mathematical form it can be expressed as:

$$\psi^{n_1 n_2 n_3} = \sum_i A^{n_1}_{i_1} A^{n_2}_{i_1 i_2} A^{n_3}_{i_2}. \tag{2.18}$$

Each tensor A can be different to other tensors A, therefore it is useful to distinguish them by their indices. In (2.18) the indices i are contracted, or summed over, and are called bond indices. Any tensor can be exactly represented in MPS form for a large enough dimension of the bond indices i [63]. Each bond index has what is called a bond dimension or tensor-train rank. To understand the matrix product state factorization one has to understand the bond dimension, which is the dimension of the bond index connecting one tensor in the chain to the next, and can vary from bond to bond. The bond dimension can be thought of as a parameter controlling the expressivity of a MPS/TT network. In the equation (2.18) the bond dimension is the dimension of the indices i, where for example, i_1 is connecting site 1 and site 2, they are therefore entangled which makes matrix product state essentially a 1D structure of entanglement. The bond dimension fundamentally says how much correlation there is between two sites [55][66][61], and, in a larger sense, it gives the dimensionality that can be applied to a system. It is with this that MPS keeps the ambitious promise of being able to transform a grid geometry into a linear geometry while losing very little information. A good analogy to visualize this is to think of an MPS like a genetic code which is also a linear geometry of simple connections and yet a singular connection can change the whole outcome, or in other words, "it only takes 1% to separate us from monkeys". The bond dimension determines the space complexity of the MPS and how many degrees of freedom the system can have.

Considering a tensor with N indices of dimension d, the tensor can always be represented exactly as an MPS with bond dimension $m = d^{N/2}$. In most applications MPS is

used as an approximation where the bond dimension is either fixed at a moderate size or determined adaptively.

A tensor like the one mentioned above (N indices of dimension d) is generically specified by d^N parameters. In contrast, representing such a tensor by an MPS/TT network of dimension m requires Ndm^2 parameters. This number of parameters can even be further reduced by imposing or exploiting certain constraints on the factor tensors. This difference in the number of parameters represents a massive compression from a set of parameters growing exponentially with N, to a set of parameters growing just linearly with N and that in itself represents a huge advantage in the use of MPS networks to simplify the analysis of systems.

An important point regarding matrix product states is that they have a gauge degree of freedom, they are not unique. Considering an arbitrary identity matrix G:

$$GG^{-1} = 1 \tag{2.19}$$

which, in diagram notation, can be represented as:

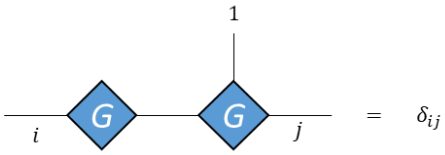


Figure 2.10: Representation of the product of G and its inverse G^{-1} in graphical form. The product of this is a Kronecker Delta, which in graphical terms is a line

So taking the MPS of three sites of Figure 2.9(b) and inserting the arbitrary product of G and G inverse (Figure 2.10) in between the first two tensors like in Figure 2.11(a) the product will still be the same.

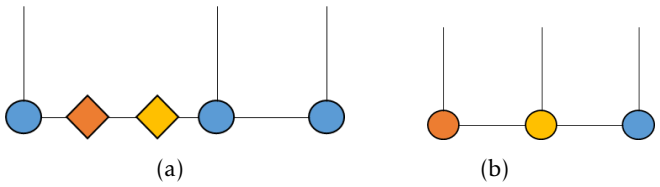


Figure 2.11: Representation of the gauge of freedom in Matrix Product States. (a) MPS with inserted gauge matrices. (b) Resulting MPS after multiplying the matrices with the left and right site respectively

The result will be a different matrix product state representation (Figure 2.11(b)) of the same state as before and that is the gauge degree of freedom.

Another property of MPS is the possibility of contraction between matrix product states. As previously shown, there is an MPS ($|\psi\rangle$) which when contracting with another similar MPS ($|\phi\rangle$) can be written as:

$$\langle\psi|\phi\rangle. \tag{2.20}$$

This can be represented in graphical form as in Figure 2.12.

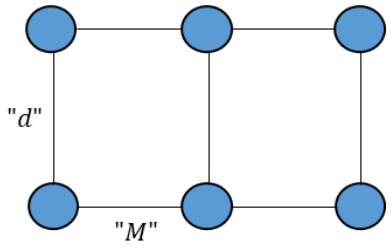
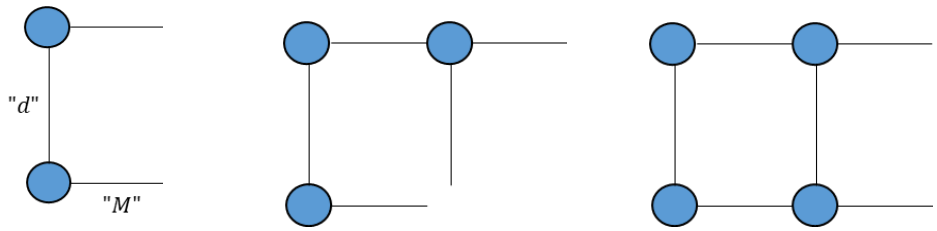


Figure 2.12: Graphical representation of a contraction of two Matrix Product States

In order to compute this contraction efficiently one has to consider that each of the individual objects are small tensors and can be computed to overlap in pieces and in the correct order (Figure 2.13).



(a) M^2d (b) M^3d (c) M^3d

Figure 2.13: MPS Contraction: (a) beginning by combining the first upper site tensor with the bottom site tensor and then (b) add the second upper site tensor and then (c) contract the second bottom site tensor which returns to the same position as in (a). The value in each subcaption represents the dimension in each scenario.

Contracting along the direction shown in Figure 2.13 will result in the efficient computational scheme and the total overlap will have a cost of $O(M^3dL)$

2.5 Matrix Product Operators

An operator can also be written as a tensor very similarly as before in (2.6):

$$\hat{O} = \sum_{nn'} O^{n_1 n_2 n_3 \ldots n_L}_{n_1' n_2' n_3' \ldots n_L'} |n_1 n_2 n_3\rangle\langle n_1 n_2 n_3| \tag{2.21}$$

In the same way that it is possible to write a state down as a product of bits on each site, it is also possible to write down a general operator acting on L sites as a product of small operators that act on each site correspondingly, and all the operators are entangled (2.22). A Matrix Product Operator is a way of writing down a general operator as an entangled product of operators[67][54][51].

$$O^{n_1 n_2 n_3 \ldots n_L}_{n_1' n_2' n_3' \ldots n_L'} = \sum W^{n_1 n_1'}_{i_1} W^{n_2 n_2'}_{i_1 i_2} \ldots W^{n_L n_L'}_{i_L} \tag{2.22}$$

The graphical representation of (2.22) (Figure 2.14) is very similar to the MPS representation (Figure 2.9).

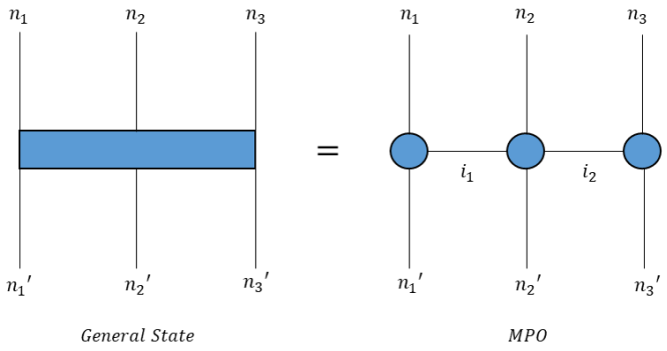


Figure 2.14: Graphical representation of a Matrix Product Operator as an entangled product of operators.

Matrix Product Operators are analogous to Matrix Product States therefore they also have bond dimensions. Considering first the Heisenberg Hamiltonian:

$$H = \sum_{\langle ij \rangle} S_i \cdot S_j \tag{2.23}$$

To find the bond dimension, first a cut is made to the system to separate it into the left and right systems (Figure 2.15).

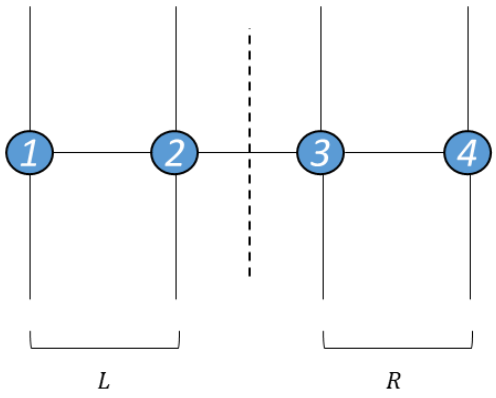


Figure 2.15: Representation of a typical MPO where the dotted line represents the joining of the pairs on both sides

This division is represented as:

$$H_L \otimes 1_R + 1_L \otimes H_R + \sum_{\alpha=x,y,z} S_2^{\alpha} \cdot S_3^{\alpha}. \tag{2.24}$$

Here we have the summations of each side of the Hamiltonian (H_L and H_R) multiplied by the identity of the opposite side (1_R and 1_L) added to all the couplings between left and right ($\sum_{\alpha=x,y,z} S_2^\alpha \cdot S_3^\alpha$). This is a general way of writing a Hamiltonian in terms of bits on the left and on the right. Because there are five terms in the equation, the bond dimension of this MPO is five.

In terms of computation, the scenario where an MPO acts on an MPS is very straightforward. Considering an MPS and MPO of such form and dimension as in Figure 2.16.

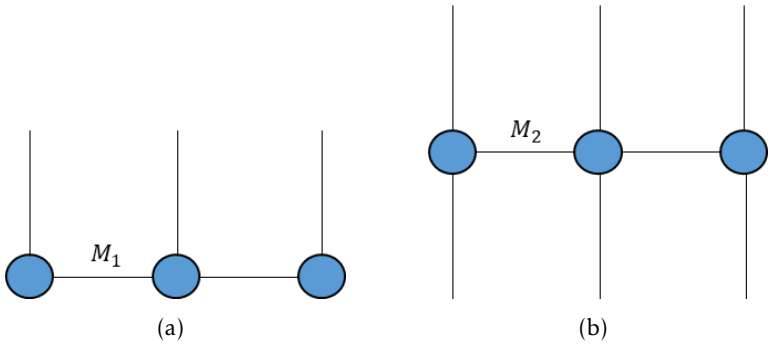


Figure 2.16: Graphical representation of an MPS and an MPO with corresponding dimensions M_1 and M_2 respectively.

In graphical terms this operation would simply be as represented in Figure 2.17.

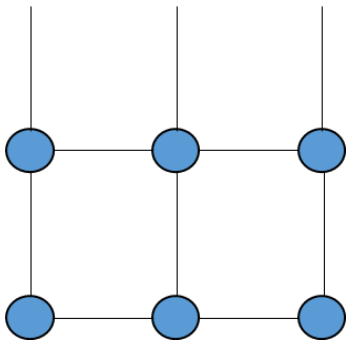


Figure 2.17: Contraction between an MPS and MPO

And the final result would be a new Matrix Product State with a dimension equal to the product of the bond dimensions (M_1 and M_2)(Figure 2.18).

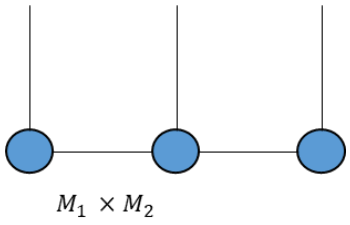


Figure 2.18: Final graphical outcome of an MPO acting on an MPS

A result of this is that entanglement has increased. An operator acting on a state increases its entanglement, so it's important, in practical terms, to be able to control the

level of entanglement to help computation.

2.6 Introduction to DMRG

The Density Matrix Renormalization Group (DMRG) method was introduced in 1992, rooted in the numerical Renormalization Group by Wilson [90][10]. Though later modified to simulate small two-dimensional systems, it was initially developed as a numerical approach effective for replicating ground state characteristics of one-dimensional quantum lattices, such as the Heisenberg model. The DMRG approach iteratively adds degrees of freedom to a numerical representation of a microscopic Hamiltonian in a specific basis [86, 88, 71]. The original Hamiltonian will then be changed to integrate out and account for the less significant ones. There will be new and changed couplings in the revised Hamiltonian. By doing this, a streamlined effective Hamiltonian that captures the system's fundamental physics is obtained.

An efficient simulation with DMRG is possible if the entanglement of a subsystem, with respect to the whole, is bounded (or grows logarithmically with its size), and ground states of one-dimensional lattices satisfy this requirement. In a higher dimension, the entanglement is subject to an area law. This connection between efficiency and entanglement behavior was demonstrated by Vidal in 2003 [84].

DMRG can also have another interpretation in relation to its tensor network counterpart: MPS, as an algorithm that optimizes the tensor network of an MPS and then it becomes a dominant eigenvector of a matrix H.

DMRG is mainly used in different applications of the realms of physics and chemistry in problems related to quantum m-body systems in order to find the ground state of an hamiltonian. It is also being extended to other computating applications in those same realms. [89, 91].

2.7 Tensor Network Renormalization

To tackle the problem of understanding the collective low-energy behaviour of a strongly interacting quantum many-body system in D spatial dimensions (described by a microscopic Hamiltonian H) there have been plenty of methods proposed which are based on the Renormalization Group (RG), that is, on studying how the physics depends on the scale of observation. An RG is, by definition, a formal apparatus that allows systemic investigation of the changes of a physical system as viewed at different scales [29].

Perturbative RG methods like Momentum Space Renormalization Group (MSRG) can be used to address weakly interacting systems, but in the case of strongly interacting systems, non-perturbative, *real space* RG methods are often required, like the ones pioneered by Kadanoff [26] and Wilson, which opened the path to other methods such as White's DMRG (Density Matrix Renormalization Group) [2] for quantum spin chains. The latter established how to systematically preserve the ground state wave-function during

real-space coarse-graining. Another RG non-perturbative method was the one created by Levin and Nave [49]: TRG (Tensor Renormalization Group) which taught how to coarse grain (Figure 2.19) Euclidean path integrals of dimensional quantum systems.

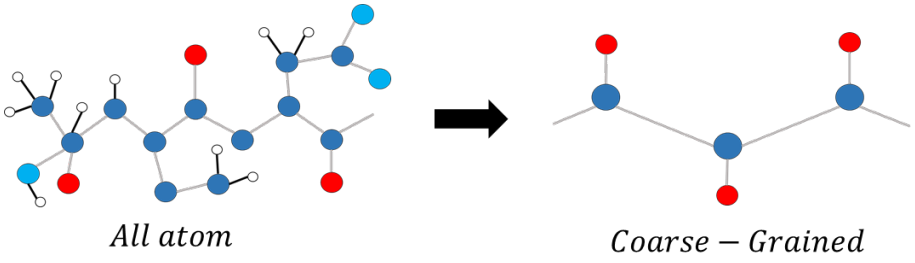


Figure 2.19: Representation of a Coarse-Grained Model. The aim of this type of model is simulating the behaviour of complex systems using their coarse-grained (simplified) representation [25].

Despite the flexibility of both methods, they result in a coarse-grained, useful description of the system that yet includes certain unimportant microscopic details. This prevents them from defining an appropriate RG flow with the right fixed point structure. In D ≥ 2 dimensions, where the accumulation of damaging and unnecessary microscopic degrees of freedom is more pronounced, they also struggle to manage essential systems [28].

The concept of entanglement renormalization was proposed to solve the difficulty stated above in the context of ground state wave functions [82]. *Disentanglers* eliminate short-range entanglement, making it possible to produce an appropriate RG flow and an RG transformation that is computationally viable even under extreme conditions. Additionally, the Multi-scale Entanglement Renormalization Ansatz (MERA) of entanglement renormalization results in an effective tensor network description of ground states for critical systems [83].

Tensor Network Renormalization (TNR) is a different approach that was made to address the same issues in the context of Euclidean path integrals [29]. Given that a tensor network composed of copies of a single tensor A that extends in both the space and Euclidean time directions is used to represent the Euclidean path integral $Z \equiv tr(e^{-\beta H})$:

$$A \to A' \to A'' \to \cdots \to A_{fp}. \tag{2.25}$$

The sequence of tensors (1) is produced by TNR. This sequence corresponds to increasing lengths which flow towards an infrared fixed-point tensor (A_{fp}). This tensor retains only the universal features of the phase or phase transition. The key of this approach is once again the removal of short-range correlations by disentanglers. There is, therefore, a close connection between MERA and TNR. In fact, when applied to the Euclidean path integral restricted to the upper half plane, TNR generates a MERA for the ground state of Hamiltonian H and produces a MERA for the thermal Gibbs state $\rho_\beta \equiv e^{-\beta H/Z}$ at finite

inverse temperature β, as well as for the low energy eigenstates of H on a finite periodic chain [28].

TNR is a better alternative route than MERA because it bypasses the costly energy minimization of previous algorithms and has significant conceptual and computational advantages.

Considering the work made by Vidal and Evenbly [29] where, given a translation invariant Hamiltonian H in one dimension, a standard procedure was used in order to produce a two-dimensional tensor network representation of Euclidean time operator $e^{-\beta H}$ or Euclidean path integral $\mathrm{tr}(e^{-\beta H})$. The tensor is made of copies of a single tensor A. If both the system size L and the inverse temperature β are infinite, then the network spans the entire (x, τ)-plane, where x and τ label space and Euclidean time, respectively.

The following geometries were obtained by introducing a horizontal cut at $\tau = 0$, L and β were chosen to be finite or infinite. In the first one (Figure 2.20), TNR is applied everywhere on the upper half plane (which corresponds to the ground state $|\Psi\rangle$ of H on an infinite lattice) except near $\tau = 0$.

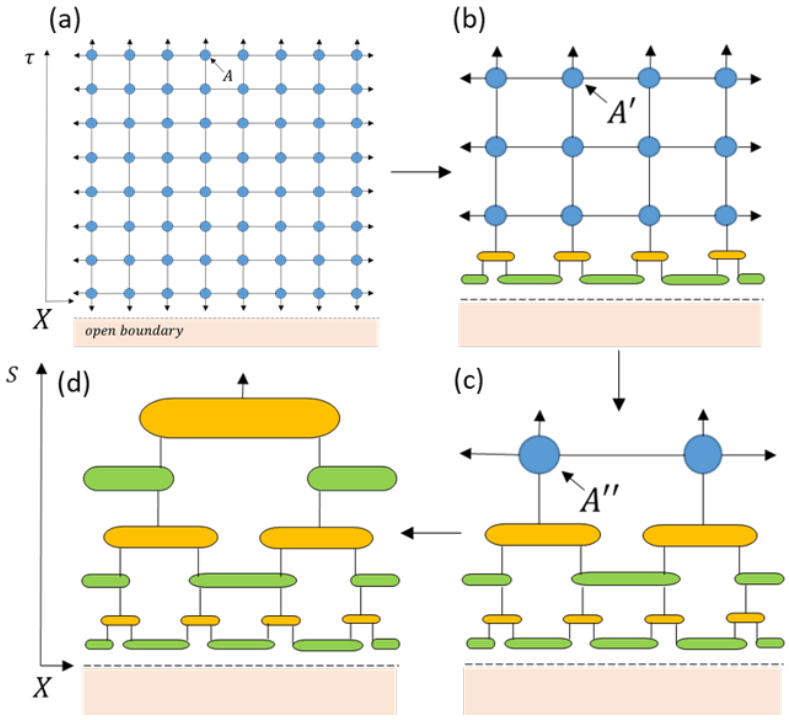


Figure 2.20: (a) The original tensor network of the ground state $|\Psi\rangle$. On (b) and (c) it is represented the results of coarse graining: a tensor network of tensors A' and A'' respectively together with rows of disentanglers and isometries. In the figure (d) it is obtained a full MERA approximation for state $|\Psi\rangle$ through iteration. Adapted from [29].

TNR acts through an intricate sequence of local replacements and the final result is a coarse-grained tensor network with effective tensor A' for most of the upper half plane

in accordance with the equation (2.25), together with a double row of special tensors: disentanglers and isometries.

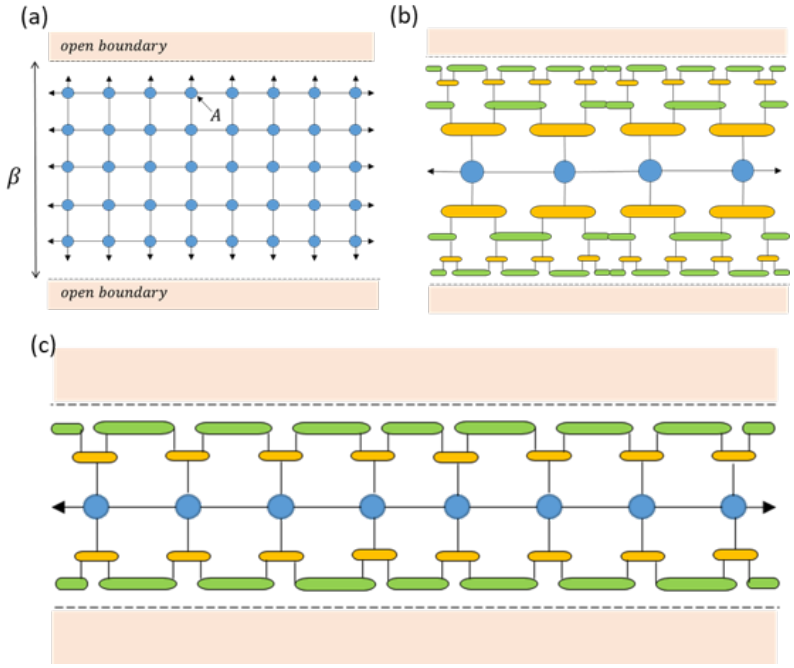


Figure 2.21: (a) Initial Tensor Network on an infinite strip of finite width β. In (b) and (c) it is represented the results of the coarse-graining. Adapted from [29].

Considering now a horizontal strip of finite width β (Figure 2.21) which is proportional to the thermal state $\rho_\beta \equiv e^{-\beta H/Z}$. This time though there are two boundaries, each with an infinite row of open indices: the incoming and outgoing indices of the Euclidean time evolution operator $e^{-\beta H}$. Just like in Figure 2.20 TNR is used to coarse-grain the tensor network, except near its open boundaries.

After $O(\log_2(\beta))$ iterations, a MERA representation of the thermal state is obtained, made of $O(\log_2(\beta))$ double layers of disentanglers and isometries for both the incoming and outgoing indices, together with a central row of tensors.

2.8 Language Model

Although the study of physics and the text world initially appear to be entirely unrelated, there are in fact many remarkable similarities between the two. Establishing this connection can be done by using linguistics, which is the "scientific study of language, and its form, meaning, and context."The MERGE project, which Chomsky directed, was an effort to comprehend the biological view of language and develop a technique for producing linguistic structures [50, 17]. With this approach, /textitK is created by combining two syntactic objects, X and Y, where its origin is indicated by brackets (X,Y) [31].

$$MERGE : X, Y \rightarrow K = X, Y \tag{2.26}$$

One of the subfields of physics is the search for the fundamental laws of Nature, although the term *emergentism* is used to describe the early understanding of emergent laws for aggregates of numerous fundamental entities, not the knowledge of these fundamental rules, which is known as *reductionism*. The essential idea in this discussion is *emergence*: the collective attributes of aggregates of systems may differ significantly from those of the individual systems themselves for a variety of reasons[1]. The study of linguistics can be compared to the study of physics, specifically the Minimalist Program and Emergence.

On a physical level, MERGE may be seen as a form of information coarse-graining according to time scales [43], which enables the representation and interpretation of a text universe as a tensor network. The simplest MERGE accounts for a probability distribution of three variables (α, β, and ψ) using a 3-index tensor of components $M_{\alpha\beta\psi}$ [81].

/par Take into account the probability distribution that two given linguistic elements α and β (e.g. two words) combine to create a new element γ. This distribution can be described by a probability map M with V_{in_1}, V_{in_2} and V_{out} as input and output vector spaces.

$$M : V_{in_1} \otimes V_{in_2} \rightarrow V_{out} \tag{2.27}$$

A 3-index probability tensor named $M_{\alpha\beta\gamma}$ provides the coefficients for this map. The probability of combining α and β into γ (the language input of MERGE) are the entries of this tensor. Physically, the tensor keeps the basic degrees of freedom of the common object at one time scale while coarse-graining the variables α and β at a different time scale. Variable γ is the end outcome of this coarse-graining. The normalizing criteria for probabilities is observed in the tensor.

$$\sum_{\alpha,\beta,\gamma} M_{\alpha\beta\gamma} = 1 \tag{2.28}$$

In terms of physics, this tensor coarse-grains the degrees of freedom α and β at a certain time scale into a new degree of freedom γ at a different time scale. This leads one to the conclusion that a syntax tree's structure can be simply translated into a *tensor network* for the probability distribution $p_{w_1,\ldots,w_n}$ of a phrase. Every syntactic MERGE is represented by the 3-index tensor $M^{[i]}_{\alpha\beta\gamma}$ with i, where i is only a label for identifying certain tensors [73].

3

Matrix Product State in Action

As discussed in the chapter of theoretical concepts, high dimensional tensors have a great cost: there is a scaling of the memory cost of storing and accessing elements of tensors with increasing rank and dimension. While the computational complexity of accessing an item in a multidimensional array is $O(1)$, the main cost is the exponentially growing memory required to store a tensor [42, 76, 85][66].

Starting with a tensor similar to the ones expressed before:

$$T^{s_1,\ldots,s_N} \tag{3.1}$$

where N represents the rank of the tensor, or in physical terms, the size of the system. To exemplify the memory requirements for this tensor, which for reference would look similar to the one in Figure 2.8 in the case of $N = 3$, a code is written in Python [56]:

```
#The first step is to create the "simple" yet complex single tensor

def create_tensor(dimension, rank):
  dim_list = tuple([dimension for _ in range(rank)])
  return np.random.random_sample(dim_list)

#Define the range of the ranks of the tensor to compare the difference given
    the rank of the tensor
ranks = range(2,6)
dimensions = range(2,40)

#Creating a function that will analyse the memory needed for each tensor
for rank in ranks:
  memory = []
  for dim in dimensions:
      tensor = create_tensor(dim, rank)
      component = tuple(np.random.randint(0, dim, rank))
      memory.append(np.sum([x.nbytes for x in tensor]))
      data = tensor[component]
  plt.loglog(dimensions, memory,'o',ls=':', label = f'N = {rank}')

#Creating the Output
```

```
plt.legend()
plt.xlabel('Dimension')
plt.ylabel('Memory cost')
plt.show()
```

Listing 3.1: Python Code of Tensor Creation

The code shown in Listing 3.1 is constituted by a function where a tensor of a given rank with random integers is constructed the code also creates a plot which is printed in the end (Figure 3.1) and this graphic shows basically a tensor with random entries, with ranks and physical dimensions that run over small ranges and also their respective memory requirements. It shows that the memory required to store a tensor is an exponential growth and scales as d^N. This is a problem because it will quickly saturate the available computational resources.

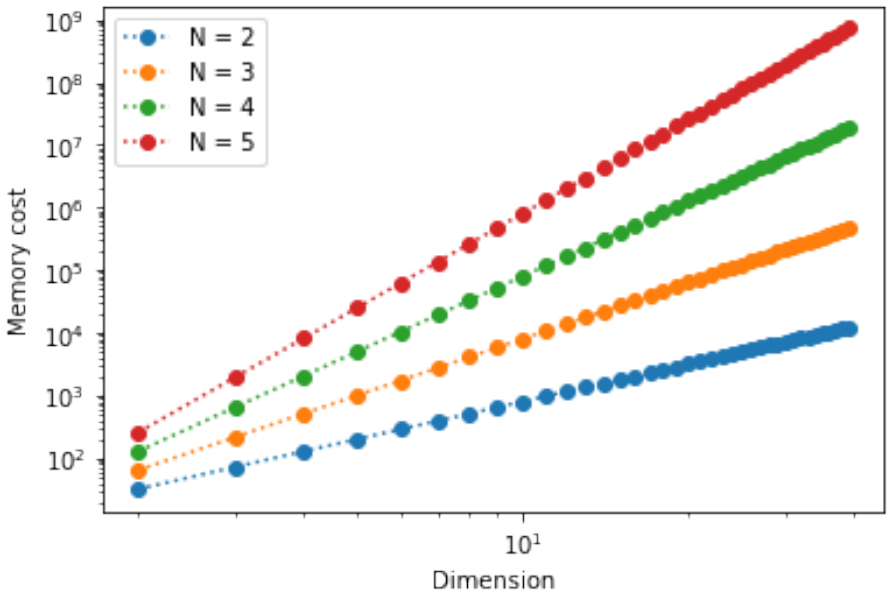


Figure 3.1: Graphic output of the code in listing 3.1 which shows that the memory required to store a tensor scales as d^N. The growth is exponential which quickly saturates one's computational resources

The solution to avoid this problem of dimensionality is the Matrix Product State. The way in which an MPS is created is through an algorithm that uses Singular Value Decomposition (SVD) of a matrix [55, 69]. To exemplify, consider the matrix Ψ with indices n and m and can be written as:

$$\Psi^{nm} = \sum_i L_i^n \sigma_i R_i^m \tag{3.2}$$

In (3.2), L_i^n and R_i^m represent the left and right singular vectors and σ_i represents the singular values. The singular vectors are orthogonal meaning that:

$$\sum_n L_i^n L_j^n = \delta_{ij}$$
$$\sum_n R_i^n R_j^n = \delta_{ij} \tag{3.3}$$

In graphical terms, the equation (3.2) can be represented as:

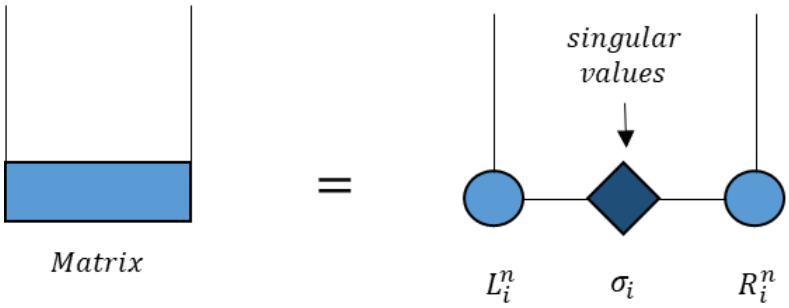


Figure 3.2: A state of two site is "pulled apart"and creates a Matrix Product State in the process.

For a general state of three sites, the process of using SVD is iterative. Starting with the basic general state representation previously shown (Figure 3.3).

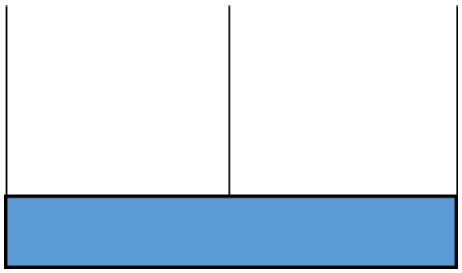


Figure 3.3: Simple General State

The first set of singular value decomposition is made only in one site, separating the state into two parts ("n"and "m"). So the first half would be the site one which is pulled out and the site two and site three are the second half (Figure 3.4(a)).

The next step is to pull apart site 2 and 3, applying the singular value decomposition (3.4(b)). The final result will be a matrix product state with three states and two singular value components (Figure 3.5).

The process of Single Value Decomposition is important for the computation of Matrix Product States [7, 64]. The code shown in Listing 3.2 begins by creating directly the node structure of the MPS. The first function constructs each block of the MPS, constructing a new matrix for the MPS with random numbers from 0 to 1. The second function will contruct the MPS itself.

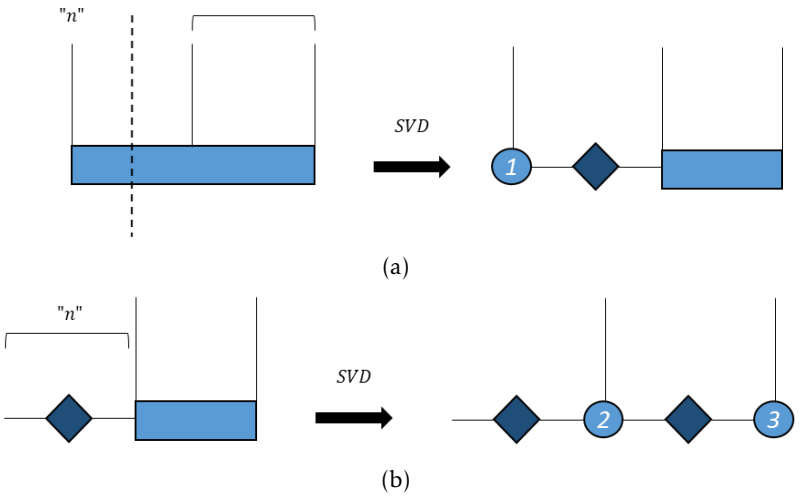


Figure 3.4: Singular Value Decomposition process of pulling apart the sites of the general state

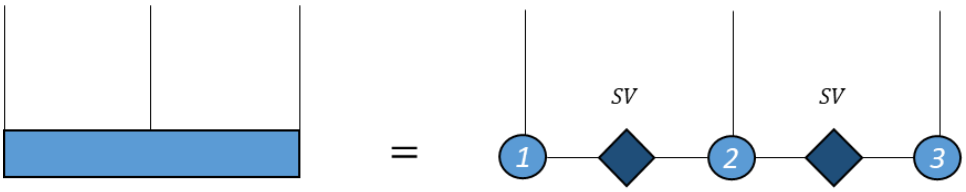


Figure 3.5: The Matrix Product State is represented in what is called the "Vidal form"because it is presented without absorption of the singular values (SV).

```
#In order to create the MPS grid, the first step is to create each block with
    random dimensions and then joining them together in a grid geometry in the
    function create_MPS
def block(*dimensions):
  size = tuple([x for x in dimensions])
  return np.random.random_sample(size)

def create_MPS(rank, dimension, bond_dimension):
  mps = [
      tn.Node( block(dim, bond_dim) )] +
      [tn.Node( block(bond_dim, dim, bond_dim)) for _ in range(rank-2)] +
      [tn.Node( block(bond_dim, dim) )
      ]

  #connect edges to build mps
  connected_edges=[]
  conn=mps[0][1]^mps[1][0]
  connected_edges.append(conn)
  for k in range(1,rank-1):
      conn=mps[k][2]^mps[k+1][0]
      connected_edges.append(conn)

  return mps, connected_edges

```

```python
#Once again doing the plots to compare the dimensions and the memory
dimensions = range(2,9,2)
MPS_ranks = range(2,150)
MPS_memory = []

for dim in dimensions:
    bond_dim = 2
    MPS_memory = []
    for i in range(len(MPS_ranks)):
        rank = MPS_ranks[i]

        mps_nodes, mps_edges = create_MPS(rank, dim, bond_dim)
        MPS_memory.append(np.sum([x.tensor.nbytes for x in mps_nodes]))

    # Plot Results
    plt.loglog(MPS_ranks, MPS_memory, 'o',ls=':', label = f'd = {dim}')

plt.legend()
plt.xlabel('Tensor Rank')
plt.ylabel('MPS memory')

plt.show()
```

Listing 3.2: Python Code of MPS creation

Here, a graphic is created showing the results of calculating the memory size of a Matrix Product States of different dimensions and ranks (Figure 3.6). The results show a difference in the memory requirement: the scaling is now polynomial ($\sim N$), meaning higher physical dimensions can be probed with less memory.

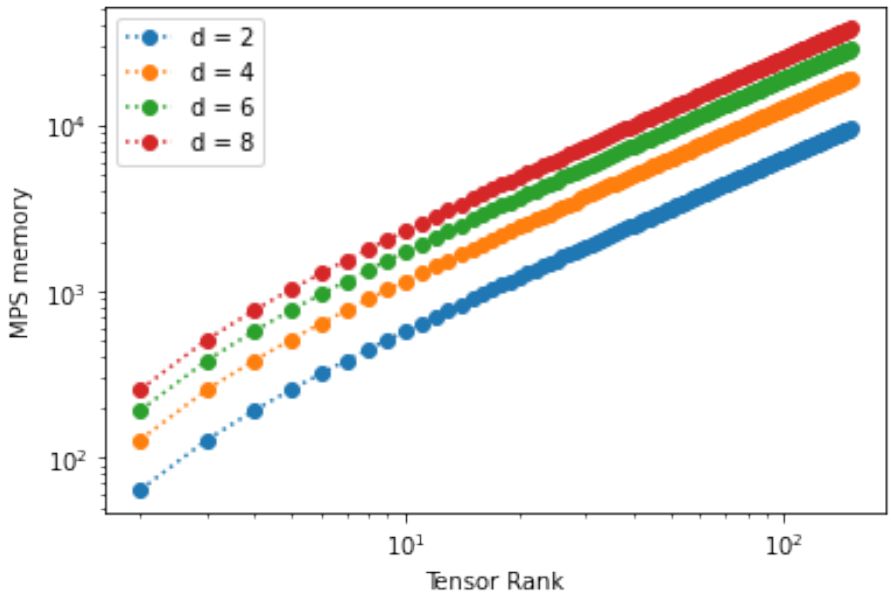


Figure 3.6: Graphical Output of Listing 3.2 showing the potential of Matrix Product State

4

Density Matrix Renormalization Group in Action

4.1 The DMRG algorithm

The tensorial structure of the Hilbert space of a composite system leads to an exponential growth of the resources needed for a simulation with the number of the system constituents [16, 71]. In regard to the number of parameters of a ground state there are two possible scenarios: it can be limited for non critical systems or it can be growing polynomially for critical systems. Therefore it is possible to rewrite the state of the system in a more efficient way, in other words, the system can be described by using a number of coefficients which is much smaller than the dimension of the Hilbert space [23, 87, 21].

In the real space blocking RG procedure, we fisrst begin with a block $\mathscr{B}$ of size L, which is a small piece of the quantum system that is being analysed, and a Hamiltonian. The first one exists on an m-dimensional Hilbert Space andthe second describes how two blocks interact. The next step is to project the composite 2-block system representation of dimension m^2 onto the subspace spanned by the m lowest-lying energy eigenstates, a new truncated representation is obtained. Each operator is consequently projected onto the new m-dimensional basis. The whole process is repeated over and over until the system that was wanted is obtained [16, 15, 78].

4.1.1 Infinite-system DMRG for one-dimensional lattice systems

The objective of DMRG is to search for an approximate ground state of a 1D chain of neighbor interacting sites and does this through an iterative course of action where there is a progressive enlargement of the system. Each of the interacting sites are living in a Hilbert space of Dimension D [59]. The Hamiltonian of the system can be written as:

$$\hat{H} = \sum_i \sum_q J(q)\hat{S}_i(q)\hat{T}_i + 1(q) + \hat{B}(q)\hat{V}_i(q) \tag{4.1}$$

where we have the coupling constants which are represented by $J(q)$ and $B(q)$, and we also have the sets of operators ($\{\hat{S}_i(q)\}_q$, $\{\hat{T}_i(q)\}_q$ and $\{\hat{V}_i(q)\}_q$) which are acting on the i-th

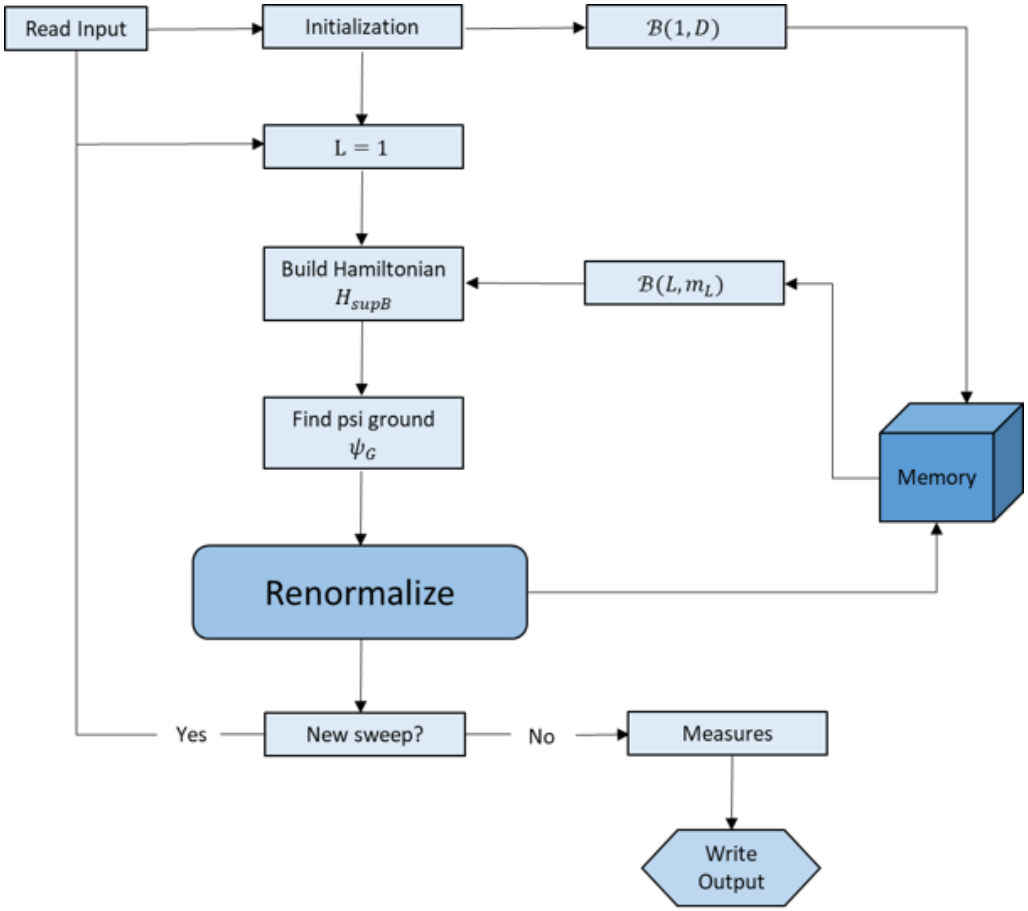


Figure 4.1: A basic representation of the infinite DMRG algorithm. For simplicity it is supposed that system is globally reflection symmetric.

site, the q referring to the various elements of these sets.

The algorithm starts with a *block* composed of one site $\mathcal{B}(1, D)$, where 1 and D represent the number of sites and the number of states used to describe it, respectively. The block Hamiltonian $\hat{H}_B$ for the site $\mathcal{B}(L, m_L)$ includes only the local and interaction terms. The next objective is to construct the *enlarged block*, by adding a site to the right of the block which was created previously. The corresponding Hamiltonian $\hat{H}_E$ is composed by the local Hamiltonians of the block and the site, plus the interaction:

$$\hat{H}_E = \hat{H}_B + \hat{H}_S + \hat{H}_B S. \tag{4.2}$$

A coupling is then made by the enlarged block that was created with another right enlarged block. If the system has global reflection symmetry, then it is possible to just reflect the left enlarged block in order to obtain the right enlarged block Hamiltonian $\hat{H}_{E'}$. A *super block* is then built: $\hat{H}_{supB}$, an Hamiltonian created by the interaction of the two enlarged blocks. $\hat{H}_{supB}$ describes the system on its whole:

$$\hat{H}_{supB} = \hat{H}_E + \hat{H}_{E'} + \hat{H}_{SS'}. \tag{4.3}$$

The sites S and S' are now referred to as the *free sites*. $\hat{H}_{supB}$ is diagonalized in order to find the ground state $\psi_G.\psi_G$ is rewritten in ket notation as:

$$|\psi_G\rangle = \psi_{a\alpha\beta b}|a\alpha\beta b\rangle \tag{4.4}$$

In order to distinguish the blocks and free sites, hereafter they will be referred to by Latin indexes and Greek indexes, respectively. From $|\psi_G\rangle$ one evaluates the reduced density matrix $\hat{\rho}_L$ of the left enlarged block, by tracing out the right enlarged block:

$$\hat{\rho}_L =_R |\psi_G\rangle\langle\psi_G| = \psi_{a\alpha\beta b}\psi_{a'\alpha'\beta b}|a\alpha\rangle\langle a'\alpha'| \tag{4.5}$$

The process of renormalization of the enlarged block consists in finding a representation in terms of a reduced basis with at most m elements (fixed *a priori*)[71, 23]. This whole concept is the what the DMRG algorithm is based upon and it corresponds to a truncation of the Hilbert space of the enlarged block, since $m_{L+1} = min(m_L D, m)$. These states are chosen to be the first m_{L+1} eigenstates of ρ_L, which correspond to the largest eigenvalues. This truncated change of basis is performed by using the $m_L D \times m_{L+1}$ rectangular matrix $\hat{O}_{L\to L+1}$ (where the subscripts stand for the number of sites enclosed in the input block and in the output renormalized block), whose columns, in matrix representation, are the m_{L+1} eigenstates. The function $g(a,\alpha) = D(a-1)+\alpha$ is introduced to simplify notations. This function acts on a block index a and on the next free site index α and gives an index of the enlarged block running from 1 to $m_L D$.

A truncated enlarged block ($\mathcal{B}(L+1, m_{L+1})$) is the outcome of the whole renormalization process. As said before DMRG is an iterative process so this enlarged block coincides with the new block which will start the next iteration. This consists in the new block Hamiltonian:

$$\begin{aligned}\hat{H'}_B &= \hat{O}^{\dagger}_{L\to L+1}\hat{H}_E\hat{O}_{L\to L+1} \\ &= O^{*g(a,\alpha)c}_{L\to L+1}H_E^{g(a,\alpha)g(a',\alpha')}O^{g(a',\alpha')c'}_{L\to L+1}|c\rangle\langle c'|\end{aligned} \tag{4.6}$$

and in the local operators:

$$\hat{S}'_{L+1}(q) = \hat{O}^{\dagger}_{L\to L+1}(q)\hat{O}_{L\to L+1} \tag{4.7}$$

written in the new basis and necessary for the next step which is the construction of the interaction between the block site most to the right and the free site. The output block $\mathcal{B}(L+1, m_{L+1})$ includes also the matrix $\hat{O}_{L\to L+1}$ which identifies the basis states of the new block.

4.1.2 DMRG of MPS

In the context of an MPS, the DMRG algorithm works by optimizing two neighboring MPS tensors at a time, combining them into a single tensor to be optimized [72, 58]. The optimization is performed using an iterative eigensolver approach. Before the next step the single tensor is factorized using an Single Value Decomposition (SVD) or density matrix decomposition in order to restore the MPS form.

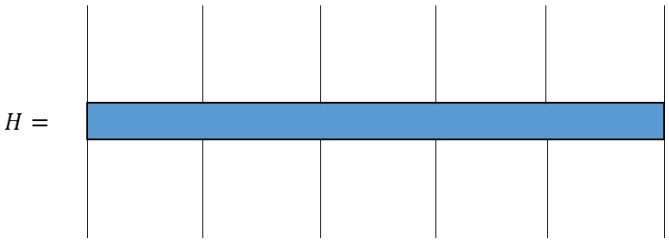


Figure 4.2: Hermitian Matrix H

To tackle the problem one considers a Hermitian Matrix H (Figure 4.2) acting in vector space that is a tensor product of N smaller spaces, each of dimension d. The algorithm seeks the dominant eigenvector of H in the form of an MPS tensor network, like the one in Figure 4.3, where E_0 is the minimum eigenvalue of H:

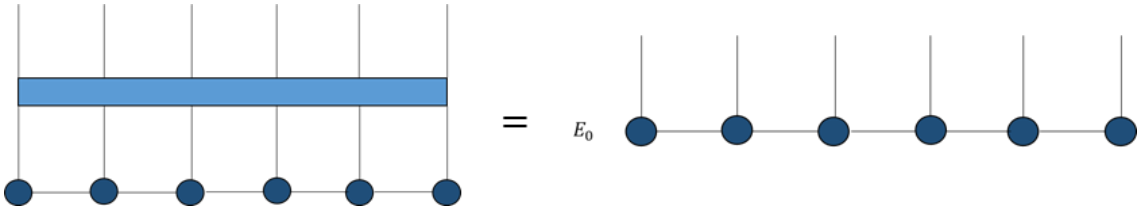


Figure 4.3: The minimum eigenvalue of H is $E_0(E_1 \leq E_2...)$

The Hermitian Matrix H can be represented as a MPO tensor network form which is the most natural one for the DMRG algorithm, and can efficiently represent various situations that are of interest, like H being a sum of local terms.

Following the process of Singular Value Decomposition in Matrix Product States which can result in an MPS with enormous bond dimension the next step is to shrink down this dimension. One solution is to write the MPS in the particular gauge where the singular values appear (Figure 4.4) and take those singular values and throw away the small ones. This way it is possible to end up with a smaller MPS that only takes a subset of the singular which is equivalent to truncating the singular values. This is known as SVD compression: each site is compressed independently of new information of other sites, it is a "local update". This is not the optimal solution because the process of throwing away singular values is done independently of each other.

A better way to compress an MPS is to solve the optimization problem (4.8) where the objective is to minimize the difference between the initial MPS with large bond dimension

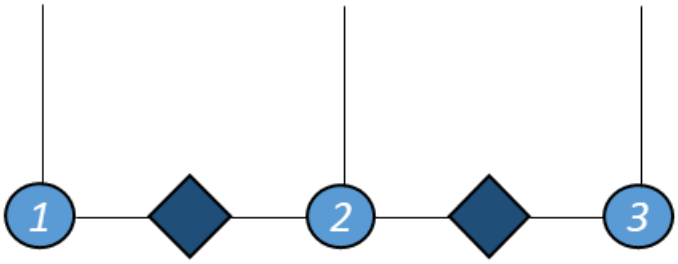
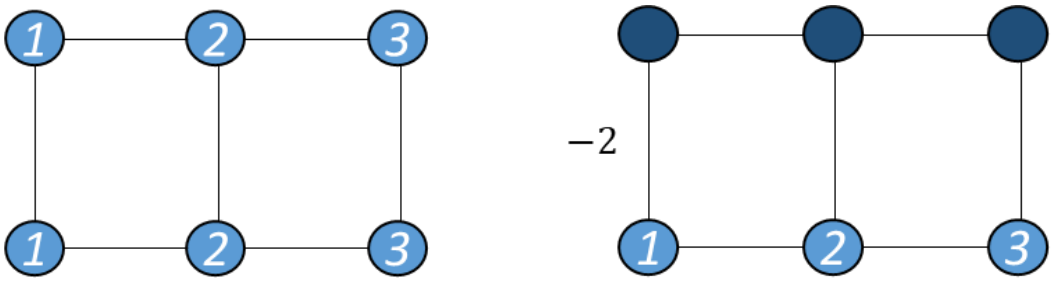


Figure 4.4: MPS in Vidal form: no absorption of the singular values

Ψ and a good approximation with smaller bond dimension Φ.

$$min_{\Phi} \langle \Psi - \Phi | \Psi - \Phi \rangle = min_{\Phi} [-2 \langle \Psi | \Phi \rangle + \langle \Phi | \Phi \rangle] \tag{4.8}$$

The solution is to minimize the sum of a linear term and a quadratic term (Figure 4.5).



(a) (b)

Figure 4.5: Terms of the solution of the optimization problem. (a) Graphical representation of the quadratic term $-2\langle \Psi | \Phi \rangle$. (b) Graphical representation of the linear term $\langle \Phi | \Phi \rangle$

The way to minimize something numerically is to evaluate the gradient of the system, i.e., the derivative with respect to the elements of the system, then follow the gradient until it vanishes.

In this case, the procedure is to differentiate the solution of 4.8 with respect to the elements of the matrices in Figure 4.5 to find the gradient. The result of differentiating the terms of the solution with respect to an element and adding them is a tensor with three indices $g[nij]$. The tensor that was optimizing (the element chosen to differentiate) is also a tensor with three indices $A^n_{ij} = a[nij]$. The gradient algorithm ultimately would be written as:

$$a[nij] \rightarrow a[nij] + \epsilon g[nij] \tag{4.9}$$

This equation is what's called the gradient step and doing this continuously will eventually go to zero and the compression would be optimized.

The DMRG is based on the observation that explicitly evaluating this gradient and following it is actually not necessary if one works in the mixed canonical form.

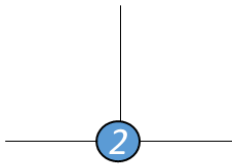


Figure 4.6: Vector with three indices

So considering the object in Figure 4.6 as a vector $A^n_{ij} = a[nij]$, the relationships in Figure 4.7 can be assumed: in the first, the linear object, there is a product between two vectors (Figure 4.7(a)) and for the quadratic object there is a product between a vector, a matrix and a vector (Figure 4.7(b)).

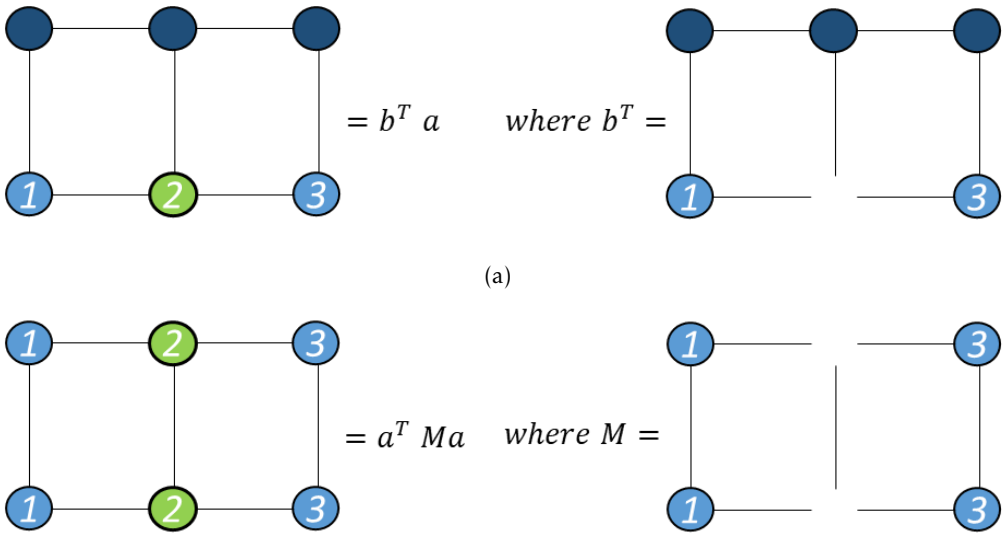


Figure 4.7: DMRG Sweep: (a) Linear Object. (b) Quadratic Object

The minimization of the sum of the terms with respect to the elements of 2 (Figure 4.6), the vector a, is just the minimization of a quadratic problem (4.10) which is just linear algebra.

$$min_a(a^T Ma - b^T a) \rightarrow Ma = b \tag{4.10}$$

In essence, the DMRG observes that in a Matrix Product State everything occurs linearly so all the operators one does boil down to linear algebra. The minimization is performed site by site (Figure 4.8) using updated vectors from previous steps solving $Ma = b$ one site at a time.

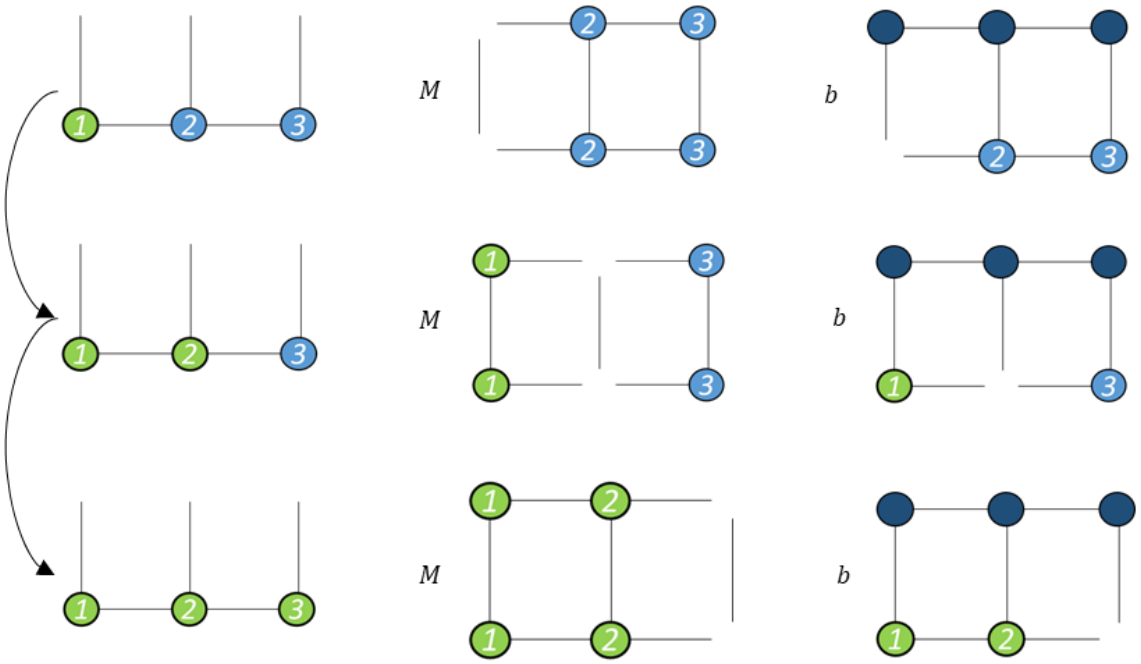


Figure 4.8: DMRG sweep algorithm in action.

4.2 DMRG of a text

The choice to use a text as the object of analysis in this dissertation is due to its simplicity and easy accessibility. The same method and programming could be used to study the behaviour of particles in space, but for that it would be necessary to have a particle accelerator. Since the math is the same and both are examples of abstract universes with correlation between objects, the easiest one to work with was chosen.

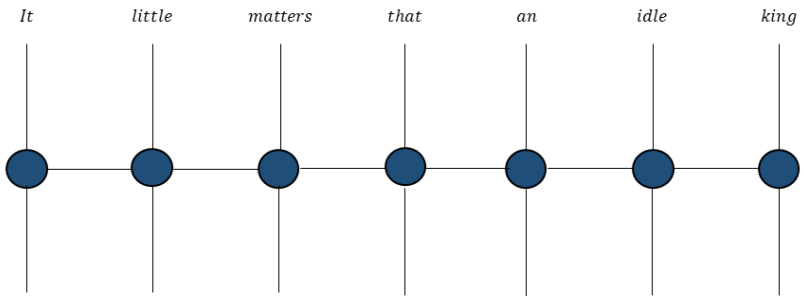


Figure 4.9: Matrix Product Operator representation of the text. The chosen text was the poem "Ulysses"from Alfred Lord Tennyson.

When approaching the programming of the DMRG of the text it was considered that each word was a particle, specifically a full spin particle (basically it is represented by a "1"). The nature of the word itself is irrelevant, whether its a noun, a verb or whether it is a common word or not, what counts in this case is the connections between the words in the text, and that is what the DMRG algorithm is going to be working with. Therefore in

diagram form, words will be represented as "1" and the spaces between the words will be "0".

There are no repeated words: N is the number of distinct words that a text has and v is the dimension of the vocabulary of said text, such so that the group of words in the text can be written as:

$$|i_1, i_2, ...i_v\rangle \tag{4.11}$$

The number of words is then transformed in to spins, becoming sites and an MPO is created with them.

After this, a Next Neighbor Heisenberg Model is created [34]. The DMRG code works with Hamiltonians in matrix product operator form so that a single DMRG code can work for a very wide range of Hamiltonians and local Hilbert spaces [41].

```
auto sites = SpinOne(N);

auto ampo = AutoMPO(sites);
for(int j = 1; j < N; ++j)
    {
    ampo += "Sz",j,"Sz",j+1;
    ampo += 0.5,"S+",j,"S-",j+1;
    ampo += 0.5,"S-",j,"S+",j+1;
    }
auto H = toMPO(ampo);

auto sweeps = Sweeps(5);
sweeps.maxdim() = 10,40,100,200,200;
sweeps.cutoff() = 1E-8;

// Create a random starting state
auto state = InitState(sites);
for(auto i : range1(N))
    {
    if(i%2 == 1) state.set(i,"Up");
    else         state.set(i,"Dn");
    }
auto psi0 = randomMPS(state);

auto [energy,psi] = dmrg(H,psi0,sweeps,{"Quiet",true});

printfln("Ground state energy = %.20f",energy);
```

Listing 4.1: ITensor Program for DMRG creation in C. The auto sites variable created through SpinOne function uses a value N which the number of distinct words the text analyzed.

The *AutoMPO* function is a very powerful system for translating sums of local operators into an MPO tensor network [40]. The "+=" operator of an *AutoMPO* object adds a operator product into the sum of operators represented by the *AutoMPO*. The operator

products on the right-hand side of the "+=" operator begin with an optional real- or complex-valued coefficient, then continue with a comma separated list of string-integer pairs. The string-integer pairs, such as ["Sz",j], represent an operator S^z_j. The *toMPO* function is called to create an MPO from an AutoMPO.

The output is a report of each sweep done and the bond dimension values of each state along with the truncation error which was always very minimal. The final output after all the sweeps is the value of the Ground State Energy.

```
Sweep=5, HS=2, Bond=3/568
I 0 q 1E-05 E -209.0142879790
I 2 q 5E-07 E -209.0142879790
        Truncated to Cutoff=1.0E-10, Min_dim=1, Max_dim=200
        Trunc. err=5.0E-16, States kept: dim=27
```

Figure 4.10: Example of an output for each bond that the DMRG sweeps through. In this example the program does 5 sweeps through the whole system.

The output example presented in Figure 4.10 is just one out of the hundreds that come out when the DRMG program sweeps through the system and outputs the bond values for each side of the word that is being swept through. The first line tracks the sweep and the bond. The next two give the value of the bond dimension and their energy. The DMRG process implemented uses singular value decomposition in order to create a compressed representation.

As mentioned before the philosophy for this process is to consider each word as a particle, a "spin up", which might counter intuitive since it disregards the nature of each word. An initial approach was made considering the nature of each word called "tokenization".

The tokenization process was essentially the substitution of each word an ID where each number would correspond to a certain characteristic of said word (Figure 4.11(a)), for example, the nature of the word: wether it is a noun, a verb, an adverb, or the position in the text, wether it is the first word, second or last. Adding up all this characteristic the final product would be that each word had a number sequence to represent it, in other words, a *token* and the next step would be turn each into a tensor and join all those tensors to form an MPS, similar to the process made with the *tensornetwork* python library in chapter 3. This approach was deemed too complicated and very time consuming, the program took a long time to create all the tensors and in the end it was considered to be against the philosophy of Tensor Networks. The objective is to analyse the a system in a more simplified way and to do it while giving a higher degree importance to the relations between the components. So a new approach was used (Figure 4.11(b)) where each was essentially considered to be a particle with spin up.

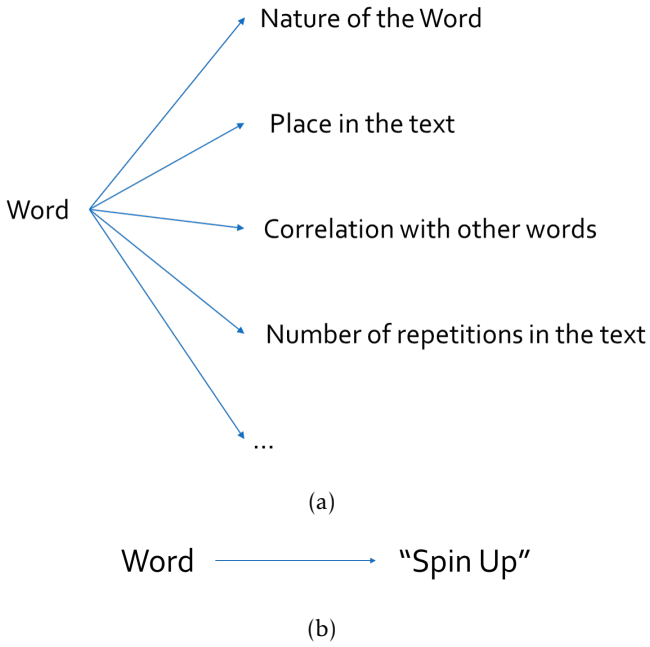


Figure 4.11: Representation of the approaches considered when programming the DMRG of a text: (a)*Tonkenization* of each word; (b) Considering all words as particles

In DMRG, the reality of an increase of the size of the system does not necessarily mean there is the need to increase the bond dimension, which is something that sounds logical *a priori*. The infinite DMRG is an example: it minimizes the energy density over translationally invariant MPS directly in the thermodynamic limit. What is observed then is that to compute expectation values of local observables, a constant bond dimension is all that it is neccessary. A precedent to this is Hastings work [35] where he proved that there exists an MPS with polynomial (in the system size) bond dimension such that the overlap with the true ground state is close to unity.

```
            Largest link dim during sweep 5/5 was 2000
            Largest truncation error: 0.0639051
            Energy after sweep 5/5 is -209.014287978959
            Sweep5/5 CPU time = 45,04s (Wall time = 1m, 3.4s)
Ground State Energy = -209.014
```

Figure 4.12: Final output of the DMRG code which serves as a conclusion and final report to the sweep

The last result given by the DMRG codes (Figure 4.12) is the most important; among the information given in it regarding the whole process of sweeping through the entire system there is the most essential value: the Ground State Energy. This value is what makes this whole process and the Tensor Networks relevant to the discussion of analyzing systems efficiently because, given the mathematical proof demonstrated in Chapter 3.7,

which states that it is possible to understand the energy of a system using only the ground state energy, it automatically gives a valid argument to tensor networks and DMRG, in this particular case (Figure 4.13).

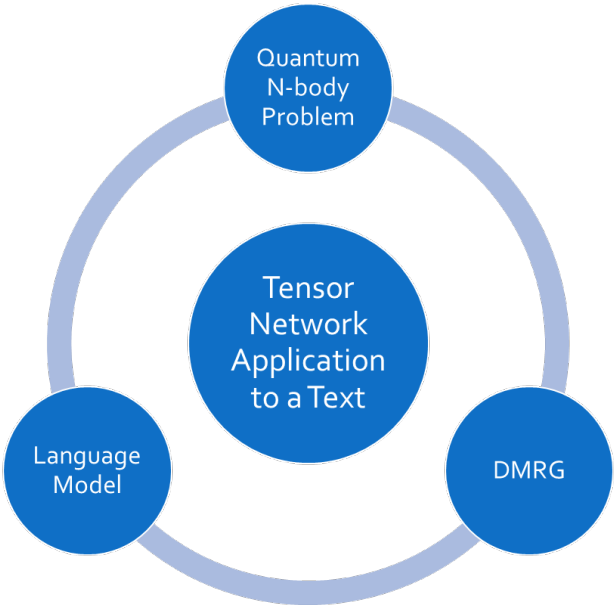


Figure 4.13:

The DMRG has processed all the connections between each word more than once and analyzed the behaviour of the system, not only as a sum of its components, but considering also their interactions without having to grind through all the different possible configurations and rearrangements that the system could take, which made the process fast and, one even might consider, more instinctive.

5

Final Remarks and the Future of Tensor Networks

The work presented in this thesis does not aim to produce an argument against the current forms of system analysis, it never compares its results with neural networks or machine learning because its aim is to study the application of the tensor networks, showing its inner workings and the concept behind it, as well as its elegance in the way that it approaches a problem. In regards to the programming of its application, other routes were possible and considered: an early proposal was to use already existing python libraries to manually create a tensor out of each word, similar to a "word2vec" code where each word goes through a process of *tokenization* and in the end is represented by a group of variables, akin to an ID, where each corresponds to the nature of the word, its place on the text and its relation to other words. In the end, the process was time consuming and didn't yield sufficient results, and was not considered not very fruitful given the conclusion reached in Chapter 4, which states that the nature of the word itself can be considered irrelevant and the whole system can be viewed as just particles and the relations between them. Consequently, the elegance of tensor networks, is in the attempt to perceive a system in the same way human brains do. After all, when we speak or write a sentence, after thinking of the first word we don't consider all the words we know when choosing the second word, which is why it can be considered counterproductive to contemplate all the different possibilities and relations when trying to analyze a system. This is why Tensor Networks are being used by scientists who are trying to create the Holographic Universe Theory.

5.1 Tensor Networks and the future of Computing

The increasing amounts of information and need for a higher level of computation have led to a fast evolution in the software and hardware worlds, meaning that technology has been evolving and growing in tremendous speed and some experts predict that this speed can lead to a halt in our power to create faster circuits, especially if the MOSFET scaling predicted by Moore continues to hold, which has since 1975 [70, 52].

With this possible scenario in sight, many scientists started thinking about the next step. One of this solutions was quantum computation, which uses the collective properties of quantum states, such as superposition, interference and entanglement, to perform calculations [75]. As previously stated, when referencing Moore's Law, since the 1960's the power of our brain machines has kept growing exponentially, allowing computers to get smaller and more powerful at the same time. However, this process is about to meet its physical limits because computer parts are approaching the size of an atom.

A transistor, the simplest form of a data processor in computers, is basically a switch that can either block, or open the way for information coming through. This information is made up of bits (0 or 1) and the transistor functions as an electric switch that can block electrons from moving from one direction to another. As transistors are shrinking to the size of only a few atoms, electrons may just transfer themselves to the other side of a blocked passage via a process called quantum tunneling [37, 38]. This is the physical barrier that can stagnate the technological progress.

In quantum computers, *qubits* would replace the bits as the smallest unit of information. A *qubit* can be any two level quantum system; in the quantum world the *qubit* doesn't have to be in just one the system's possible states. It can be in any proportion of both states at once, this is called a superposition. For example, for a classical system, 4 bits can be in one of two to the power of four different configurations at a time, that is, 16 possible combinations, only one of which can be used. Four *qubits* in superposition, however, can be in all of those 16 combinations at once. A quantum computer sets up some *qubits*, applies quantum gates to entangle them and manipulate probabilities and finally measures the outcome, collapsing superpositions to an actual sequence of 0s and 1s. (Figure 5.1)

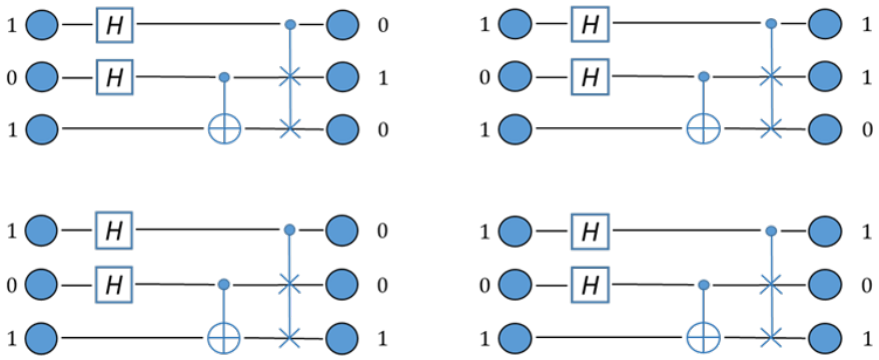


Figure 5.1: Example of a quantum circuit with three qubits of input. With a setup like this one can get the entire lot of calculations, all done at the same time.

Though it is a very clever solution to a problem that is inevitable in the future, right now the possibility of using quantum computers is still very limited due to the difficulties in their construction because of the materials needed to build one as well as other problems such as quantum decoherence [68]. The problem of the upcoming barrier in

the technological barrier is once again diagnosed with the illusion of complexity, deceiving it. If one takes the same approach of "Nature doesn't explore all possibilities" then the obstacle might not come as soon as it is previewed. This is where Tensor Networks can come as the alternative for the way computation is made and how the systems are perceived, delaying the immediate need for quantum computers. As the work presented in this thesis has proven, tensor network renormalization can act as a very powerful tool in the understanding of complex systems which can attenuate the necessity for high computational power.

Bibliography

[1] P. W. Anderson. "More is different: broken symmetry and the nature of the hierarchical structure of science." In: *Science* 177.4047 (1972), pp. 393–396.

[2] S. Approach. *Algorithms for Quantum Renormalization Groups*. 1993.

[3] A. Barenco et al. "Elementary gates for quantum information". In: *Phys. Rev. A* 52.5 (1995), pp. 3457–3467.

[4] F. Battaglia and T. George. "Tensors: A guide for undergraduate students". In: *American Journal of Physics* 81 (July 2013), pp. 498–511. DOI: 10.1119/1.4802811.

[5] J. Bedford. *An Introduction to String Theory*. 2011. DOI: 10.48550/ARXIV.1107.3967. URL: https://arxiv.org/abs/1107.3967.

[6] R. Bellman. *Dynamic Programming*. Dover Publications, 1957. ISBN: 9780486428093.

[7] J. A. Bengua. "Matrix product state decomposition in machine learning and signal processing". PhD thesis. 2016.

[8] J. Biamonte. "Lectures on Quantum Tensor Networks". In: (2019). arXiv: 1912.10049. URL: http://arxiv.org/abs/1912.10049.

[9] J. Biamonte, V. Bergholm, and M. Lanzagorta. "Tensor network methods for invariant theory". In: *Journal of Physics A: Mathematical and Theoretical* 46.47 (2013). ISSN: 17518113. DOI: 10.1088/1751-8113/46/47/475301. arXiv: 1209.0631.

[10] E. Brézin. "Wilson's Renormalization Group: A Paradigmatic Shift". In: *Journal of Statistical Physics* 157.4-5 (Mar. 2014), pp. 644–650. DOI: 10.1007/s10955-014-0971-5. URL: https://doi.org/10.1007%2Fs10955-014-0971-5.

[11] J. C. Bridgeman and C. T. Chubb. "Hand-waving and interpretive dance: an introductory course on tensor networks". In: *Journal of Physics A: Mathematical and Theoretical* 50.22 (May 2017), p. 223001. DOI: 10.1088/1751-8121/aa6dc3. URL: https://doi.org/10.1088/1751-8121/aa6dc3.

[12] F. Buisseret. "QuantumN-body problem with a minimal length". In: *Physical Review A* 82.6 (Dec. 2010). DOI: 10.1103/physreva.82.062102. URL: https://doi.org/10.1103%2Fphysreva.82.062102.

[13] L. Cammoun et al. "A Review of Tensors and Tensor Signal Processing". In: *Tensors in Image Processing and Computer Vision*. Ed. by S. Aja-Fernández et al. London: Springer London, 2009, pp. 1–32. ISBN: 978-1-84882-299-3. DOI: 10.1007/978-1-84882-299-3_1. URL: https://doi.org/10.1007/978-1-84882-299-3_1.

[14] G. Castagna. "Covariance and Controvariance: a fresh look at an old issue (a primer in advanced type systems for learning functional programmers)". In: *CoRR* abs/1809.01427 (2018). arXiv: 1809.01427. URL: http://arxiv.org/abs/1809.01427.

[15] G. K.-L. Chan and M. Head-Gordon. "Highly correlated calculations with a polynomial cost algorithm: A study of the density matrix renormalization group". In: *The Journal of chemical physics* 116.11 (2002), pp. 4462–4476.

[16] G. D. Chiara et al. "Density Matrix Renormalization Group for Dummies". In: *Journal of Computational and Theoretical Nanoscience* 5.7 (July 2008), pp. 1277–1288. DOI: 10.1166/jctn.2008.2564. URL: https://doi.org/10.1166%2Fjctn.2008.2564.

[17] N. Chomsky, Á. J. Gallego, and D. Ott. "Generative grammar and the faculty of language: Insights, questions, and challenges". In: *Catalan Journal of Linguistics* (2019), pp. 229–261.

[18] J. I. Cirac et al. "Matrix product states and projected entangled pair states: Concepts, symmetries, theorems". In: *Reviews of Modern Physics* 93.4 (Dec. 2021). DOI: 10.1103/revmodphys.93.045003. URL: https://doi.org/10.1103%2Frevmodphys.93.045003.

[19] P. Comon. "Tensors: a Brief Introduction". In: *Signal Processing Magazine, IEEE* 31 (May 2014), pp. 44–53. DOI: 10.1109/MSP.2014.2298533.

[20] A. Critch and J. Morton. "Algebraic geometry of matrix product states". In: *Symmetry, Integrability and Geometry: Methods and Applications (SIGMA)* 10 (2014). ISSN: 18150659. DOI: 10.3842/SIGMA.2014.095. arXiv: 1210.2812.

[21] E. F. D'Azevedo et al. *Targeting Multiple States in the Density Matrix Renormalization Group with The Singular Value Decomposition*. 2019. DOI: 10.48550/ARXIV.1902.09621. URL: https://arxiv.org/abs/1902.09621.

[22] L. Debnath. "A brief historical introduction to matrices and their applications". In: *International Journal of Mathematical Education in Science and Technology* 45.3 (2014), pp. 360–377. DOI: 10.1080/0020739X.2013.837521. eprint: https://doi.org/10.1080/0020739X.2013.837521. URL: https://doi.org/10.1080/0020739X.2013.837521.

[23] *Density Matrix Renormalization Group*. https://tensornetwork.org/mps/algorithms/dmrg/. Accessed: 2021-09-30.

[24] D. Deutsch and P. R. S. L. A. "Quantum computational networks". In: *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 425.1868 (1989), pp. 73–90. ISSN: 2053-9169. DOI: 10.1098/rspa.1989.0099.

[25] P. Du, S. W. Rick, and R. Kumar. "Towards a coarse-grained model of the peptoid backbone: The case of N, N -dimethylacetamide". In: *Physical Chemistry Chemical Physics* 20.36 (2018), pp. 23386–23396. ISSN: 14639076. DOI: 10.1039/c8cp03283a.

[26] E. Efrati et al. "Real-space renormalization in statistical mechanics". In: *Reviews of Modern Physics* 86.2 (May 2014), pp. 647–667. ISSN: 1539-0756. DOI: 10.1103/revmodphys.86.647. URL: http://dx.doi.org/10.1103/RevModPhys.86.647.

[27] G. Esposito. *An introduction to quantum gravity*. 2011. DOI: 10.48550/ARXIV.1108.3269. URL: https://arxiv.org/abs/1108.3269.

[28] G. Evenbly and G. Vidal. "Tensor Network Renormalization". In: *Physical Review Letters* 115.18 (2015), pp. 1–6. ISSN: 10797114. DOI: 10.1103/PhysRevLett.115.180405. arXiv: 1412.0732.

[29] G. Evenbly and G. Vidal. "Tensor Network Renormalization Yields the Multiscale Entanglement Renormalization Ansatz". In: *Physical Review Letters* 115.20 (2015), pp. 1–11. ISSN: 10797114. DOI: 10.1103/PhysRevLett.115.200401. arXiv: 1502.05385.

[30] M. Fannes, B. Nachtergaele, and R. Werner. "Finitely correlated states on quantum spin chains". In: *Communications in Mathematical Physics* 144 (Mar. 1992). DOI: 10.1007/BF02099178.

[31] Á. Gallego and R. Orus. "Language Design as Information Renormalization". In: *SN Computer Science* 3 (Mar. 2022). DOI: 10.1007/s42979-021-01002-y.

[32] R. B. Griffiths. "Quantum Locality". In: *Foundations of Physics* 41.4 (Oct. 2010), pp. 705–733. DOI: 10.1007/s10701-010-9512-5. URL: https://doi.org/10.1007%2Fs10701-010-9512-5.

[33] P. Grinfeld. *Introduction to Tensor Analysis and the Calculus of Moving Surfaces by Pavel Grinfeld.* eng. 1st ed. 2013. New York, NY: Springer New York : Imprint: Springer, 2013. ISBN: 9781461478669.

[34] S.-J. Gu et al. "Entanglement of the Heisenberg chain with the next-nearest-neighbor interaction". In: *Phys. Rev. A* 70 (5 Nov. 2004), p. 052302. DOI: 10.1103/PhysRevA.70.052302. URL: https://link.aps.org/doi/10.1103/PhysRevA.70.052302.

[35] M. B. Hastings. "An area law for one-dimensional quantum systems". In: *Journal of Statistical Mechanics: Theory and Experiment* 2007.08 (Aug. 2007), P08024–P08024. DOI: 10.1088/1742-5468/2007/08/p08024. URL: https://doi.org/10.1088%2F1742-5468%2F2007%2F08%2Fp08024.

[36] M. Hayashi et al. "Quantum Entanglement". In: *Introduction to Quantum Information Science*. Springer, 2015, pp. 167–203.

[37] T. Hey. "Quantum computing: an introduction". In: *Computing & Control Engineering Journal* 10.3 (1999), pp. 105–112.

[38] M. Hirvensalo. *Quantum computing*. Springer Science & Business Media, 2003.

[39] W. Hunziker and I. M. Sigal. "The quantum N-body problem". In: *Journal of Mathematical Physics* 41 (Mar. 2001). DOI: 10.1063/1.533319.

[40] ITensor. *MS Windows NT AutoMPO*. 2021. URL: https://itensor.org/docs.cgi?vers=cppv3&page=classes/autompo (visited on 10/20/2021).

[41] ITensor. *MS Windows NT DMRG Interface*. 2021. URL: https://itensor.org/docs.cgi?vers=cppv3&page=classes/dmrg (visited on 10/20/2021).

[42] D. Jaschke, M. L. Wall, and L. D. Carr. "Open source Matrix Product States: Opening ways to simulate entangled many-body quantum systems in one dimension". In: *Computer Physics Communications* 225 (Apr. 2018), pp. 59–91. DOI: 10.1016/j.cpc.2017.12.015. URL: https://doi.org/10.1016%2Fj.cpc.2017.12.015.

[43] L. P. Kadanoff. "Scaling laws for ising models near T_c". In: *Physics Physique Fizika* 2 (6 June 1966), pp. 263–272. DOI: 10.1103/PhysicsPhysiqueFizika.2.263. URL: https://link.aps.org/doi/10.1103/PhysicsPhysiqueFizika.2.263.

[44] J. R. Klauder. *Quantum Gravity Made Easy*. 2019. DOI: 10.48550/ARXIV.1903.11211. URL: https://arxiv.org/abs/1903.11211.

[45] J. C. Kolecki. "An Introduction to Tensors for Students of Physics and Engineering". In: *Unixenguaedu* 7.September (2002), p. 29. ISSN: 18733514. URL: http://unix.eng.ua.edu/%7B~%7Dcheclass/che304/che304/%7B%5C_%7DMedia/tensors%7B%5C_%7Dtm2002211716.pdf.

[46] M. Krenn et al. "Generation and confirmation of a (100 × 100)-dimensional entangled quantum system". In: *Proceedings of the National Academy of Sciences* 111.17 (2014), pp. 6243–6247. DOI: 10.1073/pnas.1402365111. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.1402365111. URL: https://www.pnas.org/doi/abs/10.1073/pnas.1402365111.

[47] G. C. Krizek. *Einstein's 1935 papers: EPR=ER?* 2017. DOI: 10.48550/ARXIV.1704.04648. URL: https://arxiv.org/abs/1704.04648.

[48] R. Laughlin and D. Pines. "The Theory of Everything". In: *Proc. Natl. Acad. Sci. USA* 97 (Jan. 2000), pp. 28–. DOI: 10.7551/mitpress/9780262026215.003.0017.

[49] M. Levin and C. P. Nave. "Tensor renormalization group approach to two-dimensional classical lattice models". In: *Physical Review Letters* 99.12 (2007), pp. 1–4. ISSN: 00319007. DOI: 10.1103/PhysRevLett.99.120601. arXiv: 0611687v2 [arXiv:cond-mat].

[50] J. M. Lipski. In: *Language* 73.4 (1997), pp. 874–877. ISSN: 00978507, 15350665. URL: http://www.jstor.org/stable/417344.

[51] L. Lootens et al. "Matrix product operator symmetries and intertwiners in string-nets with domain walls". In: *SciPost Physics* 10.3 (Mar. 2021). DOI: 10.21468/scipostphys.10.3.053. URL: https://doi.org/10.21468%2Fscipostphys.10.3.053.

[52] C. A. Mack. "Fifty years of Moore's law". In: *IEEE Transactions on semiconductor manufacturing* 24.2 (2011), pp. 202–207.

[53] J. Maldacena. "Eternal black holes in anti-de Sitter". In: *Journal of High Energy Physics* 7.4 (2003), pp. 453–469. ISSN: 10298479. DOI: 10.1088/1126-6708/2003/04/021.

[54] *Matrix Product Operator*. https://tensornetwork.org/mpo/. Accessed: 2021-09-30.

[55] *Matrix Product State / Tensor Train*. https://tensornetwork.org/mps/. Accessed: 2021-09-30.

[56] *Matrix Product State Python*. https://tensornetwork.readthedocs.io/en/latest/basic_mps.html. Accessed: 2021-09-30.

[57] H. Matsueda, M. Ishihara, and Y. Hashizume. "Tensor network and a black hole". In: *Physical Review D - Particles, Fields, Gravitation and Cosmology* 87.6 (2013), pp. 1–7. ISSN: 15507998. DOI: 10.1103/PhysRevD.87.066002. arXiv: 1208.0206.

[58] I. P. McCulloch. "From density-matrix renormalization group to matrix product states". In: *Journal of Statistical Mechanics: Theory and Experiment* 2007.10 (2007), P10014.

[59] I. P. McCulloch. "Infinite size density matrix renormalization group, revisited". In: *arXiv preprint arXiv:0804.2509* (2008).

[60] W. Miller, A. V. Turbiner, and M. A. Escobar-Ruiz. "The quantumin/i-body problem in dimensionid/i⩾in/i– 1: ground state". In: *Journal of Physics A: Mathematical and Theoretical* 51.20 (Apr. 2018), p. 205201. DOI: 10.1088/1751-8121/aabb10. URL: https://doi.org/10.1088%2F1751-8121%2Faabb10.

[61] N. Nakatani and G. K.-L. Chan. "Efficient tree tensor network states (TTNS) for quantum chemistry: Generalizations of the density matrix renormalization group algorithm". In: *The Journal of Chemical Physics* 138.13 (Apr. 2013), p. 134113. DOI: 10.1063/1.4798639. URL: https://doi.org/10.1063%2F1.4798639.

[62] R. Orús. "Tensor networks for complex quantum systems". In: *Nature Reviews Physics* 1.9 (Aug. 2019), pp. 538–550. ISSN: 2522-5820. DOI: 10.1038/s42254-019-0086-7. URL: http://dx.doi.org/10.1038/s42254-019-0086-7.

[63] I. V. Oseledets. "Tensor-Train Decomposition". In: *SIAM Journal on Scientific Computing* 33.5 (2011), pp. 2295–2317. DOI: 10.1137/090752286. eprint: https://doi.org/10.1137/090752286. URL: https://doi.org/10.1137/090752286.

[64] S. Paeckel et al. "Time-evolution methods for matrix-product states". In: *Annals of Physics* 411 (Oct. 2019), p. 167998. DOI: 10.1016/j.aop.2019.167998.

[65] R. Penrose. *Combinatorial mathematics and its applications*. Oxford: Academic Press; 0 edition, 1971.

[66] D. Perez-Garcia et al. "Matrix Product State Representations". In: (2006). DOI: 10.48550/ARXIV.QUANT-PH/0608197. URL: https://arxiv.org/abs/quant-ph/0608197.

[67] B. Pirvu et al. "Matrix product operator representations". In: *New Journal of Physics* 12.2 (Feb. 2010), p. 025012. DOI: 10.1088/1367-2630/12/2/025012. URL: https://doi.org/10.1088%2F1367-2630%2F12%2F2%2F025012.

[68] S. Robinson. "Emerging insights on limitations of quantum computing shape quest for fast algorithms". In: *SIAM News* 36.1 (2003), pp. 1–3.

[69] H. Saberi. "Matrix-product states for strongly correlated systems and quantum information processing". In: 2008.

[70] R. R. Schaller. "Moore's law: past, present and future". In: *IEEE spectrum* 34.6 (1997), pp. 52–59.

[71] U. Schollwöck. "The density-matrix renormalization group". In: *Reviews of Modern Physics* 77.1 (Apr. 2005), pp. 259–315. DOI: 10.1103/revmodphys.77.259. URL: https://doi.org/10.1103%2Frevmodphys.77.259.

[72] U. Schollwöck. "The density-matrix renormalization group in the age of matrix product states". In: *Annals of Physics* 326.1 (Jan. 2011), pp. 96–192. DOI: 10.1016/j.aop.2010.09.012. URL: https://doi.org/10.1016%2Fj.aop.2010.09.012.

[73] N. A. Smith and M. Johnson. "Weighted and Probabilistic Context-Free Grammars Are Equally Expressive". In: *Computational Linguistics* 33.4 (2007), pp. 477–491. DOI: 10.1162/coli.2007.33.4.477. URL: https://aclanthology.org/J07-4003.

[74] T. Sochi. *Introduction to Tensor Calculus*. Mar. 2016. DOI: 10.6084/M9.FIGSHARE.3084508.

[75] A. Steane. "Quantum computing". In: *Reports on Progress in Physics* 61.2 (1998), p. 117.

[76] J. Stokes and J. Terilla. "Probabilistic modeling with matrix product states". In: *Entropy* 21.12 (2019), p. 1236.

[77] E. M. Stoudenmire and D. J. Schwab. *Supervised Learning with Quantum-Inspired Tensor Networks*. 2017. arXiv: 1605.05775 [stat.ML].

[78] E. Stoudenmire and S. R. White. "Studying Two-Dimensional Systems with the Density Matrix Renormalization Group". In: *Annual Review of Condensed Matter Physics* 3.1 (2012), pp. 111–128. DOI: 10.1146/annurev-conmatphys-020911-125018. eprint: https://doi.org/10.1146/annurev-conmatphys-020911-125018. URL: https://doi.org/10.1146/annurev-conmatphys-020911-125018.

[79] *Tensor Diagram Notation*. https://tensornetwork.org/diagrams/. Accessed: 2021-09-30.

[80] D. Tong. *Lectures on String Theory*. 2009. DOI: 10.48550/ARXIV.0908.0333. URL: https://arxiv.org/abs/0908.0333.

[81] F. Verstraete, V. Murg, and J. Cirac. "Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems". In: *Advances in Physics* 57.2 (Mar. 2008), pp. 143–224. DOI: 10.1080/14789940801912366. URL: https://doi.org/10.1080%2F14789940801912366.

[82] G. Vidal. "Class of quantum Many-Body states that can be efficiently simulated". In: *Physical Review Letters* 101.11 (2008), pp. 1–4. ISSN: 00319007. DOI: 10.1103/PhysRevLett.101.110501. arXiv: 0610099 [quant-ph].

[83] G. Vidal. "Entanglement renormalization". In: *Physical Review Letters* 99.22 (2007), pp. 1–4. ISSN: 00319007. DOI: 10.1103/PhysRevLett.99.220405. arXiv: 0512165 [cond-mat].

[84] G. Vidal. "Efficient Classical Simulation of Slightly Entangled Quantum Computations". In: *Physical Review Letters* 91.14 (Oct. 2003). DOI: 10.1103/physrevlett.91.147902. URL: https://doi.org/10.1103%2Fphysrevlett.91.147902.

[85] D.-S. Wang, D. T. Stephen, and R. Raussendorf. "Qudit quantum computation on matrix product states with global symmetry". In: *Physical Review A* 95.3 (2017), p. 032312.

[86] S. R. White. "Density matrix formulation for quantum renormalization groups". In: *Phys. Rev. Lett.* 69 (19 Nov. 1992), pp. 2863–2866. DOI: 10.1103/PhysRevLett.69.2863. URL: https://link.aps.org/doi/10.1103/PhysRevLett.69.2863.

[87] S. R. White. "Density matrix renormalization group algorithms with a single center site". In: *Phys. Rev. B* 72 (18 Nov. 2005), p. 180403. DOI: 10.1103/PhysRevB.72.180403. URL: https://link.aps.org/doi/10.1103/PhysRevB.72.180403.

[88] S. R. White. "Density-matrix algorithms for quantum renormalization groups". In: *Phys. Rev. B* 48 (14 Oct. 1993), pp. 10345–10356. DOI: 10.1103/PhysRevB.48.10345. URL: https://link.aps.org/doi/10.1103/PhysRevB.48.10345.

[89] S. R. White and R. L. Martin. "iAb initio/i quantum chemistry using the density matrix renormalization group". In: *The Journal of Chemical Physics* 110.9 (Mar. 1999), pp. 4127–4130. DOI: 10.1063/1.478295. URL: https://doi.org/10.1063%2F1.478295.

[90] A. J. Wilson et al. "What is a vector?" In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 372.1719 (2017). ISSN: 14712970. DOI: 10.1098/rstb.2016.0085.

[91] S. Wouters and D. V. Neck. "The density matrix renormalization group for ab initio quantum chemistry". In: *The European Physical Journal D* 68.9 (Sept. 2014). DOI: 10.1140/epjd/e2014-50500-1. URL: https://doi.org/10.1140%2Fepjd%2Fe2014-50500-1.