APC 523: Numerical Algorithms for Scientific Computing
Implementation of a matrix-product state algorithm to calculate approximate energy eigenvalues of a quantum system

Akshay Krishna
Princeton University
(Dated: May 15, 2018)

Many problems in quantum mechanics and quantum computing require the use of resources that scale exponentially with the size of the system. It is therefore hopeless to try to solve these problems exactly using classical resources until quantum computers are invented. However, some problems of interest can be solved approximately in polynomial time and memory using a powerful and versatile set of techniques known as the matrix-product state formalism. In this report I describe one such algorithm.

I. INTRODUCTION TO QUANTUM MECHANICS AND NOTATION

An accurate description of particles such as electrons at the microscopic level requires quantum mechanics (QM). Unlike in classical mechanics, where the state of a system is completely specified by the physical positions $\mathbf{x}_i(t)$ and momenta $\mathbf{p}_i(t)$ of its constituent degrees of freedom, in quantum mechanics, the state of a system is described by a vector in an abstract space. QM therefore is essentially an exercise in linear algebra. Just like in classical mechanics, one is interested in answering the following questions – What is the lowest energy state of the system? How does the system evolve in time? In QM, answering these reduces to solving eigenvalue equations and matrix-valued differential equations. However the weird and wonderful properties of QM like superposition and entanglement makes simulating a quantum system by solving these equations on a classical computer exponentially difficult. In this project I review a class of algorithms, known as Matrix product states, that is well-suited to approximately solving these problems, using computational resources that are polynomial, not exponential, in the size of the system. Matrix product state techniques are very powerful tools to help us understand the complexity of quantum systems using classically available computational resources.

A. Single particle quantum mechanics and superposition

First, a crash course in some quantum mechanics (QM) and linear algebra notation and jargon that will be needed to describe this project. For details see Refs. [1, 2]. In QM, the state of a single particle (or degree of freedom) with d available states is described by a vector $|\psi\rangle$ in a Hilbert space (i.e. linear vector space) of dimension d. The simplest case is when $d = 2$ (aka a two-level system), and the linear vector space has a dimension of 2, and this is assumed to be the case in what follows. An orthonormal basis for this space is denoted by the set of vectors $\{|\uparrow\rangle, |\downarrow\rangle\}$. Colloquially, $|\uparrow\rangle$ may be understood as an electron having its 'spin'-up, or aligned with a magnetic field along the z-axis, and $|\downarrow\rangle$ as an electron with its 'spin'-down, or anti-aligned with a magnetic field along the z-axis. A vector description is needed because the system can be in any arbitrary *superposition* of the basis vectors. For the two-level system, a generic state may be represented as $|\psi\rangle = c_\uparrow |\uparrow\rangle + c_\downarrow |\downarrow\rangle$, with $c_\uparrow$ and $c_\downarrow$ as arbitrary complex numbers. If the state is normalized, then $|c_\uparrow|^2 + |c_\downarrow|^2 = 1$. If the spin of the electron is measured in this superposition, the probability that it is up is $|c_\uparrow|^2$, and the probability it is down is $|c_\downarrow|^2$. This superposition principle is one of the features that makes quantum mechanics counter-intuitive and also very powerful.

The notation $|\uparrow\rangle$ and $|\downarrow\rangle$ for basis vectors is useful for analytical manipulation, but for numerical concreteness, are better represented as 2×1 column vectors

$$|\uparrow\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |\downarrow\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{1}$$

$$|\psi\rangle = \begin{pmatrix} c_\uparrow \\ c_\downarrow \end{pmatrix}. \tag{2}$$

Dual vectors are denoted $\langle\psi|$ and are represented by row vectors

$$\langle\uparrow| = \begin{pmatrix} 1 & 0 \end{pmatrix}, \langle\uparrow| = \begin{pmatrix} 0 & 1 \end{pmatrix} \tag{3}$$

$$\langle\psi| = \begin{pmatrix} c_\uparrow^* & c_\downarrow^* \end{pmatrix}. \tag{4}$$

Inner products between two states $|\psi_1\rangle$ and $|\psi_2\rangle$ are denoted $\langle\psi_1 \,|\, \psi_2\rangle$ and are to be understood as the multiplication of a row vector with a column vector to give a scalar. These are useful for evaluating various quantities of measurable interest. Outer products are denoted $|\psi_1\rangle \langle\psi_2|$ and are to be understood as the multiplication of a column vector with a row vector to give a (square) matrix, and will be needed in the algorithm that follows.

States are not static, and evolve in time under the action of a 2×2 matrix, known as the Hamiltonian $\hat{H}$. $\hat{H}$ is Hermitian ($H_{ij} = H_{ji}^*$) and its elements have the physical dimensions of energy. The time evolution of a state is governed by the Schrödinger equation

$$i\hbar \frac{d\,|\psi\rangle}{dt} = \hat{H}\,|\psi\rangle\,, \tag{5}$$

where Planck's constant $\hbar = 6.626 \times 10^{-34} Js$ is a physical constant that maintains the dimensional consistency of physical quantities on both sides of the equation. If a state $|\phi_n\rangle$ is an eigenvector (aka eigenstate) of the Hamiltonian $\hat{H}$, i.e. $\hat{H}|\phi_n\rangle = E_n|\phi_n\rangle$, with E_n the energy eigenvalue, then time evolution (by solving Eq. (5)) becomes straightforward

$$|\psi(t)\rangle = e^{-iE_n t/\hbar}|\phi_n\rangle . \tag{6}$$

This is why one is interested in solving for the eigenvalues and eigenstate of $\hat{H}$. In particular, for an isolated system at low temperatures, the system is at the lowest possible energy, and settles into the eigenstate of $\hat{H}$ with the smallest eigenvalue (aka the ground state). The subsequent time evolution of this state, is simply the accumulation of a scalar phase $e^{-iE_n t/\hbar}$, which is not relevant for most purposes.

For a two particle system with basis states $|\uparrow\rangle$ and $|\downarrow\rangle$, a generic Hamiltonian $\hat{H}$ is a 2×2 matrix that can be written as a linear combination of the three "Pauli"-like Hermitian matrices $\hat{S}_x, \hat{S}_y, \hat{S}_z$ and the identity matrix $\hat{I}$. With $|\uparrow\rangle$ and $|\downarrow\rangle$ as in Eq. (1), these matrices are

$$\hat{S}_x = \begin{pmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{pmatrix}, \qquad \hat{S}_y = \begin{pmatrix} 0 & -\frac{i}{2} \\ \frac{i}{2} & 0 \end{pmatrix},$$
$$\hat{S}_z = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & -\frac{1}{2} \end{pmatrix}, \qquad \hat{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{7}$$

As an example, if the Hamiltonian $\hat{H} = \hat{S}_x$, then the two eigenvectors are $|\phi_+\rangle = \frac{1}{\sqrt{2}}|\uparrow\rangle + \frac{1}{\sqrt{2}}|\downarrow\rangle$, with eigenvalue $E_+ = 1/2$, and $|\phi_-\rangle = \frac{1}{\sqrt{2}}|\uparrow\rangle - \frac{1}{\sqrt{2}}|\downarrow\rangle$, with eigenvalue $E_- = -1/2$.

One can also define the two real matrices $\hat{S}_\pm = \hat{S}_x \pm i\hat{S}_y$, which will be useful in the algorithmic implementation to avoid using complex numbers.

B. Many-particle quantum mechanics – entanglement

A state of many particles is described by a tensor product of the single particle states. So a state of two particles, each of which is in a two-level system, can be generically denoted as $|\psi\rangle = c_{\uparrow\uparrow}|\uparrow\uparrow\rangle + c_{\uparrow\downarrow}|\uparrow\downarrow\rangle + c_{\downarrow\uparrow}|\downarrow\uparrow\rangle + c_{\downarrow\downarrow}|\downarrow\downarrow\rangle$, with $|c_{\uparrow\uparrow}|^2 + |c_{\uparrow\downarrow}|^2 + |c_{\downarrow\uparrow}|^2 + |c_{\downarrow\downarrow}|^2 = 1$ for normalization. Here the state $|\uparrow\uparrow\rangle \equiv |\uparrow\rangle|\uparrow\rangle$ is one in which both electrons have their spins up, in $|\uparrow\downarrow\rangle \equiv |\uparrow\rangle|\downarrow\rangle$ the first electron has its spin up and the second one spin down, and so on.

When there is more than one particle in the system, the particles can be *entangled* with each other. Entanglement is a term for quantum-mechanical correlations between different particles. An unentangled state is one in which the particles are in independent superpositions of the basis states, e.g. $|\uparrow\uparrow\rangle$, in which both electrons have a 100% probability of having their spins up. A general unentangled state of two particles is

$$\begin{aligned} |\psi_0\rangle &= (c_\uparrow|\uparrow\rangle + c_\downarrow|\downarrow\rangle)(d_\uparrow|\uparrow\rangle + d_\downarrow|\downarrow\rangle) \qquad (8)\\ &= c_\uparrow d_\uparrow|\uparrow\uparrow\rangle + c_\uparrow d_\downarrow|\uparrow\downarrow\rangle + c_\downarrow d_\uparrow|\downarrow\uparrow\rangle + c_\downarrow d_\downarrow|\downarrow\downarrow\rangle , \end{aligned}$$

where $|c_\uparrow|^2 + |c_\downarrow|^2 = |d_\uparrow|^2 + |d_\downarrow|^2 = 1$. In this unentangled state, the first electron has its spin up with probability $|c_\uparrow|^2$ and spin down with probability $|c_\uparrow|^2$, The probability that the second electron has its spin up is $|d_\uparrow|^2$ and spin down is $|d_\uparrow|^2$, independent of the first electron.

An example of an entangled state is

$$|\psi_e\rangle = \frac{1}{\sqrt{2}}|\uparrow\uparrow\rangle + \frac{1}{\sqrt{2}}|\downarrow\downarrow\rangle , \tag{9}$$

where the system has a 50% probability of being in a state with both electrons having their spins up and a 50% probability of both electrons having their spins down. In this entangled state, the spins of both electrons are correlated and there is *no way* to factorize $|\psi_e\rangle$ into a product of single-particle vectors as in Eq. (8).

The Hamiltonian for a two-particle system is composed of tensor products of the single-particle matrices described in Eq. (7). In this case, a generic Hamiltonian is of the form $\hat{H} = a_{xI}\hat{S}_x^{(1)} \otimes \hat{I}^{(2)} + a_{xx}\hat{S}_x^{(1)} \otimes \hat{S}_x^{(2)} + \cdots$, where a_{xI}, a_{xx} and so on are arbitrary real numbers. The superscripts $^{(1)}$ and $^{(2)}$ denote which of the two particles the matrix acts on.

An important two particle Hamiltonian is called the Heisenberg Hamiltonian

$$\begin{aligned} \hat{H} &= \mathbf{S}^{(1)} \cdot \mathbf{S}^{(2)} \qquad (10)\\ &= \hat{S}_x^{(1)} \otimes \hat{S}_x^{(2)} + \hat{S}_y^{(1)} \otimes \hat{S}_y^{(2)} + \hat{S}_z^{(1)} \otimes \hat{S}_z^{(2)}. \qquad (11) \end{aligned}$$

The eigenstates $|\phi_i\rangle$ and corresponding eigenvalues E_i of this Hamiltonian can be shown to be

$$|\phi_0\rangle = \frac{1}{\sqrt{2}}|\uparrow\downarrow\rangle - \frac{1}{\sqrt{2}}|\downarrow\uparrow\rangle , \qquad E_0 = -3/4 \tag{12}$$
$$|\phi_1\rangle = \frac{1}{\sqrt{2}}|\uparrow\downarrow\rangle + \frac{1}{\sqrt{2}}|\downarrow\uparrow\rangle , \qquad E_1 = 1/4 \tag{13}$$
$$|\phi_2\rangle = |\uparrow\uparrow\rangle , \qquad E_2 = 1/4 \tag{14}$$
$$|\phi_3\rangle = |\downarrow\downarrow\rangle , \qquad E_3 = 1/4. \tag{15}$$

A general basis vector of N particles in a two-level system will be denoted by a string of N up/down arrows, e.g. $|\uparrow\downarrow\cdots\uparrow\rangle$. A general state will be an arbitrary superposition of all basis states and will be represented as

$$|\psi\rangle = \sum_{\sigma_1,\cdots,\sigma_N=\{\uparrow,\downarrow\}} c_{\sigma_1\cdots\sigma_N}|\sigma_1\cdots\sigma_N\rangle , \tag{16}$$

where each of the indices σ_i takes values $\uparrow$ and $\downarrow$, and the coefficients $c_{\sigma_1\cdots\sigma_N}$ are arbitrary complex numbers, which satisfy the normalization constraint $\sum_{\sigma_1,\cdots,\sigma_N} |c_{\sigma_1\cdots\sigma_N}|^2 = 1$. In the special case where only one coefficient is equal to one and the rest are identically zero, the state of the

system is unentangled because each particle has a well defined spin-up or down with 100% probability. However, for a generic (entangled) state, one needs 2^N independent non-zero complex numbers to fully represent a quantum state of N particles, and this is the origin of all difficulties. The Hamiltonian for this case is a $2^N \times 2^N$ Hermitian matrix.

If entanglement and superposition did not exist, one would not be faced with this exponential growth in complexity. Using standard numerical techniques to solve the eigenvalue problem of a many-particle $\hat{H}$ takes $\mathcal{O}(d^{6N})$ time and $\mathcal{O}(d^{4N})$ memory, is intractable for all but systems of very small N.

However, physical systems usually have symmetries which ensure that a large number of elements in H are zero. Further, many accessible states, especially the ground state are found to have 'low entanglement'.

By cleverly exploiting 'low entanglement' and the sparsity structure of $\hat{H}$, one can approximately represent Hamiltonians and ground states using far fewer parameters than the exponential scaling above would naively suggest. The next sections describes some linear algebra tools to enable this construction and also a quantitative notion of the term 'entanglement'.

II. SINGULAR VALUE DECOMPOSITION AND VON NEUMANN ENTANGLEMENT ENTROPY

The entanglement of an arbitrary state such as in Eq. (16) can be quantified as follows. A system of N particles can be divided into two subsystems, say a left subsystem with N_L particles and a right subsystem with $N_R = N - N_L$ particles. The basis for the entire system is the set of 2^N vectors $\{|\sigma_1 \cdots \sigma_N\rangle\}$, which is a tensor product of the 2^{N_L} basis vectors $\{|\sigma_1 \cdots \sigma_{N_L}\rangle\}$ of the left subsystem and the 2^{N_R} basis vectors $\{|\sigma_{N_L+1} \cdots \sigma_N\rangle\}$ of the right subsystem. Using this partition of the system into two subsystems, Eq. (16) is rewritten as

$$|\psi\rangle = \sum_{i=1,j=1}^{i=2^{N_L},j=2^{N_R}} m_{ij} \, |i\rangle_L \, |j\rangle_R \,, \tag{17}$$

where $\{|i\rangle_L\}$ is shorthand for the orthonormal basis $\{|\sigma_1 \cdots \sigma_{N_L}\rangle\}$ for the left half system, and likewise for the right half system. The 2^N coefficients $c_{\sigma_1 \cdots \sigma_N}$ are reshaped into a $2^{N_L} \times 2^{N_R}$ matrix with elements m_{ij}.

Any $m \times n$ matrix M can be converted to a form $M = USV^\dagger$, where U is a $m \times \min(m,n)$ unitary matrix, V is a an $n \times \min(m,n)$ unitary matrix, and S is a $\min(m,n) \times \min(m,n)$ diagonal matrix. In other words,

$$M_{ij} = \sum_k U_{ik} S_{kk} V^*_{jk},$$

$$\sum_k U^*_{ki} U_{kj} = \delta_{ij},$$

$$\sum_k V^*_{ki} V_{kj} = \delta_{ij},$$

where the diagonal entries $s_k \equiv S_{kk}$ of S are the singular values of M and this transformation is called the singular value decomposition (SVD). The rank of the matrix M is equal to the number of non-zero singular values.

The existence of SVD allows us to rewrite Eq. (17) as

$$|\psi\rangle = \sum_k^{d'} s_k \left(\sum_i U_{ik} \, |i\rangle_L \right) \left(\sum_j V^*_{jk} \, |j\rangle_R \right) \tag{18}$$

$$= \sum_k^{d'} s_k \, |\phi_k\rangle_L \, |\phi_k\rangle_R \,. \tag{19}$$

Here $d' \equiv \min(2^{N_L}, 2^{N_R})$ and $|\phi_k\rangle_L = \sum_i U_{ik} \, |i\rangle_L$ is a new orthonormal basis (i.e. ${}_L\langle \phi_k \, | \, \phi_{k'} \rangle_L = \delta_{kk'}$) for the left half subsystem, and similarly $|\phi_k\rangle_R = \sum_i V^*_{jk} \, |i\rangle_L$ is an orthonormal basis for R. These are called the respective Schmidt basis vectors and this is the Schmidt decomposition. If $|\psi\rangle$ is normalized, then $\sum_{ij} |m_{ij}|^2 = \sum_k |s_k|^2 = 1$.

The entanglement of the left subsystem and the right subsystem is encoded by the singular values s_k, and the entanglement entropy of the left and right subsystems with each other is described by the entanglement entropy One may describe the entanglement properties of a state by the entanglement entropy $\mathcal{S}$, defined as the von Neumann entropy of the singular values

$$\mathcal{S} = -\sum_{k=1}^{d'} |s_k|^2 \ln |s_k|^2. \tag{20}$$

If only one of the s_k is 1, and all the rest are zero, then by Eq. (19), one has a non-entangled or factorizable state with an entanglement entropy $\mathcal{S} = 0$. In this case the matrix described by elements m_{ij} in Eq. (17) has rank 1. At the other extreme, if all the d' singular values s_k have equal amplitude, the entanglement entropy $\mathcal{S} = \ln d'$. In general, an arbitrary state will have an entanglement entropy that lies within these bounds

$$0 \leq \mathcal{S} \leq \ln\big(\min(2^{N_L}, 2^{N_R})\big). \tag{21}$$

The entanglement entropy depends on how the partition between left and right subsystems is made. For $N = 2$ particles, the only non-trivial partition of the system is to assign the first electron to the left subsystyem and the second to the right. Then the entangled state $|\psi_e\rangle$ in Eq. (9) is maximally entangled because $\mathcal{S} = \ln 2$ saturates the upper bound Eq. (21).

III. THE MPS DESCRIPTION

A. Matrix-product states (MPS)

The idea of the matrix product state (MPS) ansatz [3, 4] is to rewrite the many-particle state Eq. (16) as a

'tensor network'

$$|\psi\rangle = \sum_{\sigma_1,\cdots,\sigma_N=\{\uparrow,\downarrow\}} \sum_{d_1=1}^{1} \sum_{d_2,\cdots,d_N=1}^{\chi} c^{[1]\sigma_1}_{d_1 d_2} \cdots c^{[N]\sigma_N}_{d_N d_1} |\sigma_1 \cdots \sigma_N\rangle . \quad (22)$$

So instead of a single N-index tensor of 2^N coefficients as in Eq. (16), one has now N three-index tensors $c^{[i]}, i = 1, \cdots, N$, each of which has three indices – one physical index σ_i which takes values $\uparrow$ and $\downarrow$, and two 'bond' indices d_i and d_{i+1} which take values from 1 to an integer χ. The first index d_1 is a dummy index and takes only the value 1 because one considers open boundary conditions for simplicity, i.e., the first particle has no neighbour on its left and the last particle no neighbour on its right. In other words $c^{[i]\sigma_i}_{d_i d_{i+1}}$ denotes the $(\sigma_i, d_{i+1}, d_{i+2})$ $^{\text{th}}$ element of the tensor $c^{[i]}$ associated with particle i. The total number of terms to specify the state is now $\approx 2\chi^2 N$, where the parameter χ is known as the bond dimension. Fig. 1 depicts the MPS schematically.

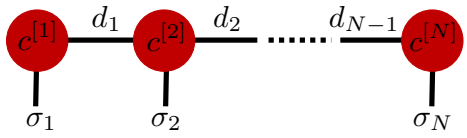

FIG. 1. The MPS of N particles in Eq. 22. Each tensor $c^{[i]}$ is represented as a circle with the number of legs indicating indices. Connected legs are summed over and can be represented by any dummy label, and open legs correspond to free physical indices σ_i.

The price paid for representing the state by a number of parameters that is linear in system size ($\approx 2\chi^2 N$) and not exponential (2^N) is that one cannot capture the entanglement properties of the state fully.

An MPS such as that in Eq. (22) may be partitioned into a left subsystem and a right subsystem just as in Eq. (17) by assigning the first N_L particles to the left subsytem and the remainder $N_R = N - N_L$ particles to the right subsystem. Then the MPS may be written as

$$|\psi\rangle = \sum_{i=1}^{2^{N_L}} \sum_{j=1}^{2^{N_R}} \sum_{d_{L+1}=1}^{\chi} c^{(L)}_{i d_{L+1}} c^{(R)}_{j d_{L+1}} |ij\rangle . \quad (23)$$

Here i is shorthand for all the physical indices $\sigma_1 \cdots \sigma_{N_L}$ of particles in the left subsystem, and likewise for the right subsystem $(\sigma_{N_{L+1}} \cdots \sigma_N)$, and

$$c^{(L)}_{i d_{L+1}} \equiv \sum_{d_1,\cdots,d_{N_L}} c^{[1]\sigma_1}_{d_1 d_2} \cdots c^{[N_L]\sigma_{N_L}}_{d_{N_L} d_{N_L+1}}$$

is a $2^{N_L} \times \chi$ matrix. Similarly $c^{(R)}_{j d_{L+1}}$ is a $\chi \times 2^{N_L}$ matrix that is constructed by evaluating all the intermediate tensor products between tensors $c^{[N_{L+1}]}$ and $c^{[N]}$ from Eq. (22).

Putting this partitioned MPS in the form of Eq. 17, one gets

$$|\psi\rangle = \sum_{i=1,j=1}^{i=2^{N_L},j=2^{N_R}} c_{ij} |ij\rangle$$

where $c_{ij} \equiv \sum_{d_{L+1}=1}^{\chi} c^{(L)}_{i d_{L+1}} c^{(R)}_{j d_{L+1}}$ defines a $2^{N_L} \times 2^{N_R}$ matrix. However, since this matrix is the product of a $2^{N_L} \times \chi$ matrix and a $\chi \times 2^{N_L}$ matrix, its rank is upper bounded by χ (for matrices A and B, $\text{rank}(AB) \leq \min(\text{rank}(A), \text{rank}(B))$). From the discussion of entanglement entropy below Eq. (20), it follows that the entanglement entropy of the MPS satisfies $\mathcal{S} \leq \ln \chi$, which is a much tighter bound than Eq. (21).

Thus an MPS with fixed finite χ cannot represent states of high entanglement well, and the MPS description becomes exact only when $\chi \sim 2^N$. Nevertheless, one can ask, for a given Hamiltonian $\hat{H}$ what is the MPS that is the best approximation to the eigenvector of $\hat{H}$ with lowest eigenvalue (aka ground state)? If the answer seems to converge to a certain value as one increases χ, then one can be fairly sure that one has attained a 'good' approximation to the true ground state, using only a small (non-exponential) number of parameters.

B. Matrix-product operators (MPO)

In order to be able to do calculations with MPS, one needs to be able to represent the Hamiltonian $\hat{H}$ using a sub-exponential number of parameters, and define matrix-vector operations using MPS. In analogy with Eq. (16), one can write a generic Hamiltonian as

$$\hat{H} = \sum_{\substack{\sigma_1,\cdots,\sigma_N=\{\uparrow,\downarrow\} \\ \sigma'_1,\cdots,\sigma'_N=\{\uparrow,\downarrow\}}} c_{\sigma_1\cdots\sigma_N,\sigma'_1\cdots\sigma'_N} |\sigma_1 \cdots \sigma_N\rangle \langle\sigma'_1 \cdots \sigma'_N| , \quad (24)$$

using the outer product notation discussed in Section I A. There are 4^N elements $c_{\sigma_1\cdots\sigma_N,\sigma'_1\cdots\sigma'_N}$ in this matrix. Just like in Eq. (22), one can write a tensor network description

$$\hat{H} = \sum_{\substack{\sigma_1,\cdots,\sigma_N=\{\uparrow,\downarrow\} \\ \sigma'_1,\cdots,\sigma'_N=\{\uparrow,\downarrow\}}} \sum_{d_1=1}^{1} \sum_{d_2,\cdots,d_N=1}^{\chi_o} w^{[1]\sigma_1\sigma'_1}_{d_1 d_2} \cdots w^{[N]\sigma_N\sigma'_N}_{d_N d_1} |\sigma_1 \cdots \sigma_N\rangle \langle\sigma'_1 \cdots \sigma'_N| . \quad (25)$$

Here each of the $w^{[i]\sigma_i\sigma'_i}_{d_i d_{i+1}}$ is a four-index tensor, with physical indices σ_i and σ'_i taking values $\uparrow$ and $\downarrow$, and 'bond' indices d_i and d_{i+1} taking values from 1 to an integer χ_o (in general different from the bond dimension χ of the MPS).

The total number of terms required to represent the so-called matrix-product operator (MPO) form of $\hat{H}$ is

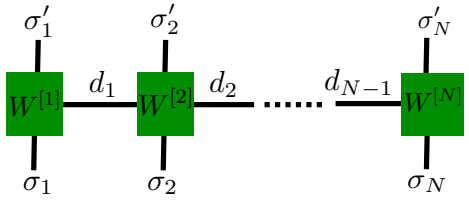

FIG. 2. The MPS of N particles in Eq. 25. Each tensor $w^{[i]}$ is represented as a square with the number of legs indicating indices. Connected legs are summed over and can be represented by any dummy label, and open legs correspond to free physical indices σ_i, σ'_i.

$\approx 4\chi_0^2 N$, a drastic reduction from 4^N. Generic Hamiltonians can only be approximately represented as MPOs, but many Hamiltonians can be represented exactly using an MPO.

In particular, the generalization of the Heisenberg Hamiltonian Eq. (11) to N particles can be represented exactly using an MPO with a bond dimension $\chi_0 = 5$. The N-particle Heisenberg Hamiltonian (with open boundary conditions) is

$$\hat{H} = \sum_{i=1}^{N-1} \mathbf{S}^{(i)} \cdot \mathbf{S}^{(i+1)} \tag{26}$$

$$= \sum_{i=1}^{N-1} \hat{S}_x^{(i)} \hat{S}_x^{(i+1)} + \hat{S}_y^{(i)} \hat{S}_y^{(i+1)} + \hat{S}_z^{(i)} \hat{S}_z^{(i+1)}. \tag{27}$$

This Hamiltonian can be represented exactly in MPO format (Eq. (25)) using the "Pauli" matrices (Eq. (7)) in block notation as

$$w^{[1]} = \begin{pmatrix} I & \hat{S}_x & \hat{S}_y & \hat{S}_z & 0 \end{pmatrix}, \tag{28}$$

$$w^{[i]} = \begin{pmatrix} I & \hat{S}_x & \hat{S}_y & \hat{S}_z & 0 \\ 0 & 0 & 0 & 0 & \hat{S}_x \\ 0 & 0 & 0 & 0 & \hat{S}_y \\ 0 & 0 & 0 & 0 & \hat{S}_z \\ 0 & 0 & 0 & 0 & I \end{pmatrix}, \quad 2 \le i \le N-1, \text{ and} \tag{29}$$

$$w^{[N]} = \begin{pmatrix} 0 \\ \hat{S}_x \\ \hat{S}_y \\ \hat{S}_z \\ I \end{pmatrix}. \tag{30}$$

In the notation of Eq. (25), the indices d_i are the block indices in the $w^{[i]}$ matrices above and σ_i are internal indices for the 2×2 blocks.

C. Canonical forms and the QR decomposition

For a given state, the MPS representation Eq. (22) is not unique as one can perform a transformation

$$c^{[i-1]\sigma_{i-1}}_{d_{i-1}d_i} \to \sum_{e_i} c^{[i]\sigma_{i-1}}_{d_{i-1}e_i} U_{e_i d_i} \text{ and}$$
$$c^{[i]\sigma_i}_{d_i d_{i+1}} \to \sum_{e'_i} U^{-1}_{d_i e'_i} c^{[i]\sigma_i}_{e'_i d_{i+1}}, \tag{31}$$

that leaves the state unchanged from the point of view of the physical indices σ_i. This allows to write the tensors in a number of useful canonical forms. In the left canonical form, tensors $c^{[i]}$ are denoted $A^{[i]}$ such that

$$\sum_{d_i \sigma_i} A^{[i]\sigma_i}_{d_i d_{i+1}} A^{*[i]\sigma_i}_{d_i d'_{i+1}} = \delta_{d_{i+1} d'_{i+1}}. \tag{32}$$

In the right-canonical form, tensors are denoted $B^{[i]}$ such that

$$\sum_{d_{i+1}\sigma_i} B^{[i]\sigma_i}_{d_i d_{i+1}} B^{*[i]\sigma_i}_{d'_i d_{i+1}} = \delta_{d_i d'_i}. \tag{33}$$

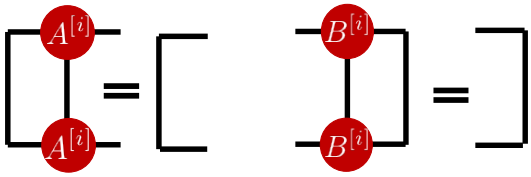

FIG. 3. The conditions for left-canonicalized (Eq. 32) and right-canonicalized (Eq. 33) MPS tensors, represented using the schematic notations of Figs. 1 and 2.

The QR decomposition is a simplification of the SVD that may be used to bring tensors into canonical form. Any $m \times n$ matrix M can be converted to a form $M = QR$, where Q is a $m \times \min(m,n)$ unitary matrix with orthonormal columns $\sum_k Q_{ki} Q^*_{kj} = \delta_{ij}$, and R is a $\min(m,n) \times n$ upper triangular matrix. The method to systematically put an MPS into left-canonical form, upto the n^{th} particle ($n \le N$), is described in Algorithm 1.

Algorithm 1 Left canonicalizing an MPS

procedure LEFTCANONICALIZE(n) ▷ $n \le N$
  for particle $i \in [1, 2, \cdots, n]$ **do**
    reshape tensor $c^{[i]}$ to a matrix $M_{\sigma_i d_{i-1}, d_i} \equiv c^{[i]\sigma_i}_{d_{i-1} d_i}$
    perform a QR decomposition $M = QR$
    reshape Q to a tensor $A^{[i]\sigma_i}_{d_{i-1} d_i} \equiv Q_{\sigma_i d_{i-1}, d_i}$
    if $i \neq N$ **then**
      update $c^{[i+1]}$ as $c^{[i+1]\sigma_i}_{d_i d_{i+1}} \leftarrow \sum_{b_i} R_{d_i b_i} c^{[i+1]\sigma_i}_{b_i d_{i+1}}$
    end if
  end for
end procedure

An analogous method can be used to right-canonicalize the MPS, starting from the N^{th} particle and moving down, using the Hermitian conjugate of the QR decomposition. It is often useful to have an MPS that is left canonicalized from particle 1 upto some particle $n-1$, and right canonicalized from particle $n+1$ to N. In this form the MPS is written as

$$|\psi\rangle = \sum_{\sigma_1,\cdots,\sigma_N=\{\uparrow,\downarrow\}} \sum_{d_1=1}^{1} \sum_{d_2,\cdots,d_N=1}^{\chi} |\sigma_1\cdots\sigma_N\rangle \\ A^{[1]\sigma_1}_{d_1 d_2}\cdots A^{[n-1]\sigma_{n-1}}_{d_{n-1}d_n} c^{[n]\sigma_n}_{d_n d_{n+1}} B^{[n+1]\sigma_{n+1}}_{d_{n+1}d_{n+2}}\cdots B^{[N]\sigma_N}_{d_N d_1}. \tag{34}$$

When a state is expressed in the form above, it is easy to evaluate the inner product $\langle\psi\,|\,\psi\rangle$, specifically with reference to the tensor $c^{[i]}$. It follows from the canonicalization conditions Eqs. 32 and 33 that

$$\langle\psi\,|\,\psi\rangle = \sum_{\sigma_n d_n d_{n+1}} c^{[n]\sigma_n}_{d_n d_{n+1}} c^{*[n]\sigma_n}_{d_n d_{n+1}}. \tag{35}$$

This form makes it easy to evaluate the entanglement entropy of a partition of the system into the a left subsystem with the first n particles, and a right subsystem with the remaining particles. For this case, by Eq. 20 and 34, $\mathcal{S} = -\sum_k |s_k|^2 \ln |s_k|^2$, where s_k's are the singular values of the reshaped matrix $c^{[n]}_{\sigma_n d_n, d_{n+1}}$.

It is also useful to calculate the quadratic form $\left\langle\psi\,\middle|\,\hat{H}\,\middle|\,\psi\right\rangle$. This can be done especially efficiently with an MPS in the canonical form of Eq. 34 as described in Algorithm 2.

Algorithm 2 Calculating $\left\langle\psi\,\middle|\,\hat{H}\,\middle|\,\psi\right\rangle$ with canonical MPS

procedure QUADRATICFORM(MPS, MPO) ▷ MPS is canonicalized, as given in Eq. 34
  define $L^{[0]}$ as a trivial three-index tensor $L^{[0]}_{1,1,1} = 1$
  for particle $i \in [1, 2, \cdots, n-1]$ **do**

$$L^{[i]}_{d_{i+1}b_{i+1}d'_{i+1}} = \sum_{d_i d'_i b_i \sigma_i \sigma'_i} L^{[i-1]}_{d_i b_i d'_i} w^{[i]\sigma_i\sigma'_i}_{b_i b_{i+1}} \times A^{[i]\sigma_i}_{d_i d_{i+1}} A^{*[i]\sigma'_i}_{d'_i d'_{i+1}}$$

  end for
  define $R^{[N+1]}$ as a trivial three-index tensor $R^{[N+1]}_{1,1,1} = 1$
  for particle $i \in [N, N-1, \cdots, n+1]$ **do**

$$R^{[i]}_{d_i b_i d'_i} = \sum_{d_{i+1}d'_{i+1}b_{i+1}\sigma_i\sigma'_i} R^{[i+1]}_{d_{i+1}b_{i+1}d'_{i+1}} w^{[i]\sigma_i\sigma'_i}_{b_i b_{i+1}} \times B^{[i]\sigma_i}_{d_i d_{i+1}} B^{*[i]\sigma'_i}_{d'_i d'_{i+1}}$$

  end for

$$X = \sum_{d_i d'_i b_i d_{i+1} d'_{i+1} b_{i+1}\sigma_i\sigma'_i} L^{[n-1]}_{d_n b_n d'_n} R^{[n+1]}_{d_{n+1}b_{n+1}d'_{n+1}} \times w^{[n]\sigma_n\sigma'_n}_{b_n b_{n+1}} c^{[n]\sigma_n}_{d_n d_{n+1}} c^{*[n]\sigma'_n}_{d'_n d'_{n+1}}$$

  return X ▷ $X = \langle\psi|\hat{H}|\psi\rangle$
end procedure

IV. OBTAINING GROUND STATES USING VARIATIONAL OPTIMIZATION

The simplest algorithm for obtaining the lowest eigenvalue states of MPO Hamiltonians goes by the name Density Matrix Renormalization Group (DMRG), for historical reasons. The algorithm hinges on the fact that for any normalized state $|\psi\rangle$ (MPS or otherwise), such that the inner product $\langle\psi\,|\,\psi\rangle = 1$, the quadratic form $\left\langle\psi\,\middle|\,\hat{H}\,\middle|\,\psi\right\rangle \geq E_{\min}$, where $E_{\min}$ is the smallest eigenvalue of $\hat{H}$.

Thus for a fixed MPO, by minimizing the function $\left\langle\psi\,\middle|\,\hat{H}\,\middle|\,\psi\right\rangle - \lambda\,\langle\psi\,|\,\psi\rangle$ with respect to the MPS one can obtain the ground state. The MPS has $\mathcal{O}(\chi^2 N)$ parameters, so this turns out to be a hard optimization problem. The problem is broken down into an iterative algorithm by optimizing over one tensor $c^{[i]}$ of the MPS at a time while keeping the remaining fixed. Using Algorithm 2 and Eq. 35 for an MPS of the form Eq. 34, one arrives at

$$\frac{\partial}{\partial c^{*[i]\sigma_i}_{d_{i-1}d_i}} \left(\langle\psi|\hat{H}|\psi\rangle - \lambda\,\langle\psi|\psi\rangle\right) = 0 \tag{36}$$

$$\sum_{d'_{i-1}d'_i b_{i-1}b_i\sigma'_i} L^{[i-1]}_{d'_{i-1}b_{i-1}d_{i-1}} R^{[i+1]}_{d'_i b_i d_i} w^{[i]\sigma_i\sigma'_i}_{b_{i-1}b_i} c^{[i]\sigma'_i}_{d'_{i-1}d'_i} = \lambda\; c^{[i]\sigma_i}_{d_{i-1}d_i}. \tag{37}$$

This is an eigenvalue equation of the form $\sum_b M_{ab}x_b = x_a$. The tensor $c^{[i]\sigma_i}_{d_{i-1}d_i}$ is reshaped into a column vector with $2\chi^2$ elements by combining all indices into one index. Similarly, the other terms can be grouped and reshaped as in Fig. 4 to obtain a compatibly sized square matrix. The solution to this eigenvalue equation with the smallest eigenvalue λ gives the optimized tensor $c[i]$ (when reshaped back to a three-index tensor) and an estimate for the eigenvalue E_{min}.

The full algorithm relies on keeping the MPS in a canonical form and sweeping from left to right to optimize tensors $c^{[1]}$ through $c^{[N]}$, turn by turn. One can the sweep back from right to left to and the again from left to right as many times as necessary to ensure convergence.

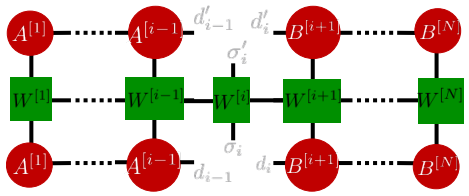

FIG. 4. Schematic for the six-index tensor M that appears in the eigenvalue equation Eq. 37. The tensor is reshaped to a $2\chi^2 \times 2\chi^2$ matrix by grouping indices d'_{i-1}, d'_i and σ'_i into a single row index, and indices d_{i-1}, d_i and σ_i into a single column index.

Algorithm 3 Variational Ground state search

procedure GROUNDSTATESEARCH($\hat{H}$) ▷ the input is a Hamiltonian $\hat{H}$
  create $\hat{H}$ in MPO form using Eqs. 28 – 30
  create a random initial MPS
  right-canonicalize all tensors $B^{[N]}$ through $B^{[1]}$
  iteratively calculate $R^{[N]}$ through $R^{[1]}$
  initialize the trivial tensor $L^{[0]}_{1,1,1} = 1$
  for particle $i \in [1, 2, \cdots N-1]$ **do** ▷ right-sweep
    solve Eq. 37 to find optimal tensor $c^{[i]}$
    left-canonicalize $c^{[i]}$ to obtain $A^{[i]}$
    update $c^{[i+1]}$ as in step 7 of Algorithm 1
    update $L^{[i]}$ as in step 4 of Algorithm 2
  end for
  for particle $i \in [N, N-1, \cdots 2]$ **do** ▷ left-sweep
    solve Eq. 37 to find optimal tensor $c^{[i]}$
    the eigenvalue λ is the current estimate for E_{min}
    right-canonicalize $c^{[i]}$ to obtain $B^{[i]}$
    update $c^{[i-1]}$
    update $R^{[i-1]}$ as in step 8 of Algorithm 2
  end for
  if error in estimated E_{min} is above tolerance **then**
    repeat steps 7-19
  end if
end procedure

The most time-consuming step in this process is the solution of the $2\chi^2 \times 2\chi^2$ eigenvalue problem Eq. 37, which most be done N times per sweep. Using regular dense linear solvers, the complexity is then $\mathcal{O}(N\chi^6)$. However, the eigenvalue problem is sparse and one only needs to solve for the smallest eigenvalue, so iterative methods can lead to a theoretical complexity of $\mathcal{O}(N\chi^4)$. However the overheads of reshaping the tensors leads the performance scaling more poorly with χ, especially as the matrix sizes become large.

V. IMPLEMENTATION

I implemented my code in Python, because its `numpy` and `scipy` libraries offer convenient, vectorized and fast wrappers to various `LAPACK` and `ARPACK` linear algebra functions. It is also very amenable to object-oriented programming, and thus useful for manipulating the MPO and MPS tensors as objects. My code consists of two main files

1. `mps.py`: This file contains a class `MPS`, with attributes `N` (number of particles), `chi` (bond dimension) and `M` (a list of tensors), among others. It has the following functions –
   - `rand_init()`: initializes the MPS with random tensors as required by step 3 in Algorithm 3
   - `left_canonicalize()`: sets the list of tensors `M` in left-canonicalized form $A^{[i]}$ using Algorithm 1
   - `right_canonicalize()`: sets the list of tensors `M` in right-canonicalized form $B^{[i]}$ using a complementary method to Algorithm 1
   - `inner_product()`: calculates inner products of the form Eq. 35
   - `get_EE()`: finds the entanglement entropy of the MPS using Eq. 20
   - `expectation_val()`: finds the quadratic form of an MPS with respect to an MPO using a method similar to Algorithm 2
   - `check_canonicalization()`: finds the quadratic form of an MPS with respect to an MPO using a method similar to Algorithm 2
2. `mpo.py`: This file contains a class `MPO`, with attributes `N` (number of particles), `chio` (operator bond dimension) and `Ham` (Hamiltonian type). It also contains a number of other parameters `Jx`, `Jy`, `Jz`, `hx`, `hz` to enable the creation of more general Hamiltonians besides the Heisenberg Hamiltonian Eq. 27. It has the following functions –
   - `make_Pauli()`: defines the various 2×2 matrices needed, Eq. 7 3
   - `make_MPO()`: creates the MPO using Eqs. 28 – 30
   - `act_on_MPS()`: performs matrix-vector products to obtain another MPS
   - `get_GS()`: the heart of the program that implements Algorithm 3

There is an auxiliary file `aux.py` that calls these other functions and interfaces with the programmer by allowing command line inputs to be passed as arguments.

VI. RESULTS AND DISCUSSION

The code is tested for the Heisenberg Hamiltonian Eq. 27, for which the Bethe ansatz shows the ground state energy to be

$$\lim_{N\to\infty} \frac{E_0}{N} = -\ln 2 + \frac{1}{4} = -0.4431... \tag{38}$$

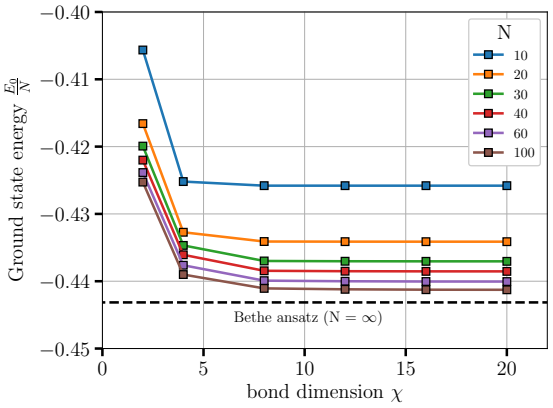

FIG. 5. The calculated ground state energy of the Heisenberg Hamiltonian for various sizes N saturates to a value independent of the bond dimension χ, indicating convergence. The exact result for an infinite system is shown for comparison.

For finite N, the exact ground state energy is higher than this value. Using the MPS variational ground state algorithm, the ground state energy can be estimated, and satisfactory convergence is seen in Fig. 5. The validity of the MPS ansatz is checked by plotting the entanglement entropy $\mathcal{S}$ of the ground state in Fig. 6. An MPS of bond dimension χ is restricts $\mathcal{S} \leq \ln \chi$. A saturation in $\mathcal{S}$ as χ increases indicates that the ground state is well-approximated by the MPS of finite (i.e. not exponential in N) bond dimension χ.

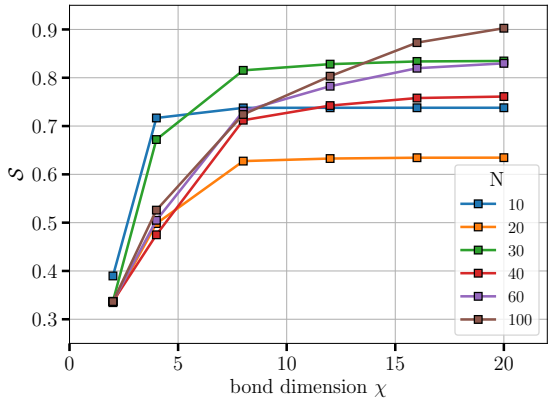

FIG. 6. The calculated entanglement entropy $\mathcal{S}$, Eq. 20, for a bipartition of the MPS ground state into two equally sized subsytems of $N/2$ particles each. For a fixed number of particles N, the entanglement entropy saturates to a constant value as the bond dimension is increased, indicating that the MPS ansatz is sufficient to capture the entanglement properties of the ground state.

The code is also tested for the transverse field Ising model Hamiltonian

$$\hat{H} = -\sum_{i=1}^{N-1} \hat{S}_z^{(i)} \hat{S}_z^{(i+1)} - h \sum_{i=1}^{N} \hat{S}_x^{(i)}. \tag{39}$$

This well studied model has a phase transition between a ferromagnetic and paramagnetic phase at $h = 1$. At $h = 0.5$, the ground state energy is known exactly [5]

$$\frac{E_0}{N} = 1 - \operatorname{cosec}\left(\frac{\pi}{4N+2}\right). \tag{40}$$

For this model too, the MPS algorithm performs extremely well in terms of convergence (Fig. 7).

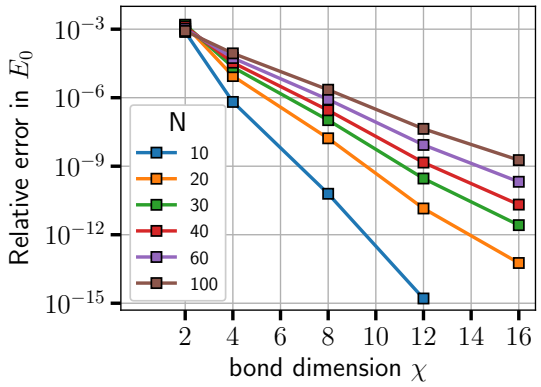

FIG. 7. The relative error in the calculated ground state energy, compared to the analytical result Eq. 40 goes down rapidly as the bond dimension χ is increased.

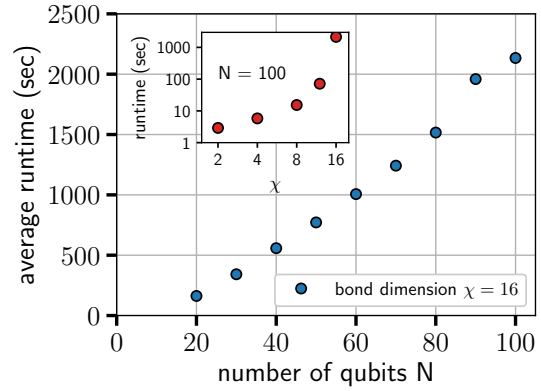

FIG. 8. For a fixed bond dimension χ, the runtime scales linearly as the number of particles N is increased. Inset: For a fixed N, the runtime is polynomial $\mathcal{O}(\chi^{4+})$ as a funtion of bond dimension.

The performance of the implementation is plotted for the transverse field Ising model in Fig. 8. As discussed in the last oparagraph of Sec. IV, the expected time

complexity is $\mathcal{O}(N\chi^{4+})$. The MPS method thus allows us to find the eigenvector of a very sparse and local $2^N \times 2^N$ Hamiltonian in a time that is linear in N.

VII. CONCLUSION

In this report, I have introduced the matrix-product states formalism. Within this formalism, there are many methods to represent quantum states and operators, and clever ways of taking advantage of the sparsity and locality of quantum systems to do approximate calculations with high fidelity. The variational ground state search is one such method, whose underlying algorithms are discussed in Sections III and IV. The python implementation is tested for the Heisenberg model and transverse field Ising models and achieves satisfactory results – the bottomline is that the grund state energy of a 100 qubit chain can be found to over 99.99999% accuracy using a bond dimension $\chi = 16$. Nominally, this state is a vector in a Hilbert space of dimension $2^{100} = 1.27 \times 10^{30}$, but with $\chi = 16$ is represented using only 51200 parameters.

This method can be used more generally for finding the ground states of other spin Hamiltonians, and the code includes a provision for customizing the parameters of the Hamiltonian. MPS based techniques span a broad range of algorithms, not only to find ground states, but also to find excited states, solve for time-evolution, test for phase transitions in toy models of interacting electrons, and so on. They are currently at the forefront of many areas of research in quantum computing and condensed matter physics.

I look forward to building on this piece of code and using it in my research in the future.

[1] R. Shankar, "Principles of Quantum Mechanics", *Springer US* (1994).

[2] Michael A. Nielsen and Isaac L. Chuang, "Quantum Computation and Quantum Information", *Cambridge University Press* (2010).

[3] Ullrich Schollwöck, "The density-matrix renormalization group in the age of matrix product states", *Annals of Physics*, 326(1), 96-192 (2011).

[4] Roman Orús, "A practical introduction to tensor networks: Matrix product states and projected entangled pair states", *Annals of Physics*, 349, 117-158 (2014).

[5] Subir Sachdev, "Quantum Phase Transitions", *Cambridge University Press* (2011).