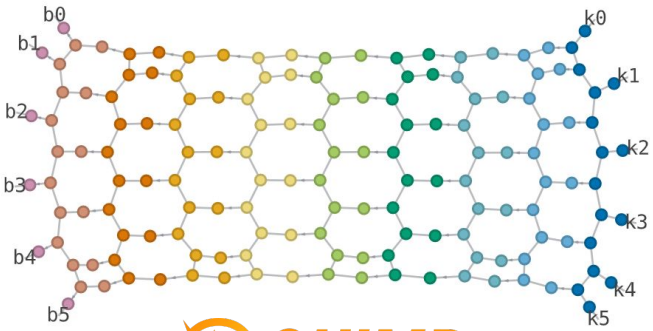
Simulating quantum circuits with tensor networks

Niccolò Baldelli & Joana Fraxanet



QUIMB

Outline

1st part: Introduction to MPS

Drawbacks of statevector simulation, tensor network notation, building an MPS exactly, cutting the bond dimension and connection to entanglement (area law).

2nd part: Simulation of quantum circuits using MPS

How to build quantum circuits with QUIMB and obtain wavefunction amplitudes, local expectation values and sampling. As an example, we will build Bell/GHZ states.

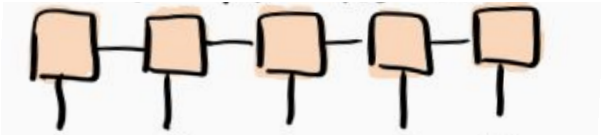
3rd part: Hands-on: building more entangled states

Let's build a random quantum circuit and study the scaling of entanglement and bond dimension. What happens if instead we use a circuit with physical meaning?

Motivation

Why Tensor Networks (TN)?

TN are a very efficient approximation to quantum states with low entanglement.

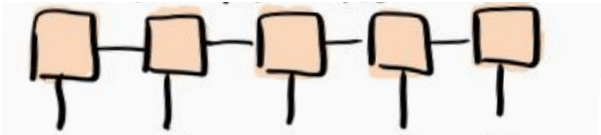


The type of states that can be approximated with TN are actually very important in quantum physics.

Motivation

Why Tensor Networks (TN)?

TN are a very efficient approximation to quantum states with low entanglement.

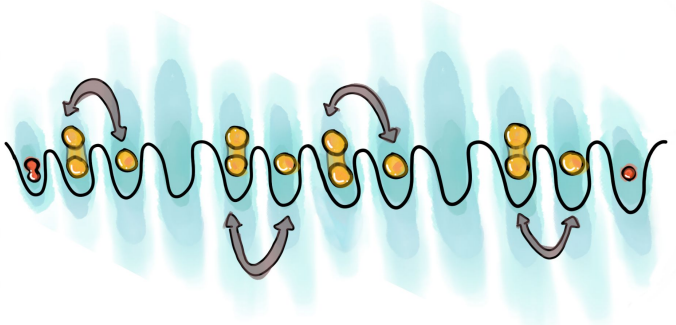


The type of states that can be approximated with TN are actually very important in quantum physics.

For example, TN they have become central to simulate large many-body quantum systems.

Many useful algorithms have been developed:

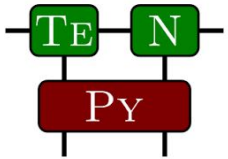
DMRG, TEBD, etc.



Motivation

But TN are not only used to simulate many-body quantum systems!

They have many other applications. For example, they are used in machine learning or they can be used to simulate quantum circuits.



Tenet.jl

Outline

1st part: Introduction to MPS

Drawbacks of statevector simulation, tensor network notation, building an MPS exactly, cutting the bond dimension and connection to entanglement (area law).

2nd part: Simulation of quantum circuits using MPS

How to build quantum circuits with QUIMB and obtain wavefunction amplitudes, local expectation values and sampling. As an example, we will build Bell/GHZ states.

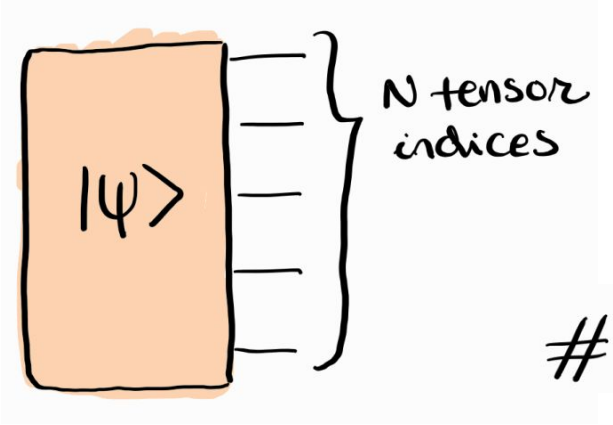
3rd part: Hands-on: building more entangled states

Let's build a random quantum circuit and study the scaling of entanglement and bond dimension.
What happens if instead we use a circuit with physical meaning?

Drawbacks of statevector simulation of QC

Storage of quantum states

$$|\Psi\rangle = \sum_{s_1,\ldots,s_N} c^{s_1,\ldots,s_N} |s_1, \ldots, s_N\rangle$$



Curse of dimensionality:

$$|\Psi\rangle \in \mathbb{C}^{d^L}$$

number of particles

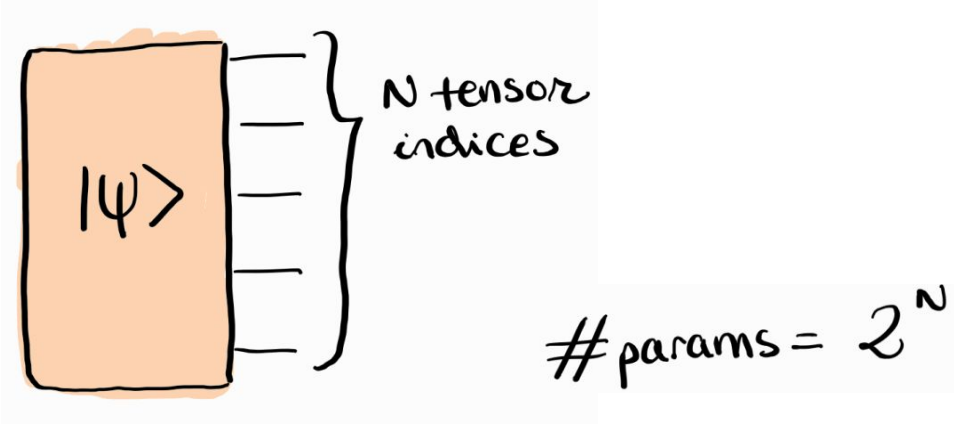
degrees of freedom (spins d=2)

L	Storage
20	16 MB
29	8 GB
50	10,000 TB
100	10^18 TB

Drawbacks of statevector simulation of QC

Storage of quantum states

$$|\Psi\rangle = \sum_{s_1,\ldots,s_N} c^{s_1,\ldots,s_N} |s_1, \ldots, s_N\rangle$$



Curse of dimensionality:

$$|\Psi\rangle \in \mathbb{C}^{d^L}$$

number of particles

degrees of freedom

(spins d=2)

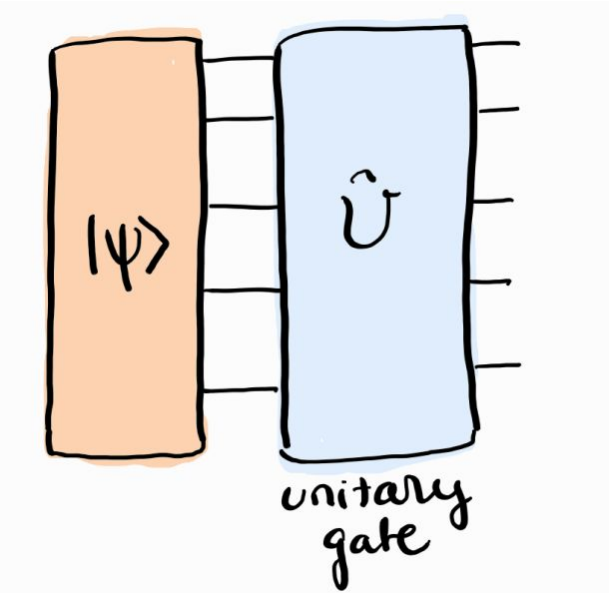
L	Storage
20	16 MB
29	8 GB
50	10,000 TB
100	10^18 TB

Drawbacks of statevector simulation of QC

Scaling with time (applying gates)

$$\hat{U} = \sum_{\substack{s_1,\ldots,s_N \\ s'_1,\ldots,s'_N}} a^{s'_1,\ldots,s'_N}_{s_1,\ldots,s_N} |s_1, \ldots, s_N\rangle\langle s'_1, \ldots, s'_N| \longrightarrow \hat{U}|\Psi\rangle$$

matrix multiplication

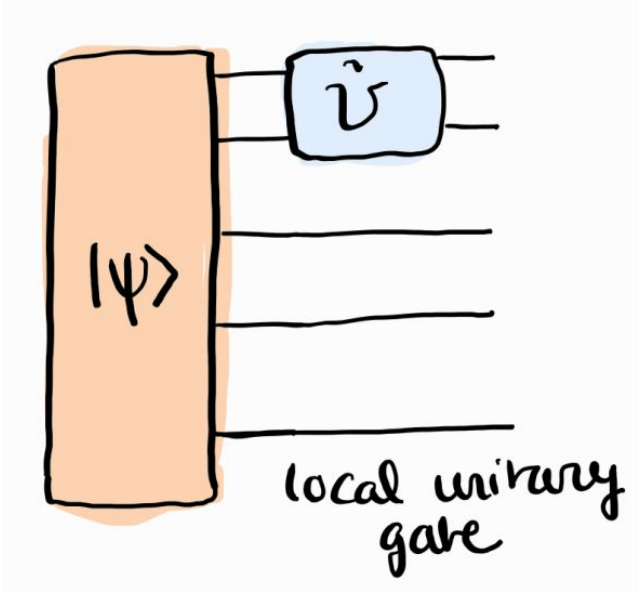


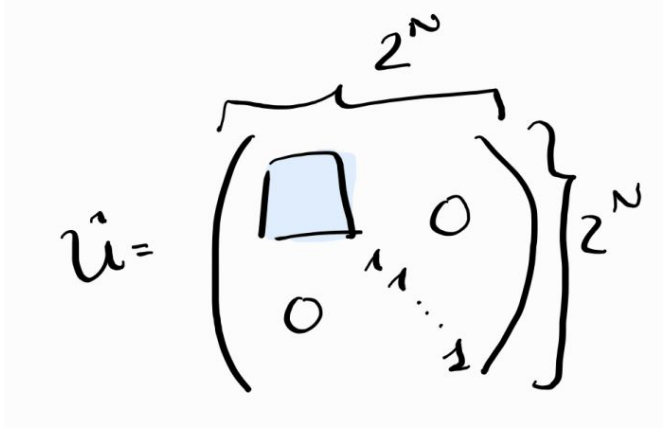
$$\#\text{params} = 2^{2N}$$

Drawbacks of statevector simulation of QC

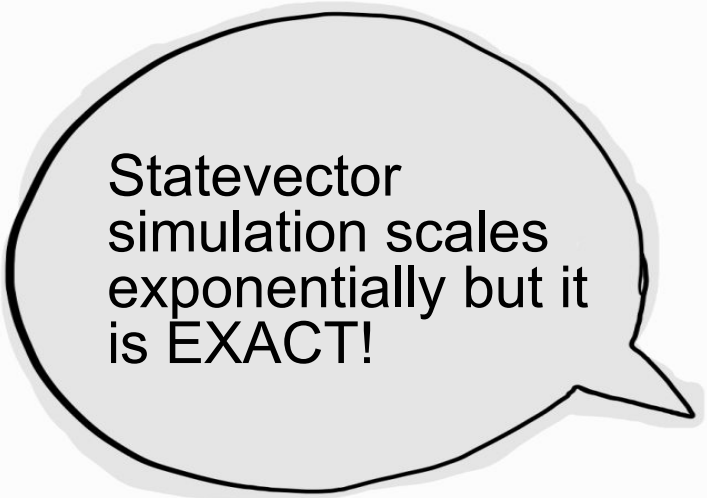
Scaling with time (applying gates)

$$\hat{U} = \sum_{\substack{s_1,\ldots,s_N \\ s'_1,\ldots,s'_N}} a^{s'_1,\ldots,s'_N}_{s_1,\ldots,s_N} |s_1, \ldots, s_N\rangle\langle s'_1, \ldots, s'_N| \quad \longrightarrow \quad \hat{U}|\Psi\rangle$$



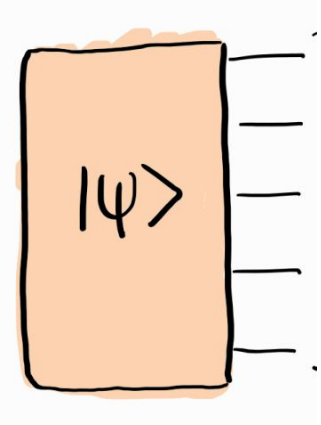


Drawbacks of statevector simulation of QC

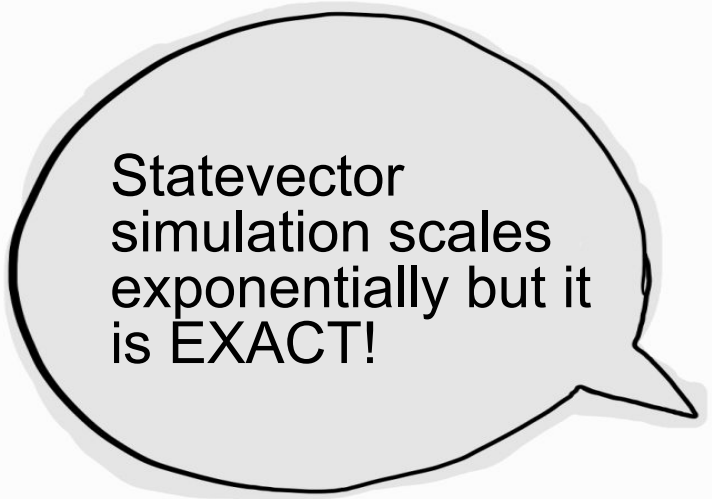


Drawbacks of statevector simulation of QC

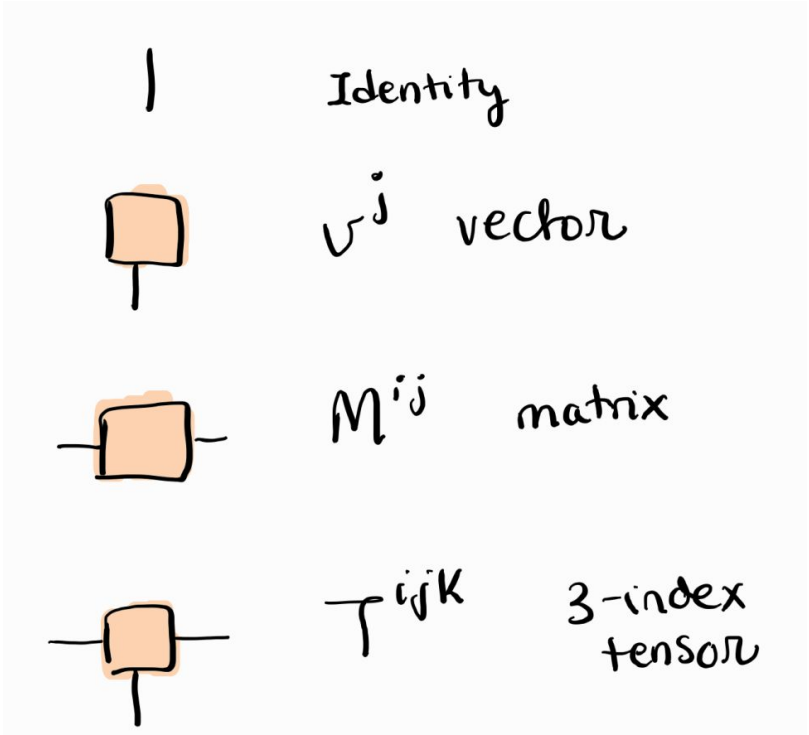
We want to find a good compression of the wavefunction



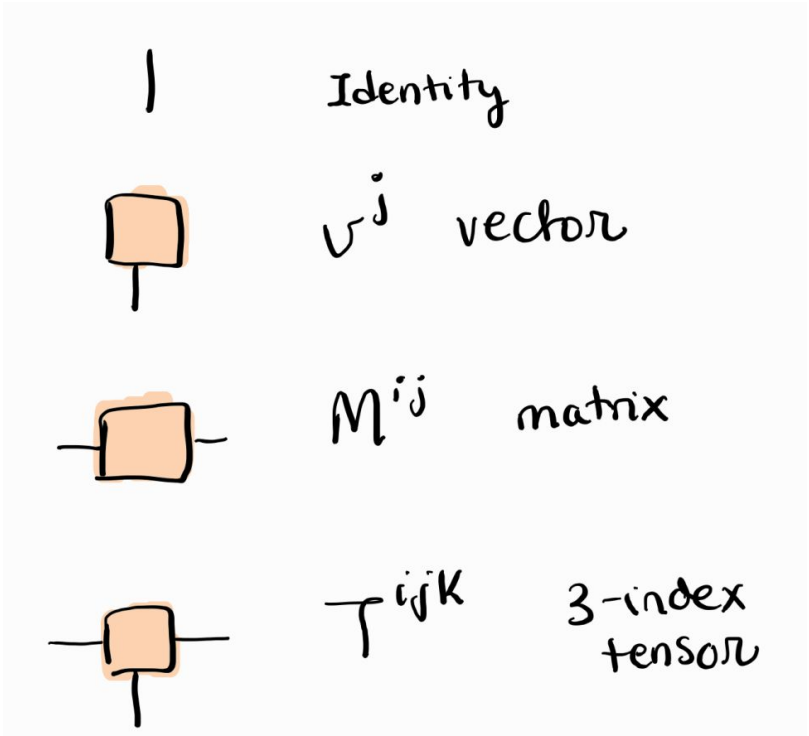
$$= \sum_{s_1,\ldots,s_N} c^{s_1,\ldots,s_N} \left| s_1, \ldots, s_N \right\rangle$$



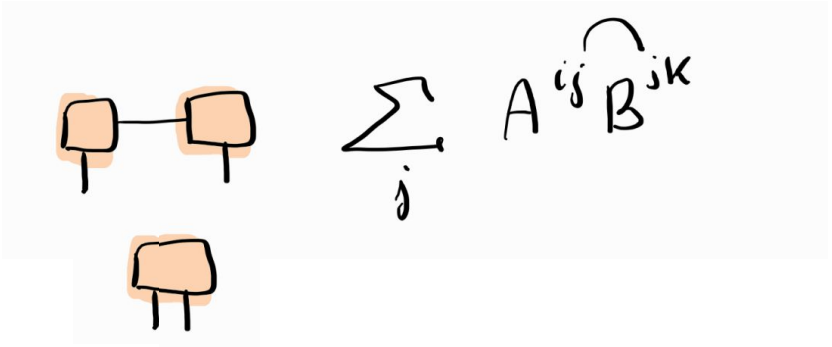
First, let's introduce tensor notation



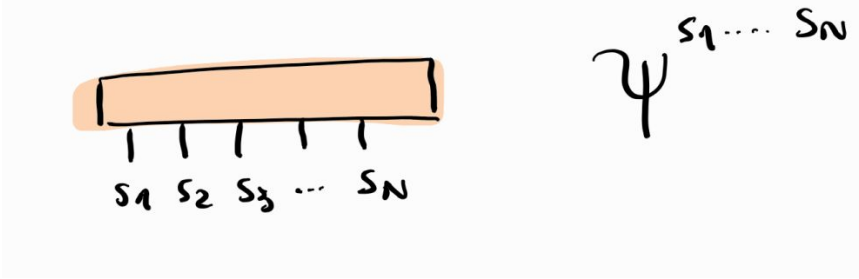
First, let's introduce tensor notation



Contraction of a bond:

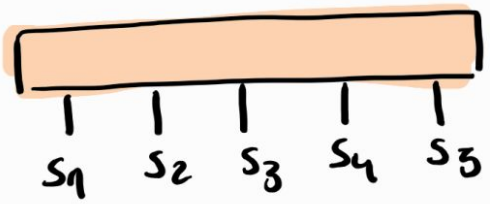


Physical indices:



Building an MPS exactly

Let's consider we have the wavefunction of a system of 5 spins:

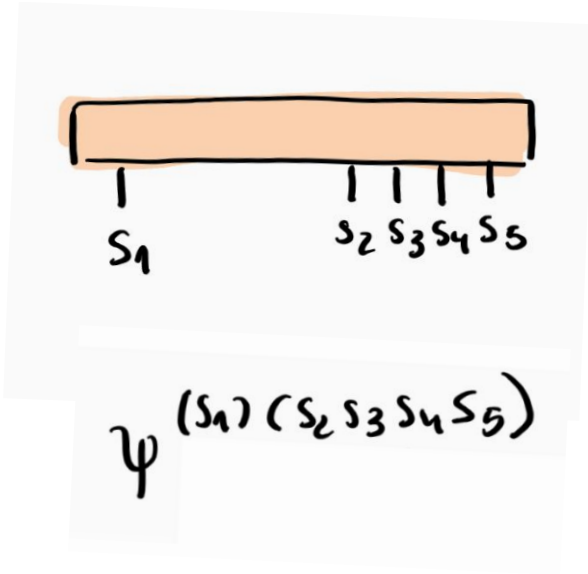


(spin d=2)

$$|\psi\rangle = \sum_{s_1 \cdots s_5} \psi^{s_1 s_2 s_3 s_4 s_5} |s_1 s_2 s_3 s_4 s_5\rangle$$

Building an MPS exactly

1. The first step is to re-group the indices. We separate the first spin:



we re-shape the tensor with size 2^5 into a matrix of size 2×2^4.

Building an MPS exactly

2. Next, we implement the **Singular Value Decomposition** (SVD):

Given a *n x m* matrix *M*,

$$M = USV^\dagger$$

where

U	*m* x min(*m*,*n*)	$U^\dagger U = Id$
S	min(*m*,*n*) x min(*m*,*n*)	positive diagonal matrix
$V^\dagger$	min(*m*,*n*) x *n*	$V^\dagger V = Id$

Building an MPS exactly

2. Next, we implement the Singular Value Decomposition (SVD):

$$\psi^{(s_1)(s_2 s_3 s_4 s_5)} = U^{s_1 \alpha_1} \lambda^{\alpha_1} \left(V^{\dagger}\right)^{\alpha_1 s_2 s_3 s_4 s_5}$$

$2 \times \chi_1$ $\quad \chi_1 \times \chi_1$ $\quad \chi_1 \times 2^4$

The dimension χ_1 is called the **bond dimension**.

$\chi_1 \leq 2$

we will talk about it later

Building an MPS exactly

2. Next, we implement the Singular Value Decomposition (SVD):

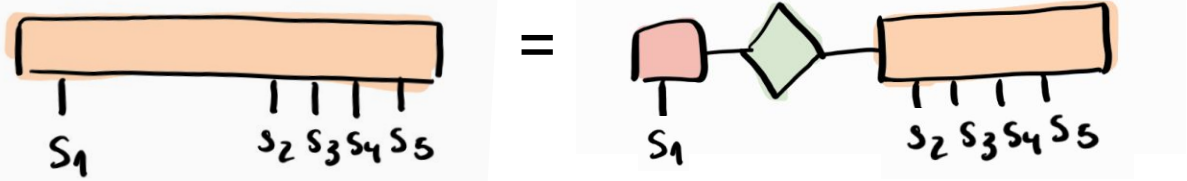
$$\psi^{(s_1)(s_2 s_3 s_4 s_5)} = U^{s_1 \alpha_1} \lambda^{\alpha_1} \left(V^{\dagger}\right)^{\alpha_1 s_2 s_3 s_4 s_5}$$

$2 \times \chi_1$ $\quad \chi_1 \times \chi_1$ $\quad \chi_1 \times 2^4$

$\chi_1 \leq 2$

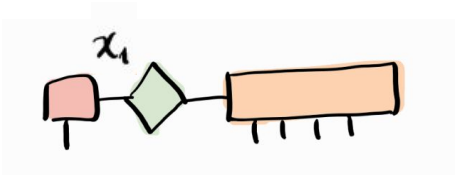
we will talk about it later

The dimension χ_1 is called the **bond dimension**.



Building an MPS exactly

3. Iterate over the chain:



$$U^{s_1 \alpha_1} \lambda^{\alpha_1} (V^\dagger)^{\alpha_1 s_2 s_3 s_4 s_5}$$

$$U^{(\alpha_1 s_2)(s_3 s_4 s_5)}$$

we re-shape the tensor to

$2\chi_1 \times 2^3$ $(= 2^2 \times 2^3)$

=

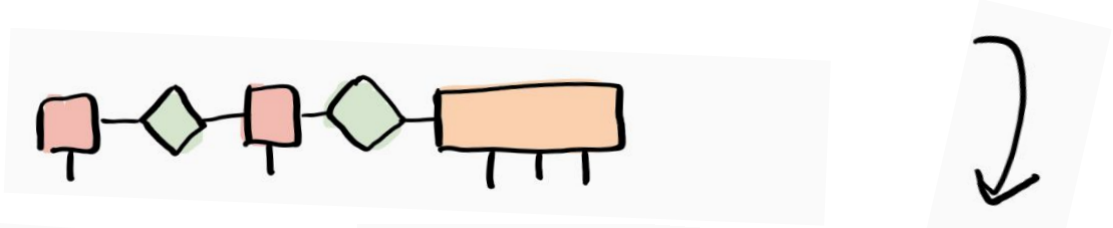
$$U^{s_1 \alpha_1} \lambda^{\alpha_1} U^{\alpha_1 s_2 \alpha_2} \lambda^{\alpha_2} (V^\dagger)^{\alpha_2 s_3 s_4 s_5}$$

Note that now:

$\chi_2 \leq \chi_1 2 \leq 2^2$

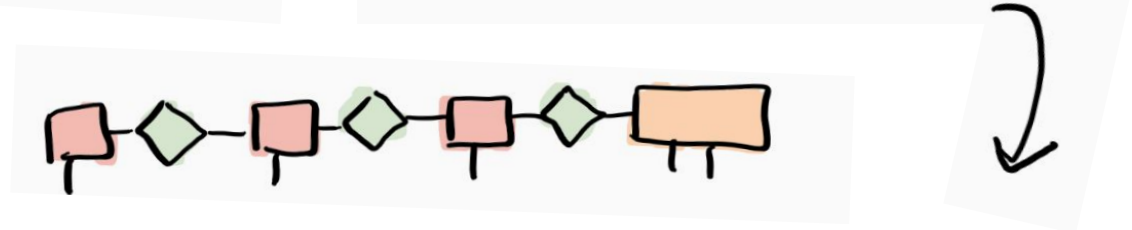
Building an MPS exactly

3. Iterate over the chain:



$$\psi^{s_1 s_2 s_3 s_4 s_5} = U^{s_1 \alpha_1} \lambda^{\alpha_1} U^{\alpha_1 s_2 \alpha_2} \lambda^{\alpha_2} V^{\alpha_2 s_3 s_4 s_5}$$

$\chi_2 \leq \chi_1 2 \leq 2^2$

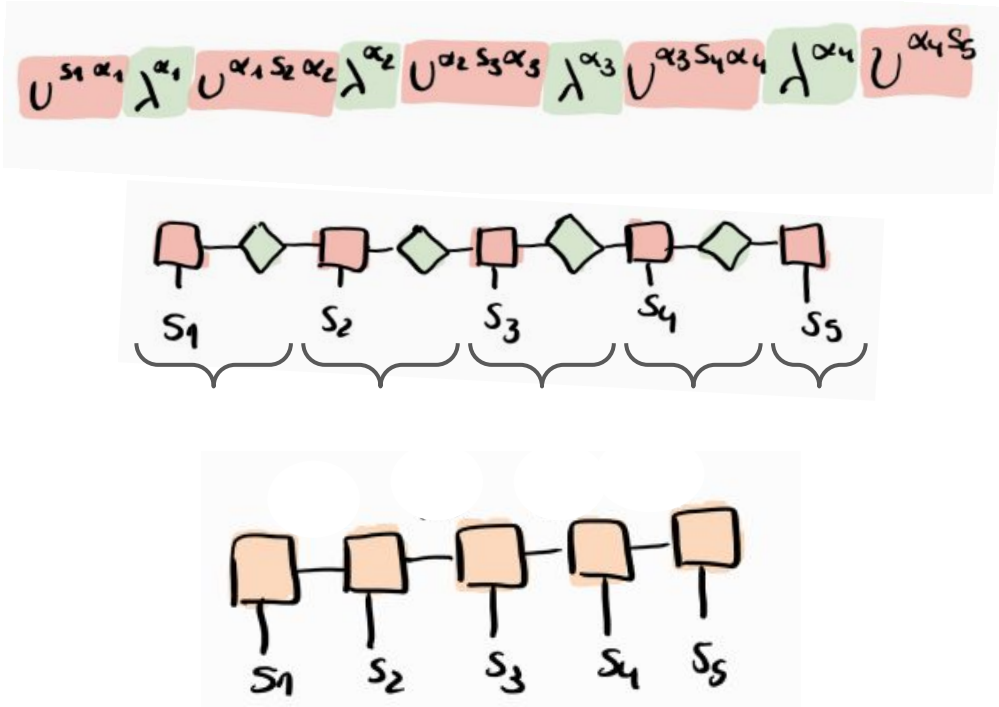


$$\psi^{s_1 s_2 s_3 s_4 s_5} = U^{s_1 \alpha_1} \lambda^{\alpha_1} U^{\alpha_1 s_2 \alpha_2} \lambda^{\alpha_2} U^{\alpha_2 s_3 \alpha_3} \lambda^{\alpha_3} \ldots$$

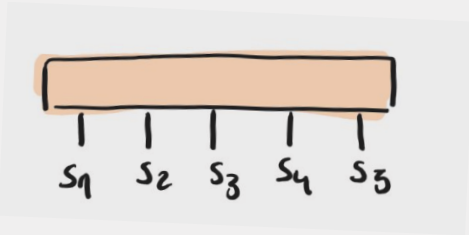
$\chi_3 \leq \min(2\chi_2, 2^2) = 2^2$

Building an MPS exactly

3. Absorb the diagonal matrices from SVD:



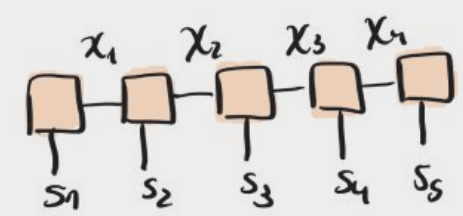
Building an MPS exactly



$$|\psi\rangle = \sum_{s_1 \cdots s_5} \psi^{s_1 s_2 s_3 s_4 s_5} |s_1 s_2 s_3 s_4 s_5\rangle$$

params ~ $d^N = 2^5$

=



$$\sum_{s_1 \cdots s_5} A^{s_1}_{\alpha_1} A^{s_2}_{\alpha_1 \alpha_2} A^{s_3}_{\alpha_2 \alpha_3} A^{s_4}_{\alpha_3 \alpha_4} A^{s_5}_{\alpha_4} |s_1 s_2 s_3 s_4 s_5\rangle$$

params $\lesssim dN(\chi_{max})^2 = 2 \cdot 5 \cdot (2)^4$

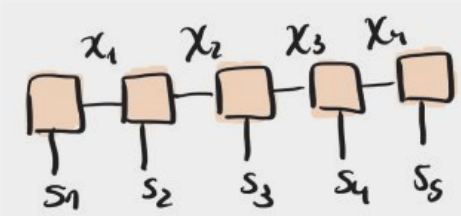
We just found a different expression for the wavefunction.
We did this **exactly**: the number of parameters is the same!

Building an MPS exactly

$$|\psi\rangle = \sum_{s_1 \cdots s_5} \psi^{s_1 s_2 s_3 s_4 s_5} |s_1 s_2 s_3 s_4 s_5\rangle$$

params ~ $d^N = 2^5$

$=$

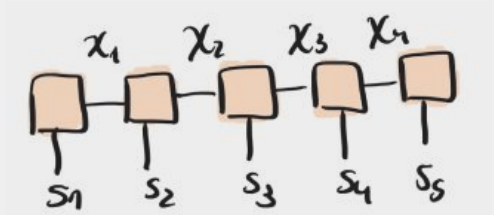


$$\sum_{s_1 \cdots s_5} A^{s_1}_{\alpha_1} A^{s_2}_{\alpha_1 \alpha_2} A^{s_3}_{\alpha_2 \alpha_3} A^{s_4}_{\alpha_3 \alpha_4} A^{s_5}_{\alpha_4} |s_1 s_2 s_3 s_4 s_5\rangle$$

params $\lesssim dN(\chi_{max})^2 = 2 \cdot 5 \cdot (2)^4$

BUT.. **THERE IS AN EFFICIENT WAY TO COMPRESS AN MPS!**

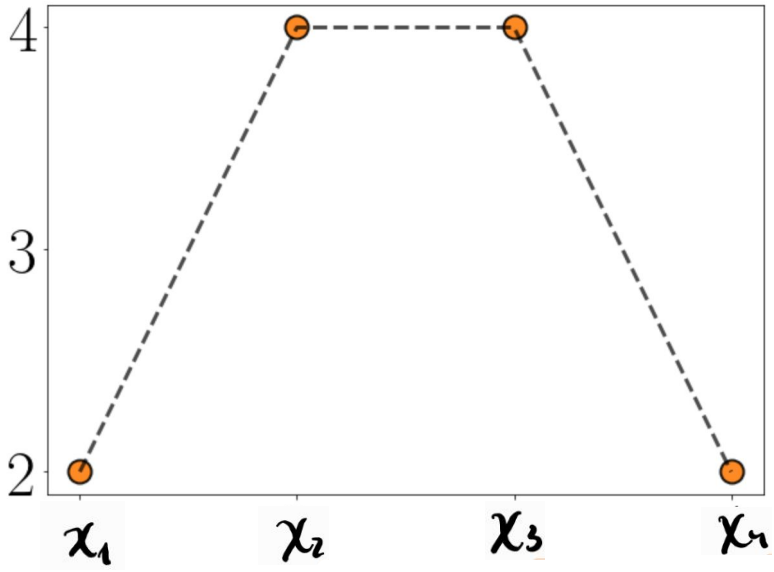
Cutting the bond dimension



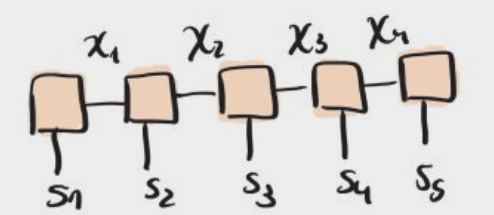
$$\sum_{s_1 \cdots s_5} A^{s_1}_{\alpha_1} A^{s_2}_{\alpha_1 \alpha_2} A^{s_3}_{\alpha_2 \alpha_3} A^{s_4}_{\alpha_3 \alpha_4} A^{s_5}_{\alpha_4} |s_1 s_2 s_3 s_4 s_5\rangle$$

Let's go back to the bond dimensions

For a system of 5 spins is not bad:



Cutting the bond dimension

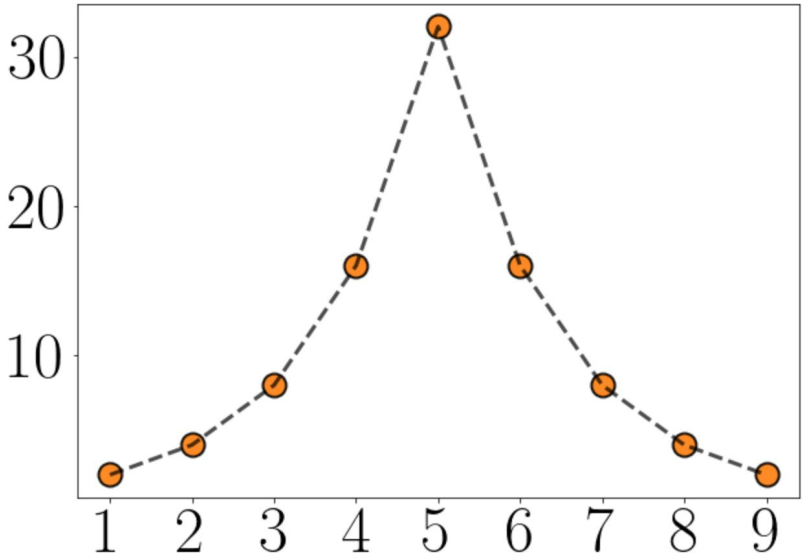


$$\sum_{s_1 \cdots s_5} A^{s_1}_{\alpha_1} A^{s_2}_{\alpha_1 \alpha_2} A^{s_3}_{\alpha_2 \alpha_3} A^{s_4}_{\alpha_3 \alpha_4} A^{s_5}_{\alpha_4} |s_1 s_2 s_3 s_4 s_5\rangle$$

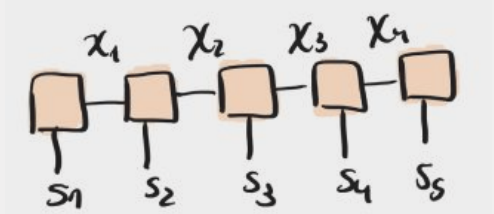
Let's go back to the bond dimensions

But it gets worse exponentially.

For 10 spins:



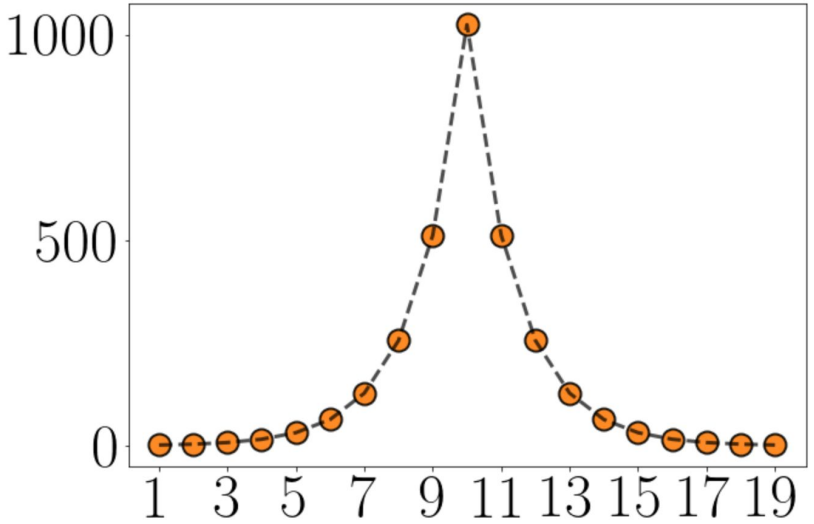
Cutting the bond dimension



$$\sum_{s_1 \cdots s_5} A^{s_1}_{\alpha_1} A^{s_2}_{\alpha_1 \alpha_2} A^{s_3}_{\alpha_2 \alpha_3} A^{s_4}_{\alpha_3 \alpha_4} A^{s_5}_{\alpha_4} |s_1 s_2 s_3 s_4 s_5\rangle$$

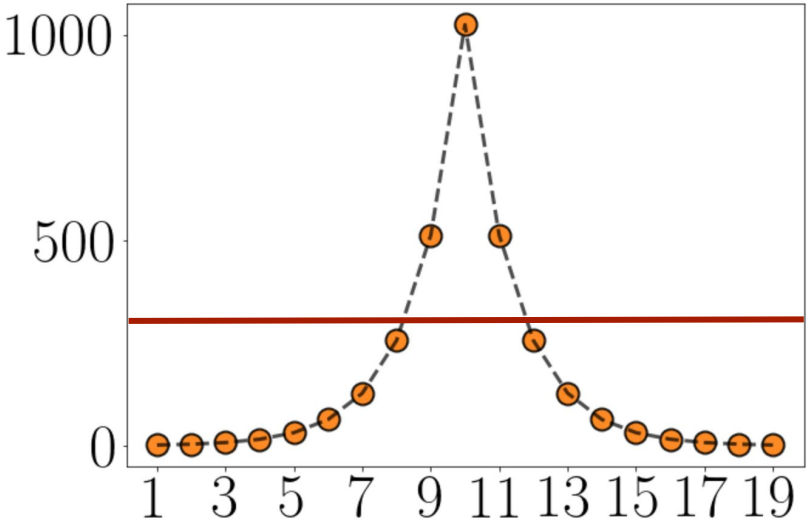
Let's go back to the bond dimensions

For 20 spins:



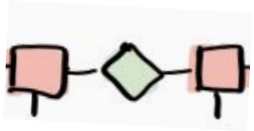
Cutting the bond dimension

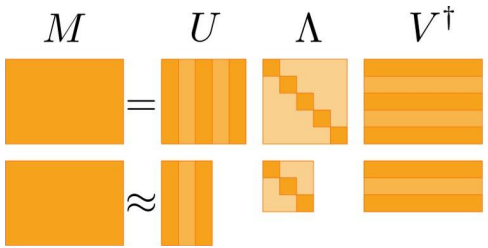
The solution is to manually fix a χ_{max} so that it does not increase exponentially.

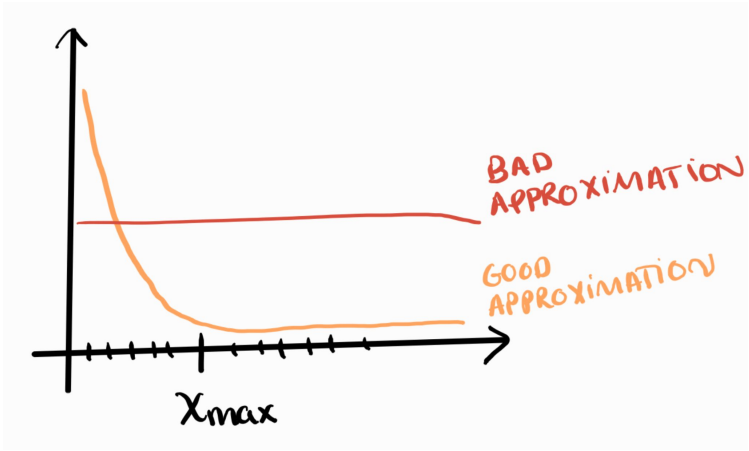


Cutting the bond dimension

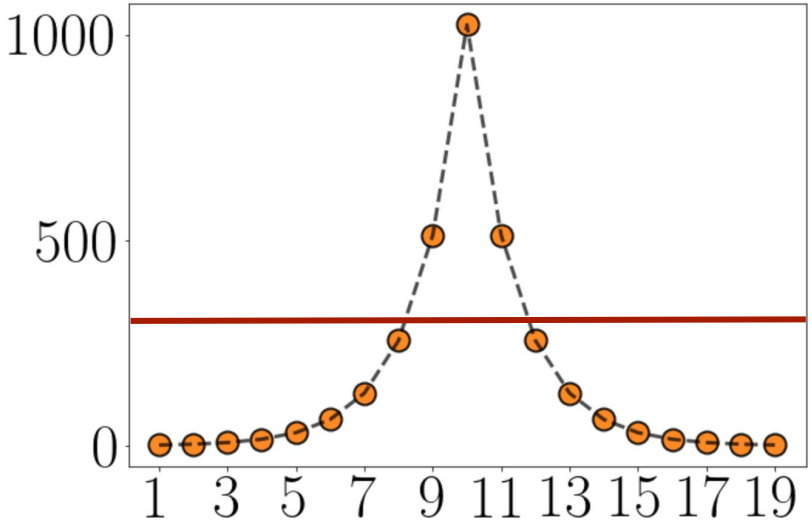
By cutting the bond dimension we get rid of the smallest singular values of a bond:





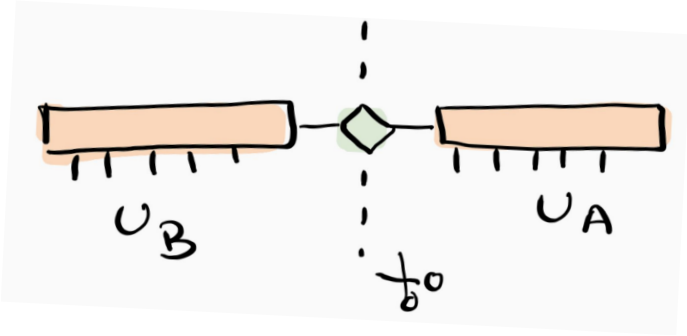


The solution is to manually fix a χ_{max} so that it does not increase exponentially.



The physical meaning of the bond dimension

The decay of the singular values is directly related to the entanglement entropy of the system!
If we cut the system in two…

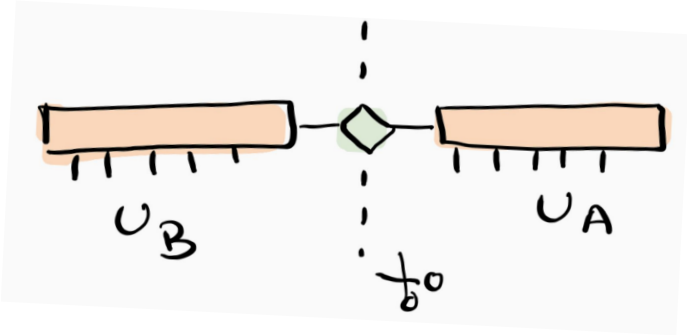


the reduced density matrix of one of the sides is defined as

$$\rho_A = tr_B[\rho] = \sum_\alpha \lambda_\alpha^2 |u_\alpha^A\rangle\langle u_\alpha^A| \qquad (\rho = |\psi\rangle\langle\psi|)$$

The physical meaning of the bond dimension

The decay of the singular values is directly related to the entanglement entropy of the system!
If we cut the system in two…



the reduced density matrix of one of the sides is defined as

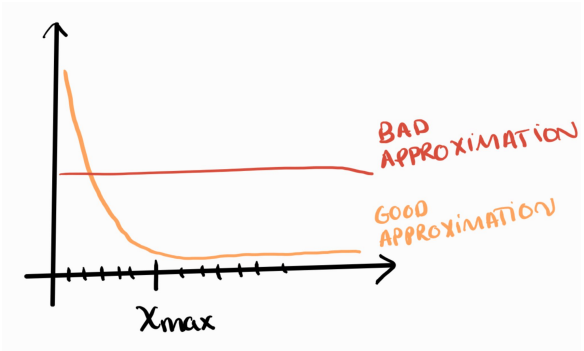
$$\rho_A = tr_B[\rho] = \sum_\alpha \lambda_\alpha^2 |u_\alpha^A\rangle\langle u_\alpha^A| \qquad (\rho = |\psi\rangle\langle\psi|)$$

Then, the bipartite entanglement entropy:

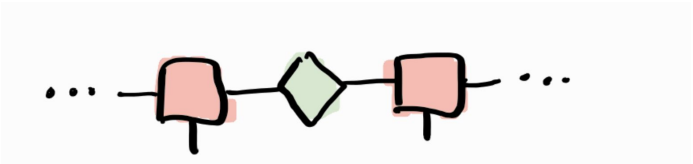
$$S(A|B) = -tr[\rho_A \log \rho_A] = -\sum_\alpha \lambda_\alpha^2 \log \lambda_\alpha^2$$

Entanglement growth: the area law

How does the entanglement entropy scale for quantum states that we are interested in?



The entanglement entropy for an MPS scales as ~N^0 (it's constant!)



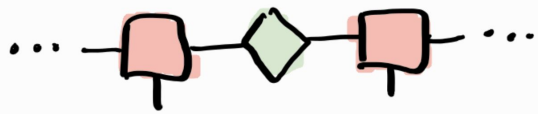
AREA LAW:

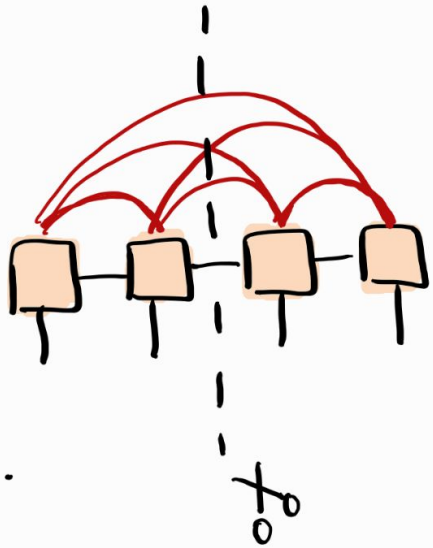
$$S \leq N^{D-1}$$

Entanglement growth: the area law

How does the entanglement entropy scale for quantum states that we are interested in?

The entanglement entropy for an MPS scales as ~N^0 (it's constant!)





AREA LAW:

$$S \leq N^{D-1}$$

long-range entanglement effect on the central bond

Entanglement growth: the area law

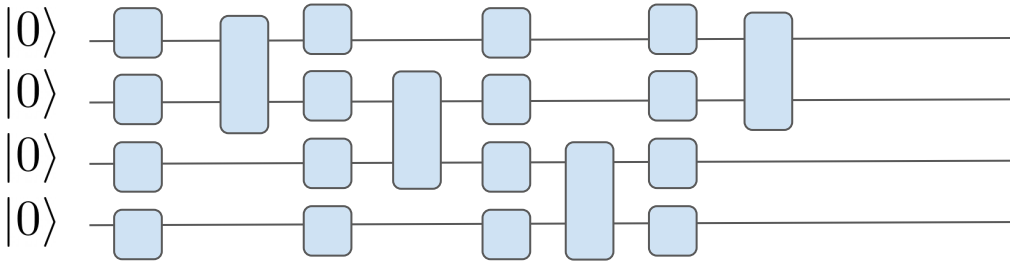
How does the entanglement entropy scale for quantum states that we are interested in?

The entanglement entropy for an MPS scales as ~N^0 (it's constant!)

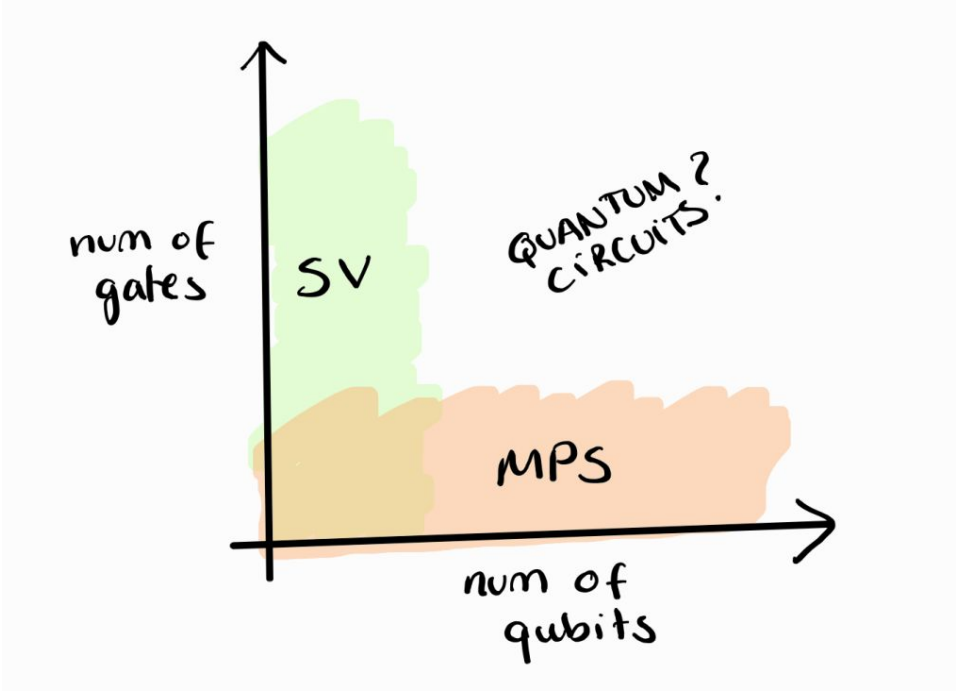
For a quantum circuit we usually:

- ➢ Start with a product state (no entanglement)
- ➢ Apply LOCAL gates

If the circuit is shallow, MPS will still give us a good representation.



Summary: Statevector vS MPS



Outline

1st part: Introduction to MPS

Drawbacks of statevector simulation, tensor network notation, building an MPS exactly, cutting the bond dimension and connection to entanglement (area law).

2nd part: Simulation of quantum circuits using MPS

How to build quantum circuits with QUIMB and obtain wavefunction amplitudes, local expectation values and sampling. As an example, we will build Bell/GHZ states.

3rd part: Hands-on: building more entangled states

Let's build a random quantum circuit and study the scaling of entanglement and bond dimension. What happens if instead we use a circuit with physical meaning?

Outline

1st part: Introduction to MPS

Drawbacks of statevector simulation, tensor network notation, building an MPS exactly, cutting the bond dimension and connection to entanglement (area law).

2nd part: Simulation of quantum circuits using MPS

How to build quantum circuits with QUIMB and obtain wavefunction amplitudes, local expectation values and sampling. As an example, we will build Bell/GHZ states.

3rd part:

Let's build a

What happe

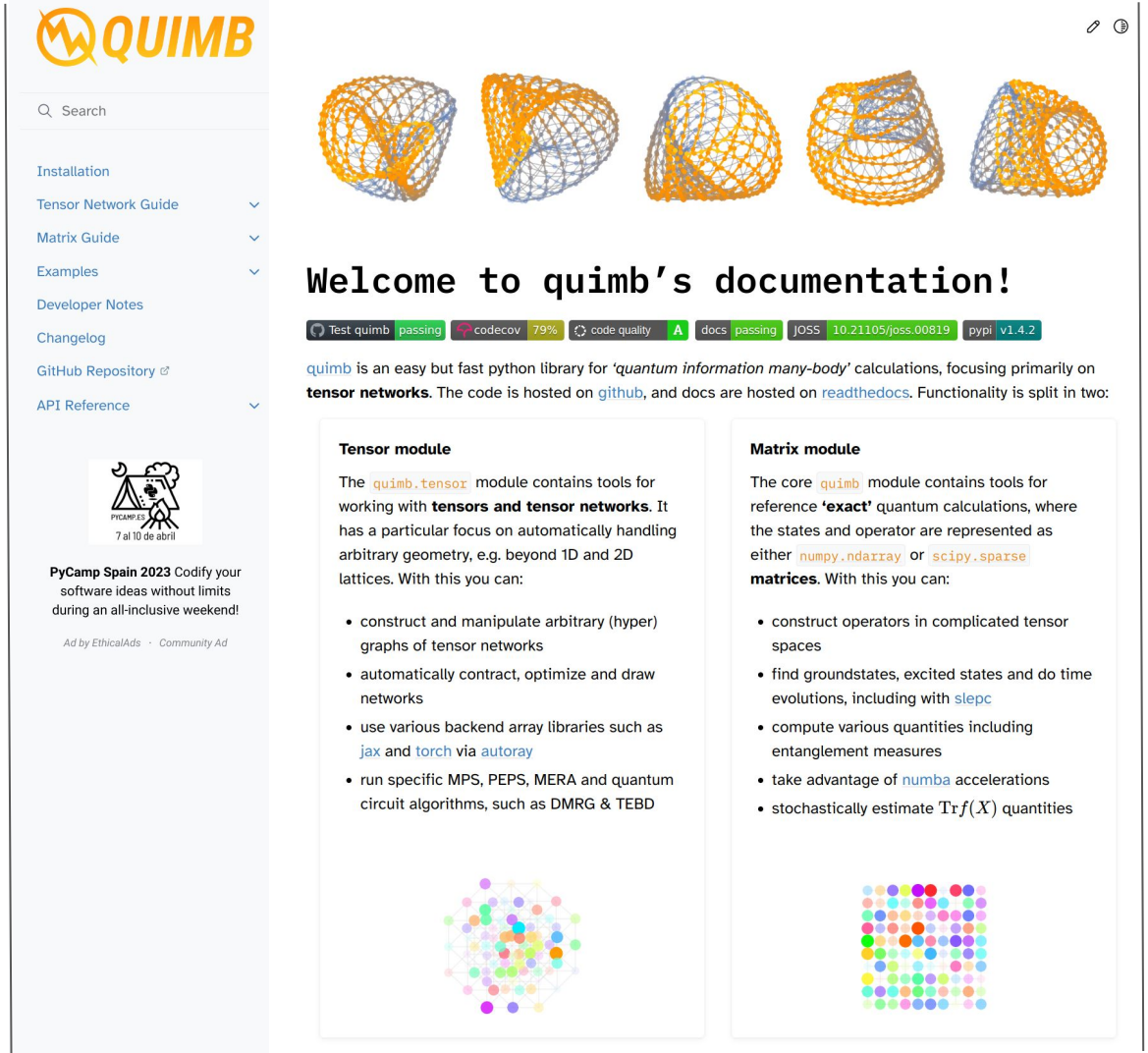
on.

https://colab.research.google.com/drive/1qhBSYJAQufrJIHY_SyRQ95tcFya0nAqO?usp=sharing

We will use QUIMB, a python library focused on tensor networks which allows us to simulate quantum circuits using **statevector** and **MPS**.

We will compare the scaling of the two methods when building a:

- ➔ Bell state
- ➔ GHZ state



We will use QUIMB, a python library focused on tensor networks which allows us to simulate quantum circuits using **statevector** and **MPS**.

We will compare the scaling of the two methods when building a:

- ➔ Bell state
- ➔ GHZ state

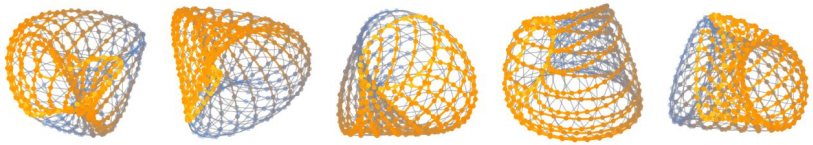
But first, let's look a bit into the library!

NOTEBOOK

QUIMB

Search

Installation
Tensor Network Guide
Matrix Guide
Examples
Developer Notes
Changelog
GitHub Repository
API Reference



Welcome to quimb's documentation!

Test quimb passing | codecov 79% | code quality A | docs passing | JOSS 10.21105/joss.00819 | pypi v1.4.2

quimb is an easy but fast python library for *'quantum information many-body'* calculations, focusing primarily on **tensor networks**. The code is hosted on github, and docs are hosted on readthedocs. Functionality is split in two:

Tensor module

The `quimb.tensor` module contains tools for working with **tensors and tensor networks**. It has a particular focus on automatically handling arbitrary geometry, e.g. beyond 1D and 2D lattices. With this you can:

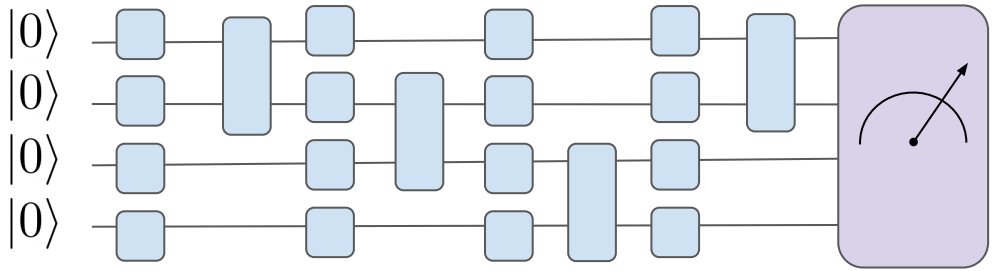
- construct and manipulate arbitrary (hyper) graphs of tensor networks
- automatically contract, optimize and draw networks
- use various backend array libraries such as jax and torch via autoray
- run specific MPS, PEPS, MERA and quantum circuit algorithms, such as DMRG & TEBD

Matrix module

The core `quimb` module contains tools for reference **'exact'** quantum calculations, where the states and operator are represented as either `numpy.ndarray` or `scipy.sparse` **matrices**. With this you can:

- construct operators in complicated tensor spaces
- find groundstates, excited states and do time evolutions, including with slepc
- compute various quantities including entanglement measures
- take advantage of numba accelerations
- stochastically estimate $\mathrm{Tr} f(X)$ quantities

How to build a quantum circuit using QUIMB



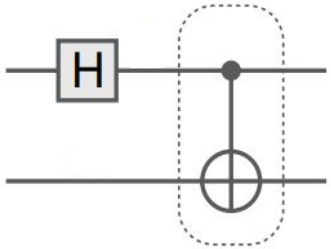
initial state (usually product state, easy to define)

set of parametrized local gates (rotations and entangling gates).

measure expectation values or sampling (we can also extract wave function amplitudes)

Starting with an example: building a Bell state

$$\frac{|00\rangle + |11\rangle}{2}$$

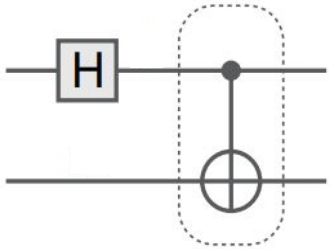


$$\mathsf{CNOT} = \sum_{ab} |a, a \oplus b\rangle\langle a, b|$$

$$\mathsf{H} = \frac{1}{\sqrt{2}} \sum_{ab} (-1)^{ab} |a\rangle\langle b|$$

Starting with an example: building a Bell state

$$\frac{|00\rangle + |11\rangle}{2}$$



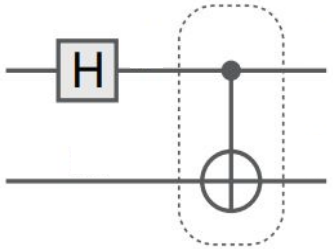
$$\text{CNOT} = \sum_{ab} |a, a \oplus b\rangle\langle a, b|$$

$$\text{H} = \frac{1}{\sqrt{2}} \sum_{ab} (-1)^{ab} |a\rangle\langle b|$$

Let's build it using statevector simulator first and then using MPS

Starting with an example: building a Bell state

$$\frac{|00\rangle + |11\rangle}{2}$$



Let's build it using statevector simulator first and then using MPS

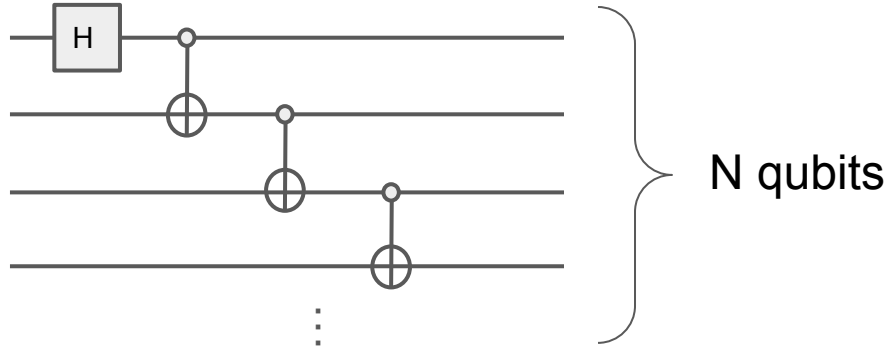
$$\text{CNOT} = \sum_{ab} |a, a \oplus b\rangle\langle a, b|$$

$$\text{H} = \frac{1}{\sqrt{2}} \sum_{ab} (-1)^{ab} |a\rangle\langle b|$$

We could build the MPS exactly. Entangling gates increase the bond dimension, but in this case only up to 2!

A bit more costly: building a GHZ state

$$\frac{|00..00\rangle + |11..11\rangle}{2}$$



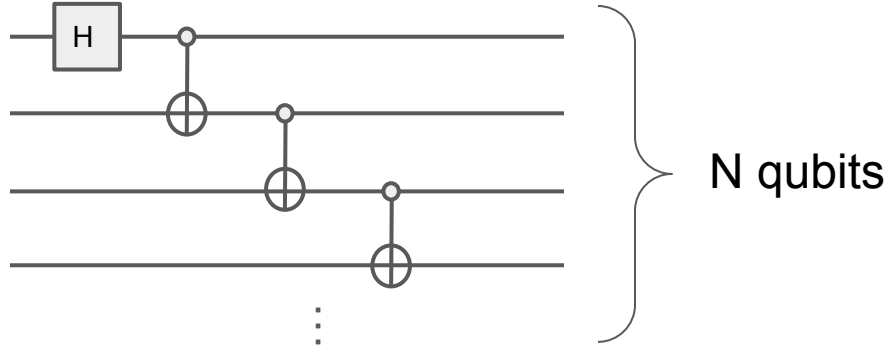
Now we increase the number of qubits!

Note that we'd have to work in a space of $2^N x 2^N$ matrices.

If we try to do this for 20 qubits with statevector, the computer cannot handle it.

A bit more costly: building a GHZ state

$$\frac{|00..00\rangle + |11..11\rangle}{2}$$



Now we increase the number of qubits!

Note that we'd have to work in a space of $2^N x 2^N$ matrices.

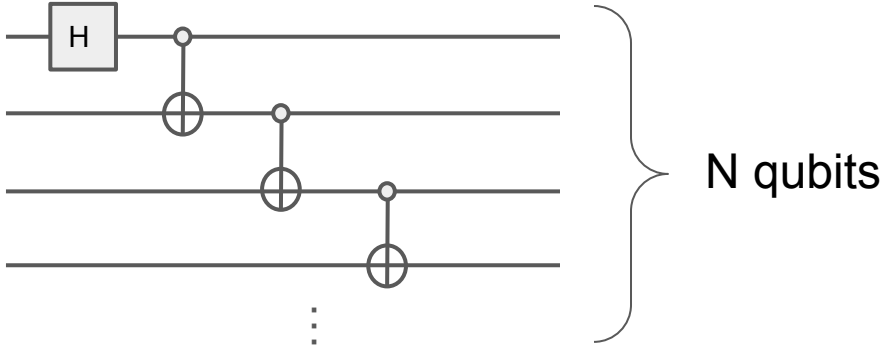
If we try to do this for 20 qubits with statevector, the computer cannot handle it.

There is still a trick we can use with statevector: sparse matrices!

NOTEBOOK

A bit more costly: building a GHZ state

$$\frac{|00..00\rangle + |11..11\rangle}{2}$$



Now we increase the number of qubits!

Note that we'd have to work in a space of $2^N x 2^N$ matrices.

If we try to do this for 20 qubits with statevector, the computer cannot handle it.

- The GHZ state has low entanglement and therefore the bond dimension stays low.
- Sparse statevector and exact MPS use the same number of parameters.

Outline

1st part: Introduction to MPS

Drawbacks of statevector simulation, tensor network notation, building an MPS exactly, cutting the bond dimension and connection to entanglement (area law).

2nd part: Simulation of quantum circuits using MPS

How to build quantum circuits with QUIMB and obtain wavefunction amplitudes, local expectation values and sampling. As an example, we will build Bell/GHZ states.

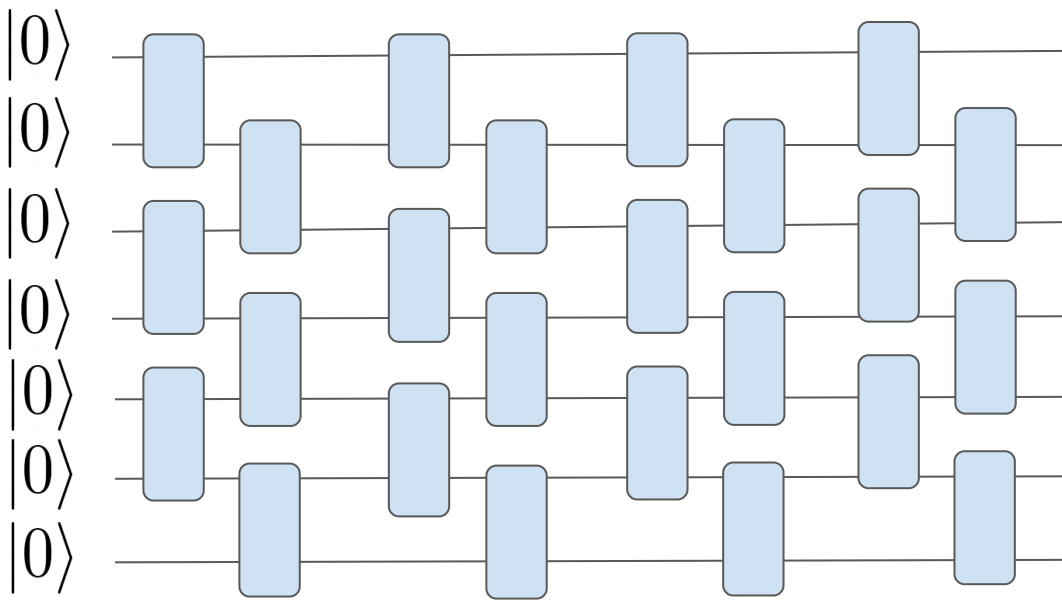
3rd part: Hands-on: building more entangled states

Let's build a random quantum circuit and study the scaling of entanglement and bond dimension. What happens if instead we use a circuit with physical meaning?

Random circuits

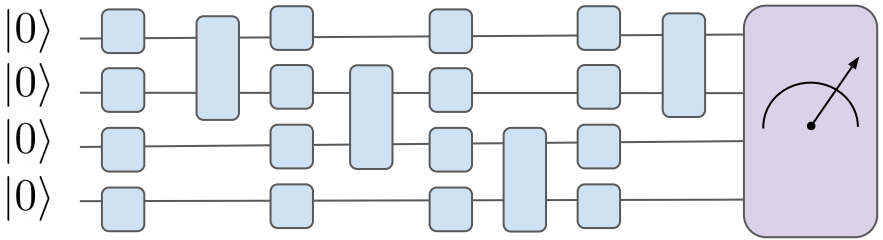
Why do we need MPS when we can use sparse matrices? To answer this question, we need to generate a state with more entanglement.

Then, both dense and sparse work badly.



Going back to… measurements

But first, let's briefly discuss how to perform measurements.



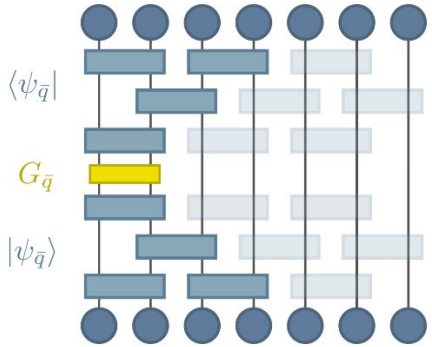
initial state (usually product state, easy to define)

set of parametrized local gates (rotations and entangling gates).

measure expectation values or sampling (we can also extract wave function amplitudes)

Going back to… measurements

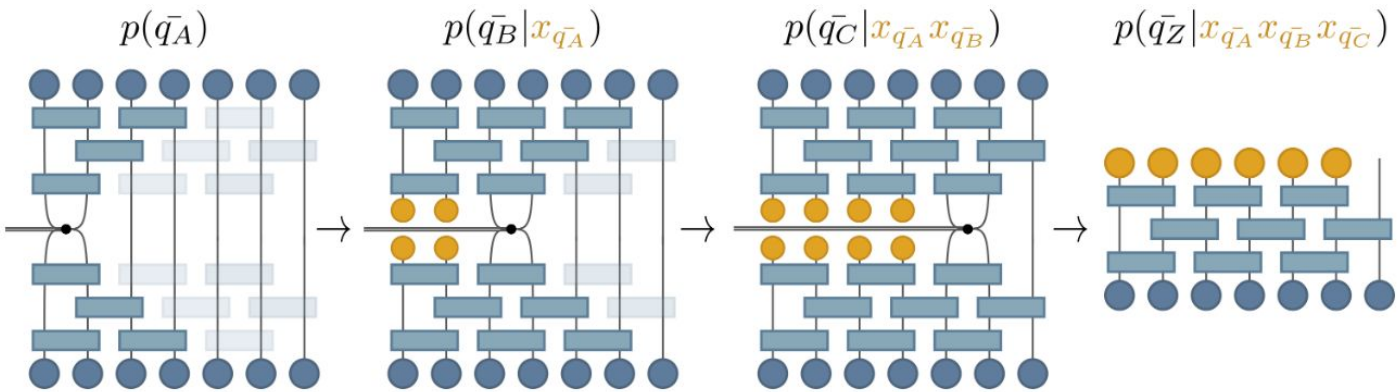
Computing a local expectation value



$$\langle\psi_{\bar{q}}|G_{\bar{q}}|\psi_{\bar{q}}\rangle$$

Going back to… measurements

Sampling

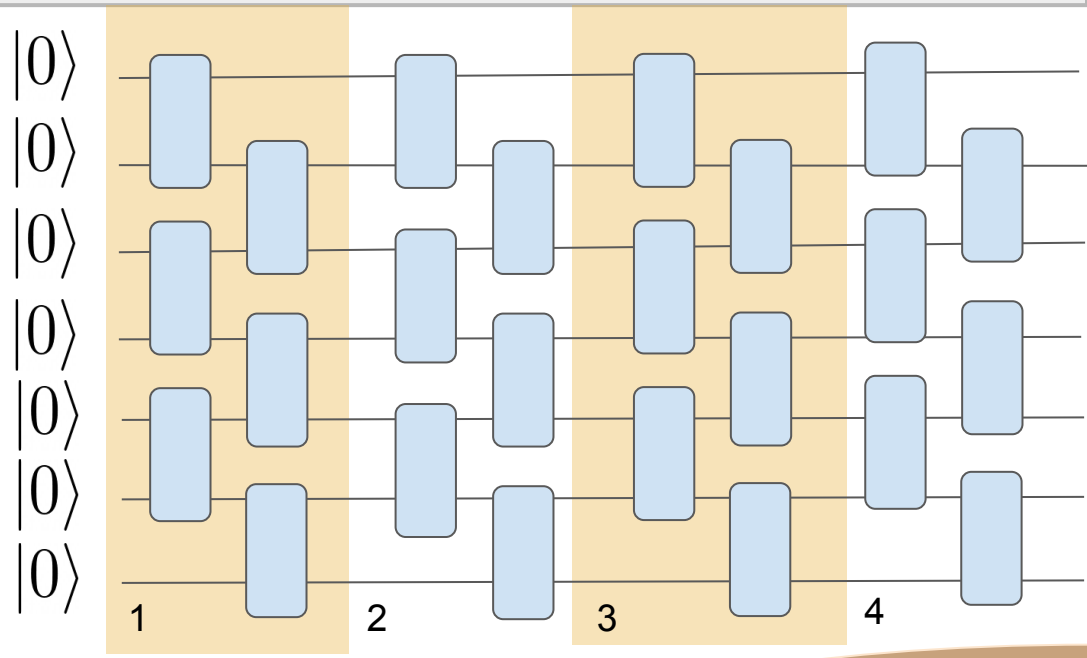


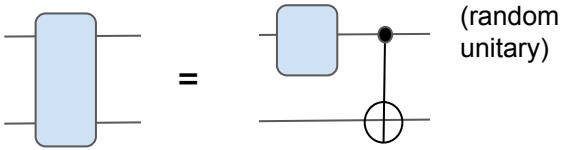
$$p(\bar{q_Z}|x_{\bar{q_A}}x_{\bar{q_B}}\ldots) = |\langle x_{\bar{q_A}}x_{\bar{q_B}}\ldots|\psi\rangle|^2$$

Random circuits

We are ready to study how does the MPS simulation perform in terms of num of layers.

We will look at max bond dimension, entanglement, fidelities, order parameters, etc.



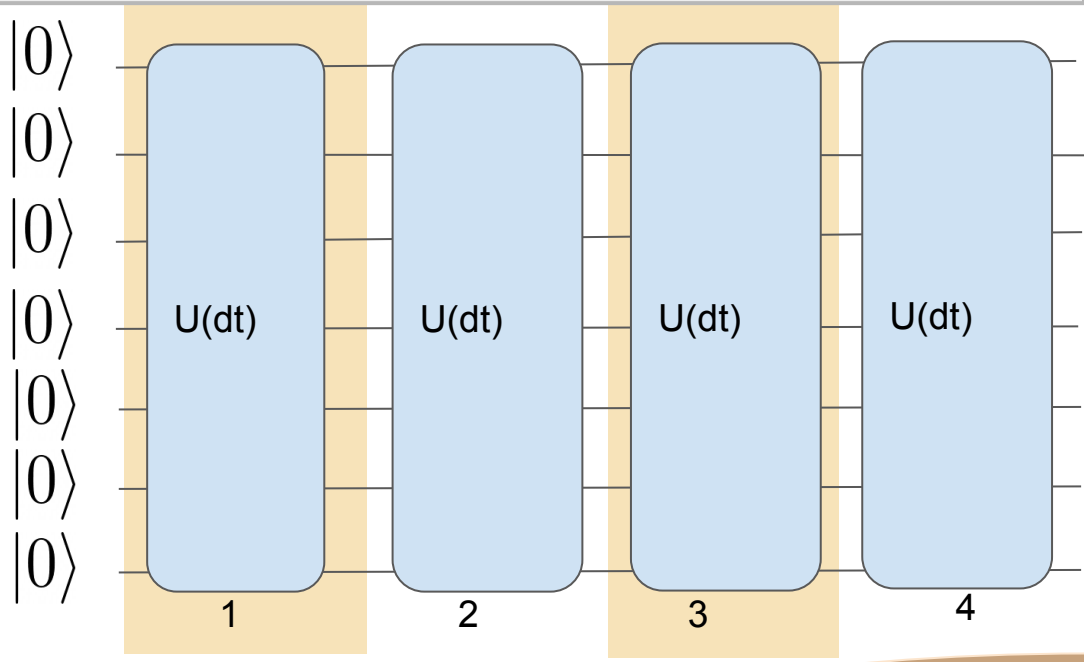


NOTEBOOK

Quench in the Ising model

Now we look at a physically relevant example.

$$H = -J \sum \sigma_i^z \sigma_{i+1}^z + g \sum \sigma_i^x .$$



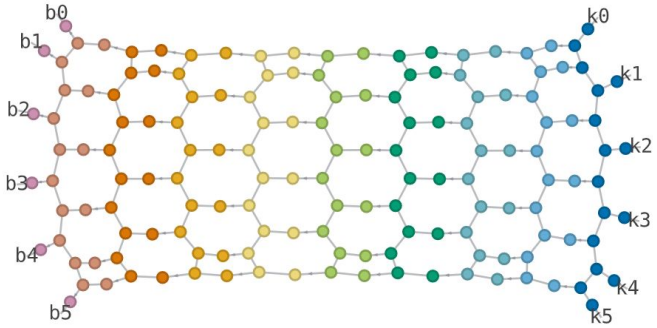
$$U(\delta t) = e^{-i\delta t J \sum \sigma_i^z \sigma_{i+1}^z + i\delta t g \sum \sigma_i^x} \sim \underbrace{\prod_i e^{-i\delta t J \sigma_i^z \sigma_{i+1}^z} \prod_i e^{i\delta t g \sigma_i^x}}_{\text{2-qubit gates} \quad \text{1-qubit gates}} .$$

NOTEBOOK

Outline

1st part: Introduction to

Drawbacks of statevector simula MPS exactly, cutting the bond dimension and connection to ent



2nd part: Simulation of

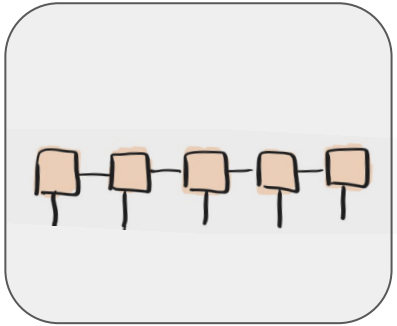
QUIMB

Spoiler: There are other types of tensor networks!

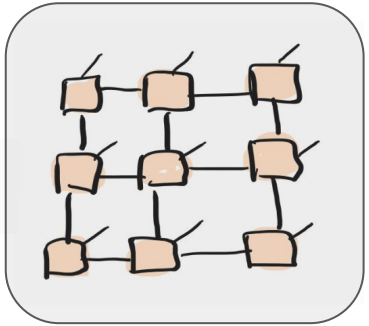
Tree-tensor networks, PEPS (2D)... and they present some advantages.
But working with them is not as easy: contraction of the tensors becomes a problem!

Let's build a random quantum circuit and study the scaling of entanglement and bond dimension.
What happens if instead we use a circuit with physical meaning?

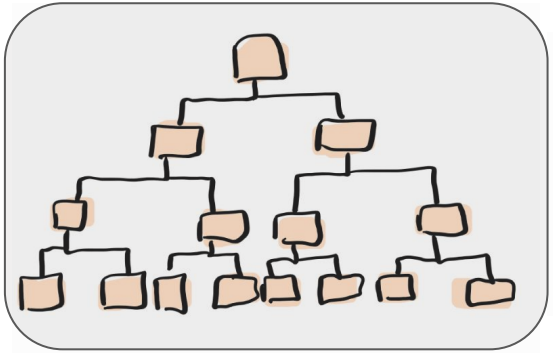
Tensor networks with other structures



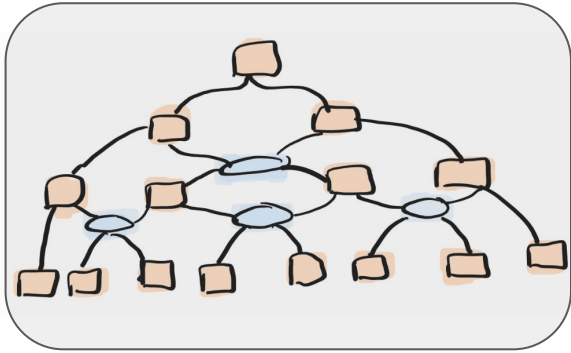
MPS



PEPS



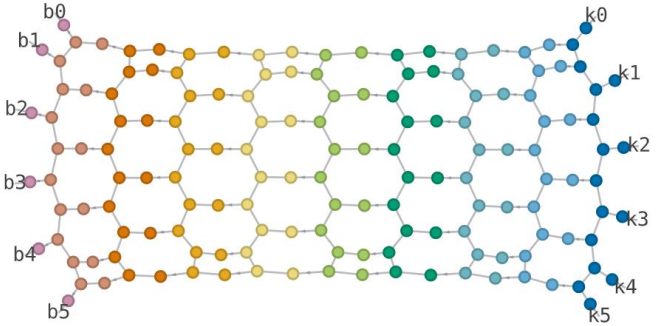
TTN



MERA

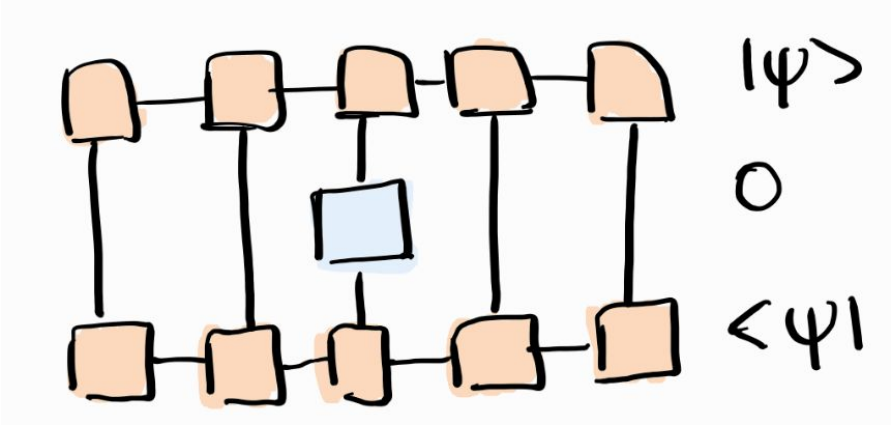
Other structures allow us to capture more entanglement!

But the problem is the **contraction** of the TN.



Another reason for MPS: canonical form

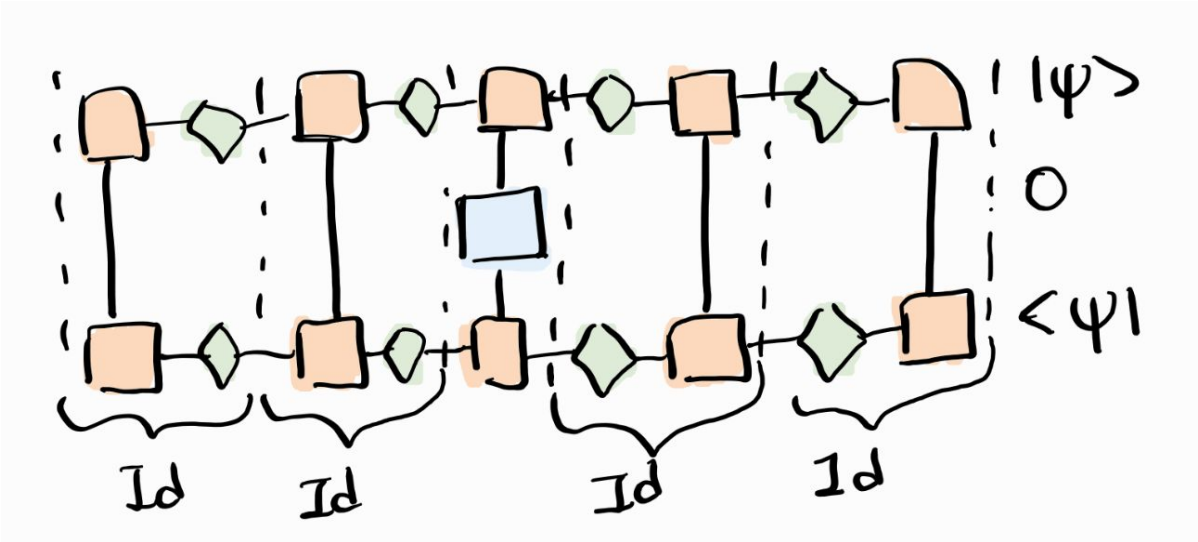
Another reason why MPS is much simpler than the other options is the **canonical form**.



This is used in DMRG, for example.

Another reason for MPS: canonical form

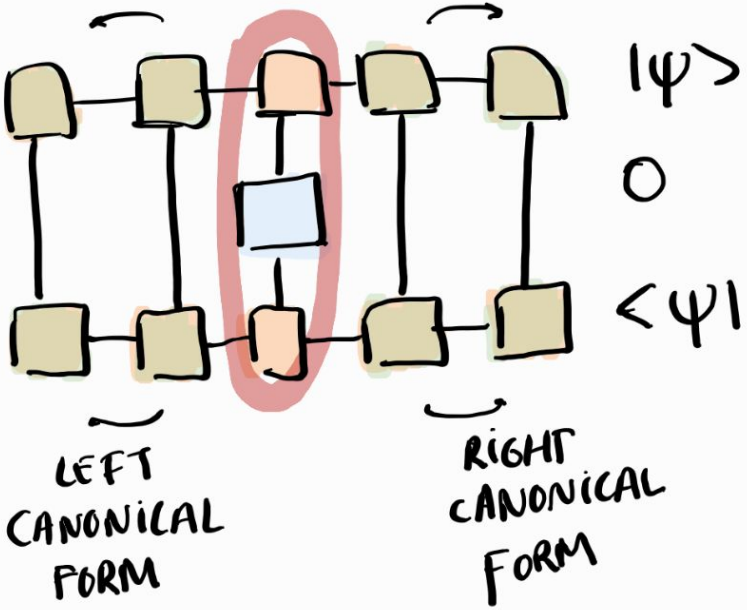
Another reason why MPS is much simpler than the other options is the **canonical form**.



This is used in DMRG, for example.

Another reason for MPS: canonical form

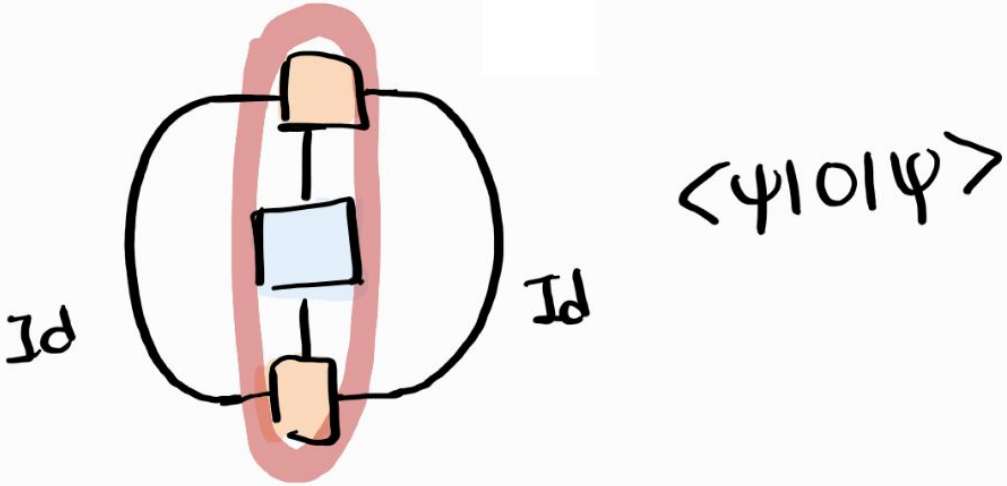
Another reason why MPS is much simpler than the other options is the **canonical form**.



This is used in DMRG, for example.

Another reason for MPS: canonical form

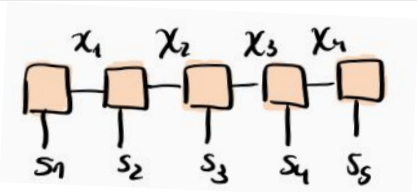
Another reason why MPS is much simpler than the other options is the **canonical form**.



This is used in DMRG, for example.

Summary & conclusions

How to exactly build an MPS starting with a quantum state.
How to compress the state by cutting the bond dimension.
Connection to entanglement.



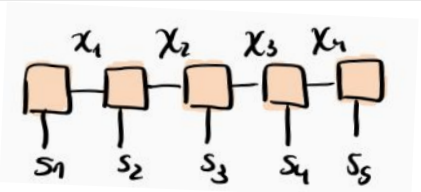
How to apply gates to an MPS using QUIMB.
Build two simple low-entangled states to compare
MPS to statevector simulation.

Build states with more entanglement and study the performance
of MPS regarding memory, computation time and fidelity.

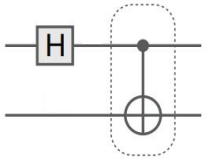
There are other structures that can capture more
entanglement but contraction becomes a problem.

Summary & conclusions

How to exactly build an MPS starting with a quantum state.
How to compress the state by cutting the bond dimension.
Connection to entanglement.



How to apply gates to an MPS using QUIMB.
Build two simple low-entangled states to compare MPS to statevector simulation.

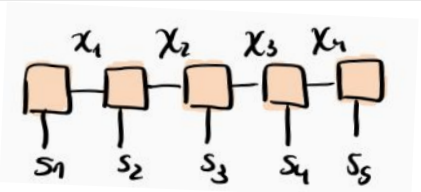


Build states with more entanglement and study the performance of MPS regarding memory, computation time and fidelity.

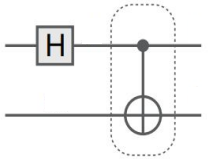
There are other structures that can capture more entanglement but contraction becomes a problem.

Summary & conclusions

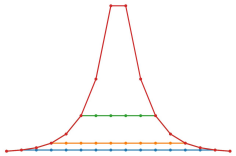
How to exactly build an MPS starting with a quantum state.
How to compress the state by cutting the bond dimension.
Connection to entanglement.



How to apply gates to an MPS using QUIMB.
Build two simple low-entangled states to compare MPS to statevector simulation.



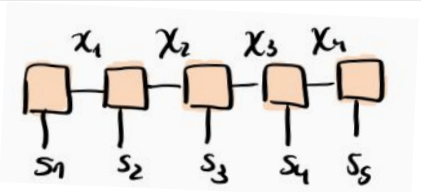
Build states with more entanglement and study the performance of MPS regarding memory, computation time and fidelity.



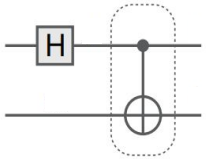
There are other structures that can capture more entanglement but contraction becomes a problem.

Summary & conclusions

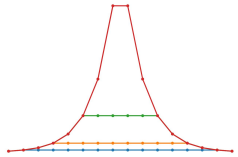
How to exactly build an MPS starting with a quantum state.
How to compress the state by cutting the bond dimension.
Connection to entanglement.



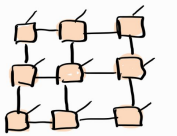
How to apply gates to an MPS using QUIMB.
Build two simple low-entangled states to compare MPS to statevector simulation.



Build states with more entanglement and study the performance of MPS regarding memory, computation time and fidelity.



There are other structures that can capture more entanglement but contraction becomes a problem.



Some references

- Quantum Tensor Networks in a Nutshell
  Jacob Biamonte and Ville Bergholm
  arXiv:1708.00006v1 [quant-ph]
- A Practical Introduction to Tensor Networks: Matrix Product States and Projected Entangled Pair States
  Roman Orus
  Journal: Annals of Physics 349 (2014)117-158
- The density-matrix renormalization group in the age of matrix product states
  Ulrich Schollwoeck
  Journal: Annals of Physics 326, 96 (2011)
- Efficient Classical Simulation of Slightly Entangled Quantum Computations
  Guifré Vidal
  Phys. Rev. Lett. 91, 147902 (2003)
- https://tensornetwork.org/
- https://www.tensors.net/

Some libraries

TeNPy

QUIMB

TensorLy

TensorCircuit

Tensor Network

Pasta

Qiskit

ITensor

Tenet.jl

Questions?