TENSOR NETWORKS *and You*

Nikko Pomata

Stony Brook University

Grad Talks: April 25 2018

OUTLINE

HOW DO YOU NETWORK TENSORS?

- Tensors: a review
- The tensor-network notation
- Tensor network examples

TENSOR NETWORK METHODS

- Matrix product states (MPS)
- Projected Entangled Pair States
- Coarse-graining tensors
- Entanglement Renormalization

TENSOR NETWORKS AND YOU: FRONTIERS IN TENSOR NETWORK RESEARCH

REMINDER: WHAT ARE TENSORS, ANYWAY?

Basic idea

A tensor is a linear combination of tensor products of vectors:

$$\boldsymbol{T} = \sum_j \boldsymbol{v}_1^{(j)} \otimes \boldsymbol{v}_2^{(j)} \otimes \cdots \otimes \boldsymbol{v}_n^{(j)}$$

$$= \sum_{i_1,i_2,\ldots,i_n} T_{i_1,i_2,\ldots,i_n} \boldsymbol{e}_{i_1} \otimes \boldsymbol{e}_{i_2} \otimes \cdots \otimes \boldsymbol{e}_{i_n}$$

in terms of basis vectors

More formally:

A tensor is a multilinear map on vector spaces

$$T^{i_1,\ldots,i_m}{}_{j_1,\ldots,j_n} = \left\langle e^{i_1} \right| \cdots \left\langle e^{i_m} \right| \boldsymbol{T} \left| e^{j_1} \right\rangle \cdots \left| e^{j_n} \right\rangle$$

REMINDER: WHAT ARE TENSORS, ANYWAY?

A tensor is *not*

- A quantity that transforms covariantly with coordinate changes (a tensor *field*)
- An observable that has multiple indices which transform under $SO(3)$ rotations (a tensor *operator*)

TENSOR PRODUCTS IN QUANTUM PHYSICS

If a quantum system is comprised of two subsystems A *and* B*, represented by Hilbert spaces* $\mathcal{H}_A$ *and* $\mathcal{H}_B$*, then the overall Hilbert space is* $\mathcal{H}_A \otimes \mathcal{H}_B$

These subsystems can be:

- The states of two different particles
- The position of a particle along the x axis, and the position of the same particle along the y axis
- A particle's position, and the same particle's spin
- The number of bosons in each of two different states

A MANY-BODY SPIN SYSTEM

Often wind up studying *lattice* systems where every lattice site is a *finite-level* system (e.g. the spin of an atom at that site)

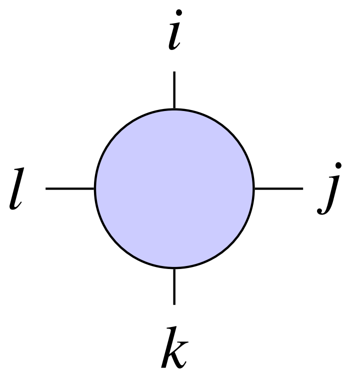
If there are N sites represented by $\mathbb{C}^d$: then

$$\mathcal{H} = \left(\mathbb{C}^d\right)^{\otimes N}$$

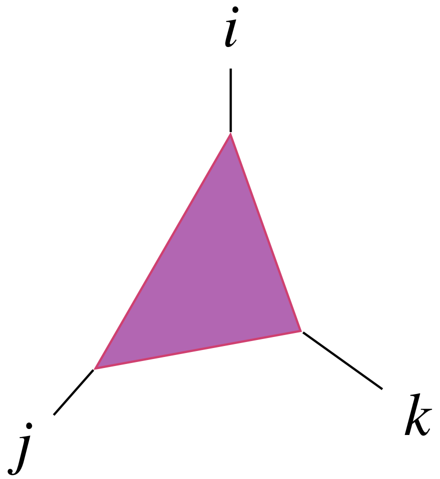
DRAWING TENSORS

In tensor-network notation, a tensor is drawn as a shape and its indices are drawn as lines:

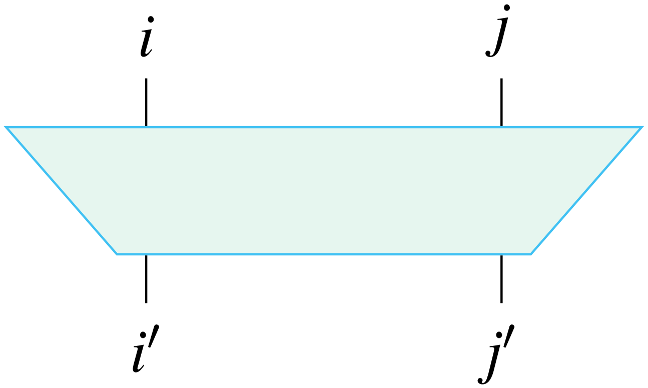
$\boldsymbol{A}_{ijkl} =$



$\boldsymbol{v}^{ij}{}_{k} =$

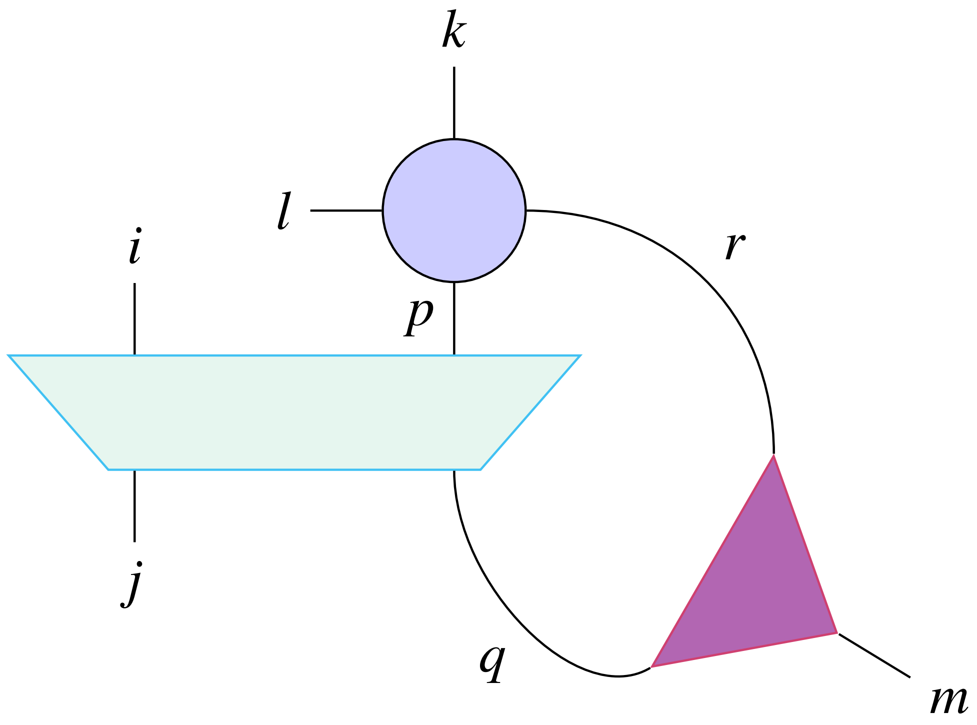


$\boldsymbol{u}^{ij}{}_{i'j'} =$



PUTTING TOGETHER TENSOR NETWORKS

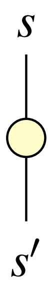
A tensor network is linked together by *contractions* (just like in GR)



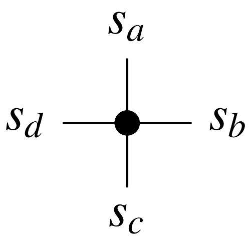
$$= \sum_{pqr} u^{ip}{}_{jq} A_{krpl} v^{rq}{}_{m}$$

EXAMPLE: THE ISING PARTITION FUNCTION

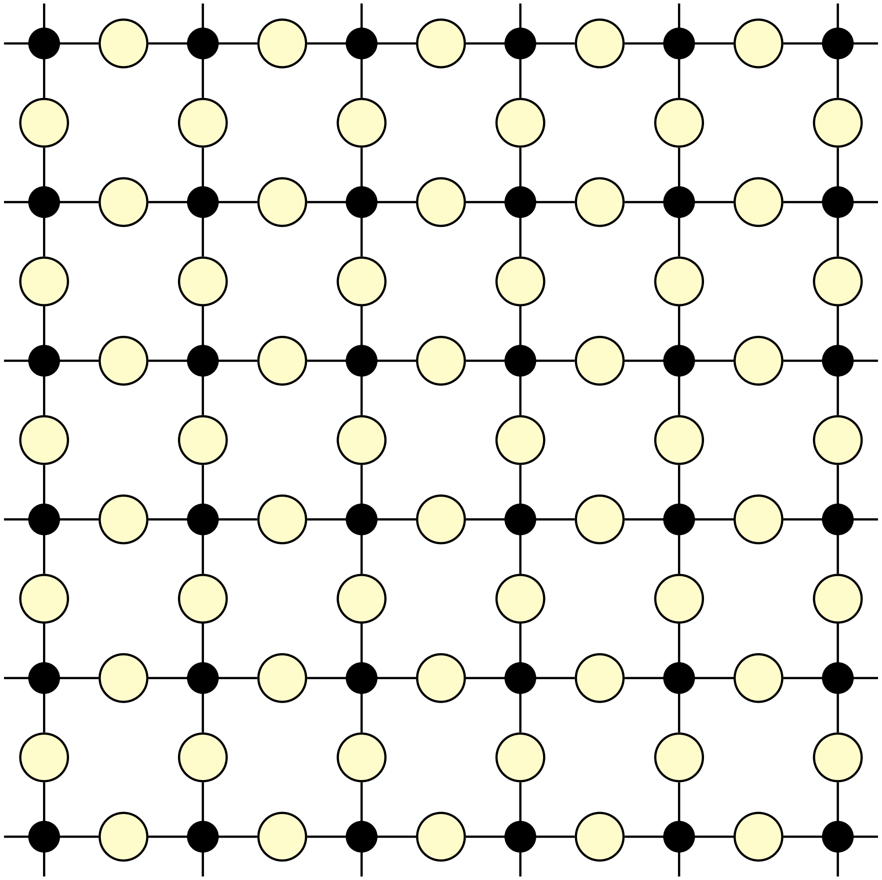
$$Z(\beta) = \sum_{\{s_i\}} \exp\left(-\beta \sum_{\langle i,j \rangle} s_i s_j\right) = \sum_{\{s_i\}} \prod_{\langle i,j \rangle} e^{-\beta s_i s_j}$$



$= e^{-\beta s s'}$



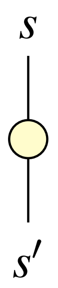
$= \delta_{s_a,s_b,s_c,s_d}$



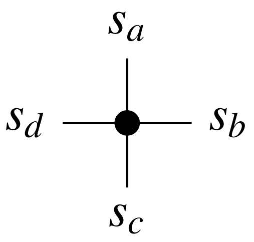
EXAMPLE: THE ISING PARTITION FUNCTION

$$Z(\beta) = \sum_{\{s_i\}} \exp\left(-\beta \sum_{\langle i,j\rangle} s_i s_j\right) = \sum_{\{s_i\}} \prod_{\langle i,j\rangle} e^{-\beta s_i s_j}$$

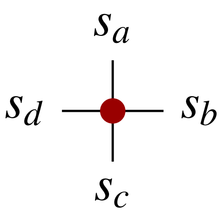
To obtain $\langle s_m s_n \rangle$:



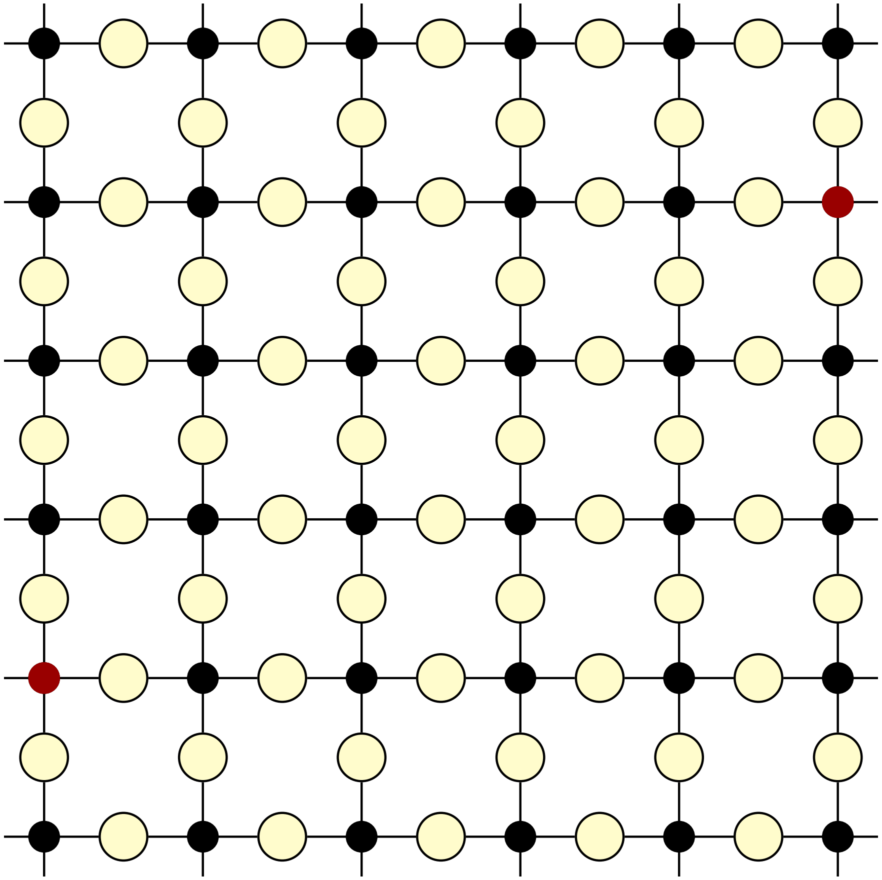
$$= e^{-\beta s s'}$$



$$= \delta_{s_a,s_b,s_c,s_d}$$

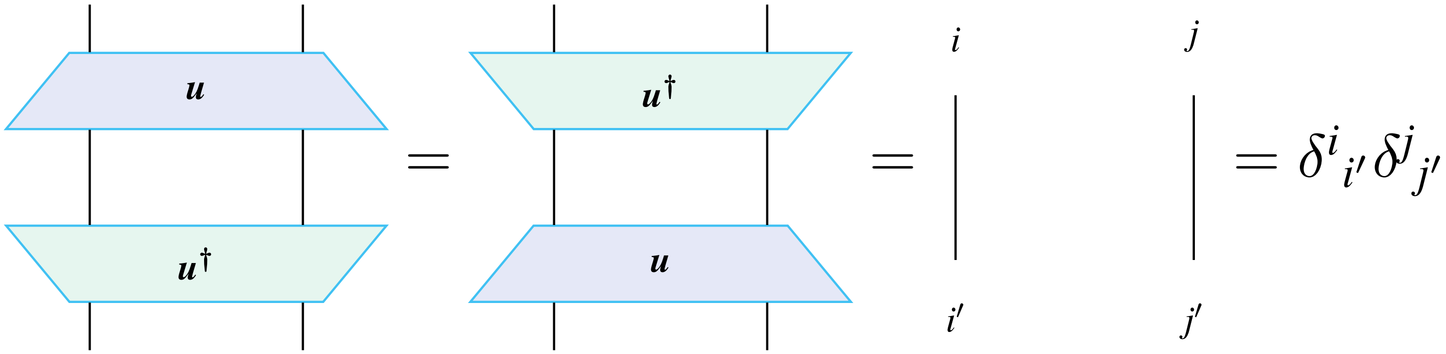


$$= s_a \delta_{s_a,s_b,s_c,s_d}$$

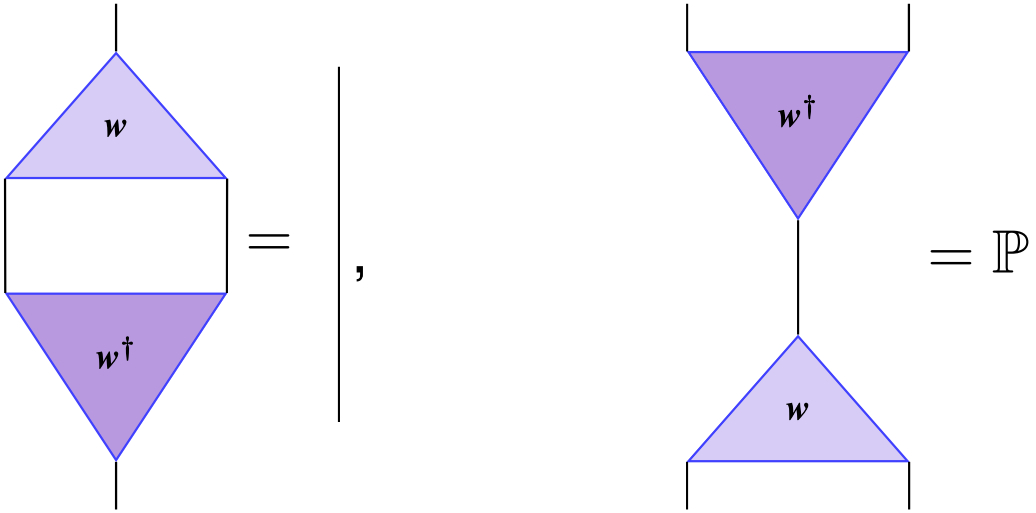


MANIPULATING TENSORS: UNITARIES AND ISOMETRIES

Often wind up dealing with unitary and isometric tensors:



$$= \delta^{i}{}_{i'}\,\delta^{j}{}_{j'}$$



$$= \mathbb{P}$$

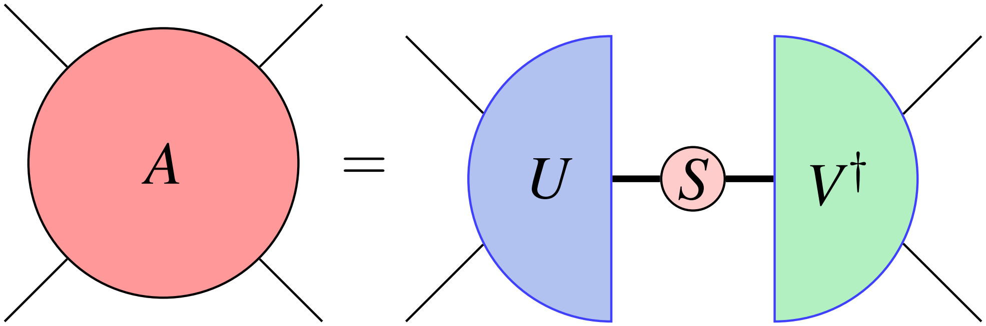
MANIPULATING TENSORS: SVD

The singular value decomposition:

$$A = USV^\dagger$$

- A is an arbitrary matrix
- S is diagonal
- U and V are unitary

In order to turn a matrix equation into a tensor equation, group indices:



HOW DO YOU NETWORK TENSORS?

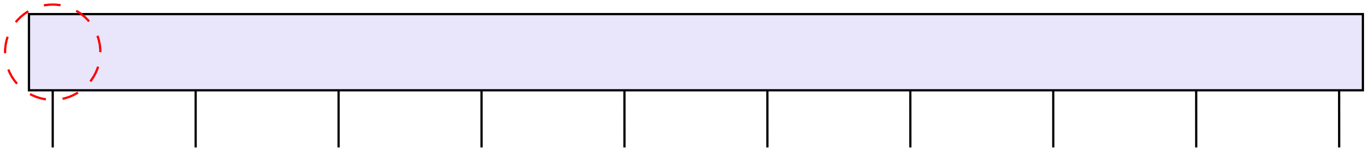
- Tensors: a review
- The tensor-network notation
- Tensor network examples

TENSOR NETWORK METHODS

- Matrix product states (MPS)
- Projected Entangled Pair States
- Coarse-graining tensors
- Entanglement Renormalization

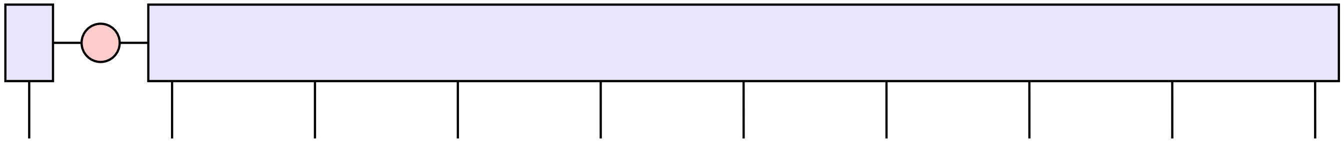
TENSOR NETWORKS AND YOU: FRONTIERS IN TENSOR NETWORK RESEARCH

OBTAINING THE MATRIX PRODUCT STATE



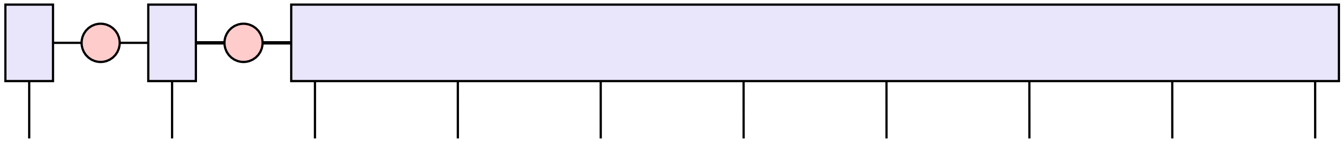
- Start with any state on a spin chain ($\mathbb{C}^{d\ \otimes N}$)
- Apply SVD to the state to divide it into two parts

OBTAINING THE MATRIX PRODUCT STATE



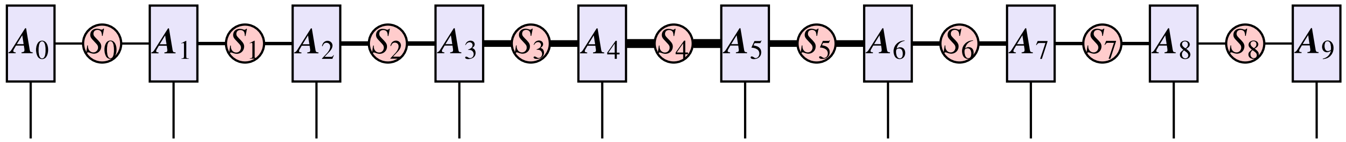
- Start with any state on a spin chain ($\mathbb{C}^{d\ \otimes N}$)
- Apply SVD to the state to divide it into two parts

OBTAINING THE MATRIX PRODUCT STATE



- Start with any state on a spin chain ($\mathbb{C}^{d\,\otimes N}$)
- Apply SVD to the state to divide it into two parts
- ... then keep applying SVD to remaining parts to divide the state site by site

Obtaining the Matrix Product State



- Start with any state on a spin chain ($\mathbb{C}^{d\ \otimes N}$)
- Apply SVD to the state to divide it into two parts
- ... then keep applying SVD to remaining parts to divide the state site by site
- This expresses amplitudes as *matrix products*:

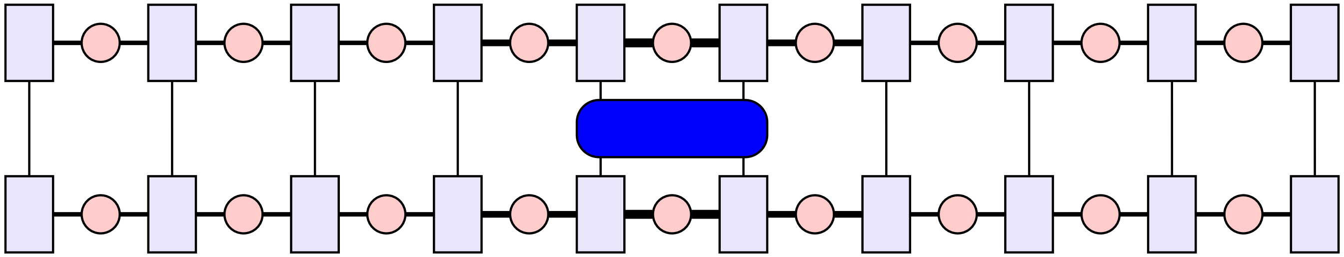
$$\langle i_0 i_1 \cdots i_{N-1} | \psi \rangle = A_0^{(i_0)} S_0 A_1^{(i_1)} S_1 \cdots S_{N-2} A_{N-1}^{(i_{N-1})}$$

- However, the bond dimension in the middle is $\sim 2^{N/2}$

Definition

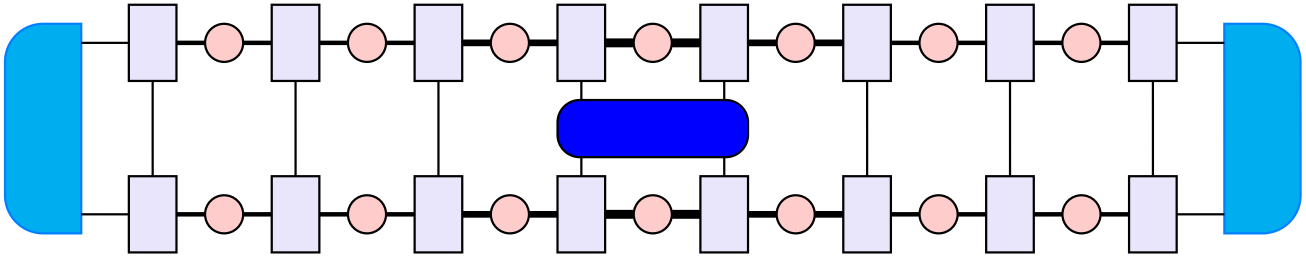
The dimension of an index that is contracted over to obtain a tensor network state is called a *bond dimension* (χ), as opposed to the *physical dimension* d.

USING MATRIX PRODUCT STATES



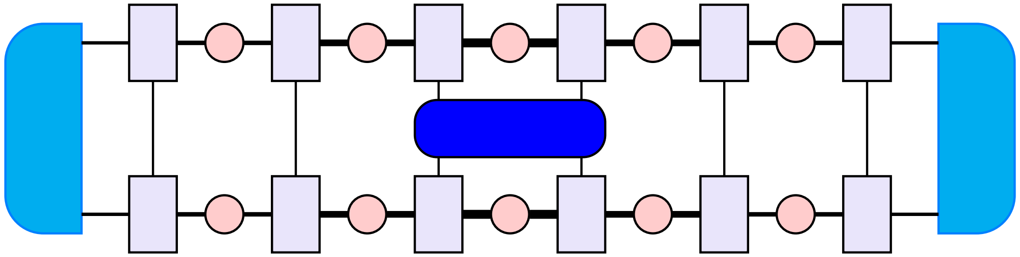
- Calculating observables: apply “transfer matrices” to “boundary conditions”

USING MATRIX PRODUCT STATES



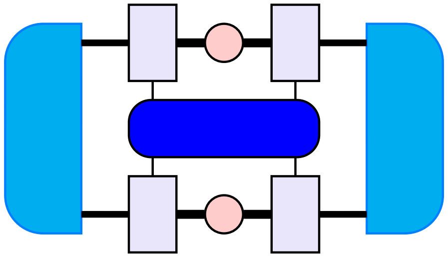
- Calculating observables: apply “transfer matrices” to “boundary conditions”

USING MATRIX PRODUCT STATES



- Calculating observables: apply “transfer matrices” to “boundary conditions”

USING MATRIX PRODUCT STATES



- Calculating observables: apply “transfer matrices” to “boundary conditions”
- Exponential growth of bond dimension χ is a big problem: makes this *harder* than just using the original state
- Solution: *Truncate* the bond dimension - fix it at an artificially small value (often $\sim 10 - 100$). For the *ground state of a “typical” local Hamiltonian*, singular values will fall off quickly enough that this is a good approximation.
- This is the fundamental approximation of tensor networks.

FINDING MATRIX PRODUCT STATES

Usually we're trying to *find* or approximate the ground state of a Hamiltonian. How do we find the best MPS?

Variational

Find a ground state by minimizing $\langle H \rangle$. From the equation

$$\langle \psi | H | \psi \rangle = E \langle \psi | \psi \rangle ,$$

remove all instances of the tensor A_i and solve the resulting generalized eigenvalue problem.
This is the *Density Matrix Renormalization Group* method, or **DMRG**, originally formulated by Steven White without matrix product states.

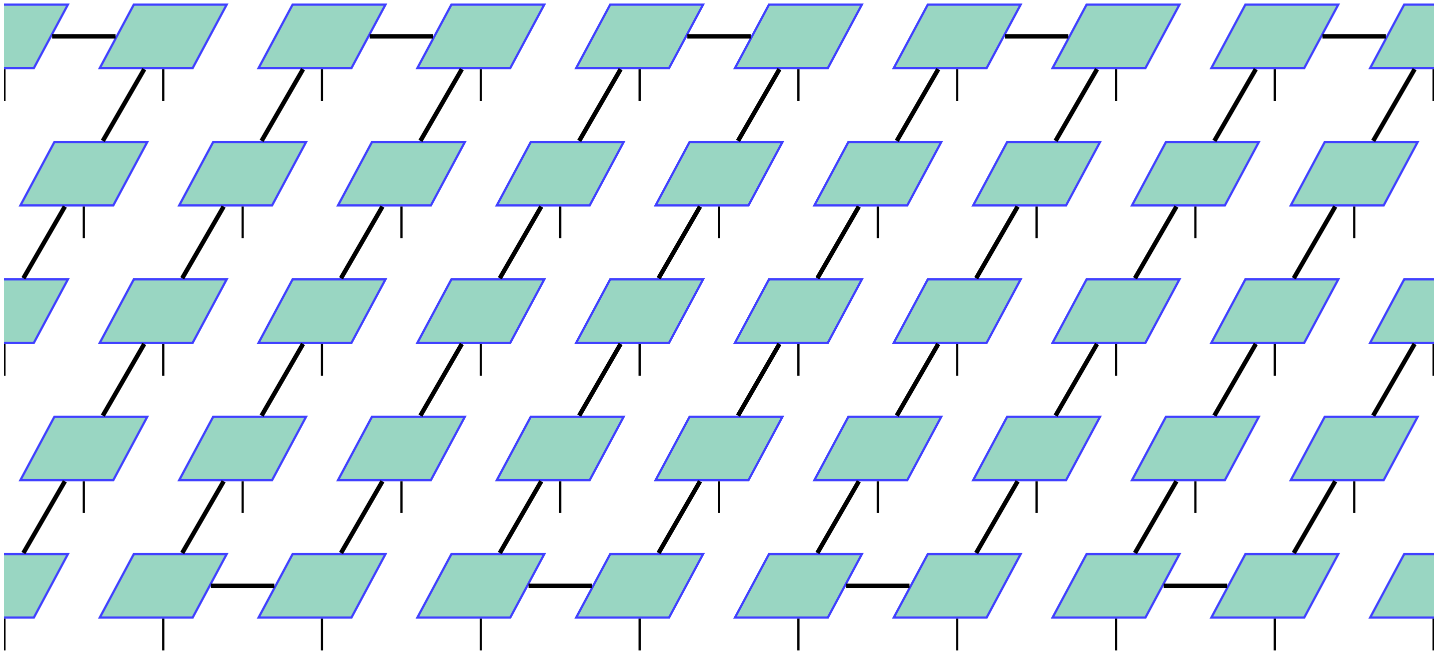
Projection

Take an arbitrary state $|\varphi\rangle$, and estimate $e^{-\beta H} |\varphi\rangle$, in the limit $\beta \to \infty$. Use the Suzuki-Trotter expansion to approximate $e^{-\beta H}$ as a product of "small" two-site operators. Use a truncated SVD to return the state to MPS form after successive applications. This is the *Time-Evolving Block Decimation* method, or **TEBD**.

- These are the two general approaches we apply to find tensor-network ground states
- These methods are exact, up to machine precision, as $\chi \to \infty$.
- Can apply to infinite, homogeneous spin chains: now find *dominant eigenvector* of transfer matrices

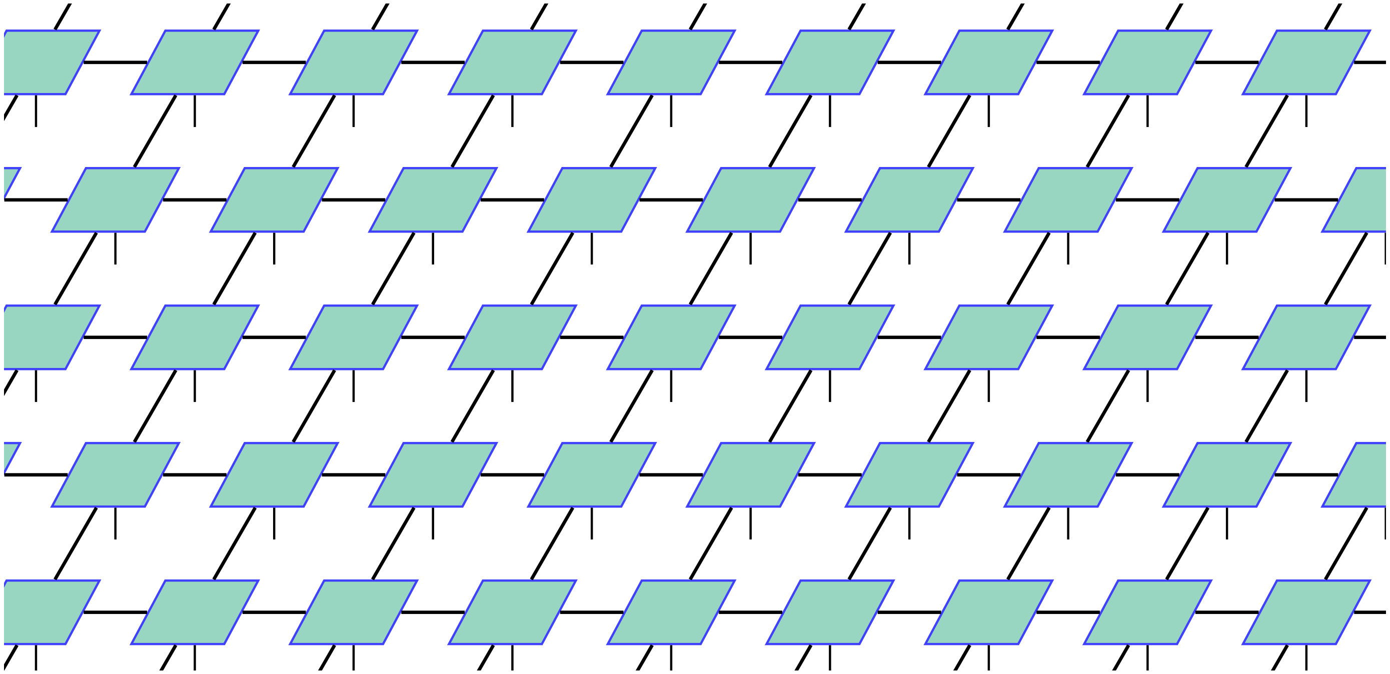
MATRIX PRODUCT STATES IN HIGHER DIMENSIONS?

First attempt: the snake method



This is reliable, but extremely inefficient for wide systems

THE PROJECTED ENTANGLED PAIR STATE (PEPS)



More elegant, more homogenous, and truly 2D

But it can't be contracted exactly or efficiently

$\Downarrow$

It's hard to optimize efficiently