Lecture 4: Matrix product states

Last time, we saw that w/ exact diag., an N-site system needs us to make and diagonalize a $2^N \times 2^N$ Hamiltonian.

$\mathcal{O}(N \cdot 2^N)$ $\qquad\qquad$ $\mathcal{O}(8^N)$

Symmetries can help: For Ising, $2^N \to 2^N/2$

For XXZ, $2^N \to 2^N/\sqrt{N}$

But not that much, it's still exponential.

~~The fundamental problem is that the ground~~

There are several fundamental reasons for this bad scaling:

1. We find all 2^N eigenstates even if we only care about the ground state
2. The ground state in the basis we are using will have $\sim 2^N$ nonzero coefficients
3. (related to 2): in writing H and $|\psi\rangle$ in the vector/matrix representation, we lost all the information about ~~space~~, locality
   ie that H is made of operators acting only on neighboring sites, or that there are no long-range interactions.

$\boxed{1}$ allows an improvement even for exact diag, using so-called "sparse matrices" and "iterative eigensolvers". We'll discuss this in the next class (Lecture 5).

$\boxed{2 \& 3}$ These motivate the introduction of the $\boxed{\text{Matrix Product State}}$ (MPS)

~~To more clearly understand the inefficiency in the method we've been using, consider~~

~~Before we talk about the matrix product state, we will talk about plain old product state.~~

Consider

Before saying more about MPSs, I want to elaborate on ② & ③ and with an example.

Let's consider $N=2$ spin $\frac{1}{2}$ s, and consider the state

$$\left(a\,|\uparrow\rangle + b\,|\downarrow\rangle\right)_A \otimes \left(c\,|\uparrow\rangle + d\,|\downarrow\rangle\right)_B$$

If we write this in the combined basis, it looks like

$$ac\,|\uparrow\uparrow\rangle + bc\,|\downarrow\uparrow\rangle + ad\,|\uparrow\downarrow\rangle + bd\,|\downarrow\downarrow\rangle$$

Ok, now we'll act on this state with the operator

$$\sigma_1^x \otimes \sigma_2^z$$

There are 2 ways to do this:

- $$\left[\sigma_1^x\left(a\,|\uparrow\rangle + b\,|\downarrow\rangle\right)\right] \otimes \left[\sigma_2^z\left(c\,|\uparrow\rangle + d\,|\downarrow\rangle\right)\right]$$
  $$= \left(a\,|\downarrow\rangle + b\,|\uparrow\rangle\right) \otimes \left(c\,|\uparrow\rangle - d\,|\downarrow\rangle\right)$$

How many calculations did this take? For each spin, we had to do 2 calculations, 1 for $|\uparrow\rangle$, 1 for $|\downarrow\rangle$

so $2\cdot N = 2\cdot 2 = 4$

• $(\sigma_1^x \otimes \sigma_2^z)\left(ac\,|\uparrow\uparrow\rangle + bc\,|\downarrow\uparrow\rangle + ad\,|\uparrow\downarrow\rangle + bd\,|\downarrow\downarrow\rangle\right)$

$= ac\,|\downarrow\uparrow\rangle + bc\,|\uparrow\uparrow\rangle - ad\,|\downarrow\downarrow\rangle - bd\,|\uparrow\downarrow\rangle$

This required one action of $\sigma_1^x \otimes \sigma_2^z$ for each of the 2^N basis states, so $2^2 = 4$

In this case, both required 4 calculations, but now suppose $N > 2$.

If $|\varphi\rangle = (a_1|\uparrow\rangle + b_1|\downarrow\rangle) \otimes (a_2|\uparrow\rangle + b_2|\downarrow\rangle) \otimes \ldots \otimes (a_N|\uparrow\rangle + b_N|\downarrow\rangle)$

applying $O_1 \otimes O_2 \otimes \ldots \otimes O_N$

we can just apply each O_i to the ith spin, and this takes $2N$ actions.

If we write out $|\varphi\rangle$ w/ 2^N coeffs, the operator must be applied 2^N times. That's way worse, and you get the same state at the end!

This second version is what we were doing in our exact diagonalization, by expanding the tensor product to get $H: 2^N \times 2^N$ / The first version uses "locality" by acting separated on each spin

There's a problem though! What if $|\varphi\rangle$ can't be written in this $|\varphi_1\rangle \otimes |\varphi_2\rangle \otimes |\varphi_3\rangle \otimes \ldots$ form?

This is called a "product state".

insert Exercise here?

<u>Consider eg</u> $|\varphi\rangle = \dfrac{|\uparrow\uparrow\rangle + |\downarrow\downarrow\rangle}{\sqrt{2}}$

Try to solve $\quad (a|\uparrow\rangle + b|\downarrow\rangle) \otimes (c|\uparrow\rangle + d|\downarrow\rangle) = |\varphi\rangle$

$$\Rightarrow ac = \frac{1}{\sqrt{2}} = bd \, , \; ad = bc = 0$$

But $ad = 0 \Rightarrow a = 0$ or $d = 0$

$\Rightarrow ac = 0$ or $bd = 0$ $\lightning$

<u>It's not possible!</u>

This is called an "<u>entangled state</u>"

Another argument for not all states are product states.

Consider N spin-$\frac{1}{2}$:

- a generic state is a linear combination of 2^N basis states

  $\Rightarrow$ need 2^N complex #s to specify the state

  (-1 real # for normalization)

  $\Rightarrow 2 \cdot 2^N - 1$ real numbers

  ie the dimension of this space is $2 \cdot 2^N - 1$

  eg $N = 2$: $8 - 1 = 7$

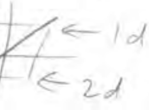
- a product state is described by

  $(a_1, b_1), (a_2, b_2), \ldots (a_N, b_N) \rightarrow 2 \cdot N$ complex #s

  But normalization $\Rightarrow 2 \cdot (2N - 1)$ real #s

  eg $N = 2$: $2(3) = 6$

So [product states] in [all states] is like a <u>line</u> in a <u>plane</u>



Almost all points are not on the line; almost all states are entangled.

This would seem to kill the dream of something more efficient than exact diagonalization!

for special states we can do better, but the GS in general will <u>not</u> be a product state.

The clever solution to this is the "matrix product state"

Instead of writing

$$|\psi\rangle = (a_1|\uparrow\rangle + b_1|\downarrow\rangle) \otimes (a_2|\uparrow\rangle + b_2|\downarrow\rangle) \otimes \ldots \otimes (a_N|\uparrow\rangle + b_N|\downarrow\rangle)$$

we convert a_1, b_1, etc to <u>matrices</u>

$$|\psi\rangle = (\underbrace{A_1}_{1\times D_1}|\uparrow\rangle + B_1|\downarrow\rangle) \otimes (\underbrace{A_2}_{D_1\times D_2}|\uparrow\rangle + B_2|\downarrow\rangle) \otimes \ldots \otimes (\underbrace{A_N}_{D_{N-1}\times 1}|\uparrow\rangle + B_N|\downarrow\rangle)$$

note: technically wrong

Then coefficient of $|\uparrow\rangle\otimes|\uparrow\rangle\otimes\ldots\otimes|\uparrow\rangle$ is the matrix product $A_1 \cdot A_2 \cdot \ldots \cdot A_N$

and so forth.

- Note that this includes the <u>product state</u> as a special case: if $D_1 = D_2 = \ldots = D_{N-1} = 1$, then all the matrices are 1×1, ie numbers
- This actually includes <u>all</u> states on N spins, but in that case some matrices must be $2^{N/2}\times 2^{N/2}$, so we don't gain much.
- Fortunately, it turns out that for real systems, you can get a very good approximation of the GS w/ small D.

Let's see an example.

① Previously we said that $|\phi^+\rangle = \frac{|\uparrow\uparrow\rangle + |\downarrow\downarrow\rangle}{\sqrt{2}}$ is not a product state.

But it is a matrix product state

$$\left(A_1 = \left(\tfrac{1}{\sqrt{2}}\ 0\right),\ B_1 = \left(0\ \tfrac{1}{\sqrt{2}}\right)\right),\ \left(A_2 = \begin{pmatrix}1\\0\end{pmatrix},\ B_2 = \begin{pmatrix}1\\0\end{pmatrix}\right)$$

term	coeff
$\|\uparrow\uparrow\rangle$	$A_1 \cdot A_2 = \frac{1}{\sqrt{2}}$
$\|\uparrow\downarrow\rangle$	$A_1 \cdot B_2 = 0$
$\|\downarrow\uparrow\rangle$	$B_1 \cdot A_2 = 0$
$\|\downarrow\downarrow\rangle$	$B_1 \cdot B_2 = \frac{1}{\sqrt{2}}$

So $|\Phi^+\rangle$ is a $D=2$ MPS

it is not a $D=1$ MPS (product state)

This minimum D that lets you represent $|\psi\rangle$ is called the "Schmidt rank"

② 3 qubits $\quad \frac{1}{\sqrt{2}}(|\uparrow\uparrow\uparrow\rangle + |\downarrow\downarrow\downarrow\rangle)$

$$\Big[\underbrace{\left(\tfrac{1}{\sqrt{2}}, 0\right)}_{A_1},\ \underbrace{\left(0, \tfrac{1}{\sqrt{2}}\right)}_{B_1}\Big],\ \Big[\underbrace{\begin{pmatrix}1&0\\0&0\end{pmatrix}}_{A_2},\ \underbrace{\begin{pmatrix}0&0\\0&1\end{pmatrix}}_{B_2}\Big],\ \Big[\underbrace{\begin{pmatrix}1\\0\end{pmatrix}}_{A_3},\ \underbrace{\begin{pmatrix}0\\1\end{pmatrix}}_{B_3}\Big]$$

☆ explain how each D corresponds to one term.

Note that there is some freedom here! unlike for product states, there's no way to define normalization for each site separately, so eg we can put the $\frac{1}{\sqrt{2}}$ on any of the sites. There's even more freedom, which

In this way we can easily represent a state w/ n terms using $D = n$:

~~and choose their~~

For each term, if it's $|\uparrow\rangle$ put a 1 in that spot on diagonal for $|\uparrow\rangle$ matrix, if it's $|\downarrow\rangle$ put a 1 in the diag for $|\downarrow\rangle$ matrix.

eg $\dfrac{|\uparrow\uparrow\downarrow\uparrow\rangle + |\downarrow\uparrow\uparrow\downarrow\rangle}{\sqrt{2}}$ $\quad n = 2 \Rightarrow D = 2$

$$\left[\begin{pmatrix} 1 & \\ & \end{pmatrix} \begin{pmatrix} & \\ & 1 \end{pmatrix} \Big\} \begin{pmatrix} 1 & \\ & 1 \end{pmatrix} \begin{pmatrix} & \\ & \end{pmatrix} \Big\} \begin{pmatrix} & \\ & 1 \end{pmatrix} \begin{pmatrix} 1 & \\ & \end{pmatrix} \Big\} \begin{pmatrix} 1 & \\ & \end{pmatrix} \begin{pmatrix} & \\ & 1 \end{pmatrix} \right] \times \left(\frac{1}{\sqrt{2}}\right)$$

When you multiply these out, each elt along the diagonal multiplies independently.

$\boxed{\text{Exercises}}$ ① Find an MPS for $\dfrac{|\uparrow\downarrow\rangle - |\downarrow\uparrow\rangle}{\sqrt{2}}$

② Find a $D = 3$ MPS for

$$\frac{|\uparrow\uparrow\downarrow\rangle + |\uparrow\downarrow\uparrow\rangle + |\downarrow\uparrow\uparrow\rangle}{\sqrt{3}} \equiv |W\rangle$$

③ Use a parameter-counting argument to show that any 3-spin state can be a $D = 2$ MPS. What is the maximum needed D for 4 spins?

④ Consider the $D = 2$ MPS:

$$\Big[\underset{A_1}{(1\ 0)}\ \underset{B_1}{(0\ 1)}\Big],\ \Big[\underset{A_2}{\begin{pmatrix} a & b \\ c & d \end{pmatrix}},\ \underset{B_2}{\begin{pmatrix} e & f \\ g & h \end{pmatrix}}\Big],\ \Big[\underset{A_3}{\begin{pmatrix} 0 \\ 1 \end{pmatrix}}\underset{B_3}{\begin{pmatrix} 1 \\ 0 \end{pmatrix}}\Big]$$

Write this out in the coeff. basis state form.

Then write $\dfrac{|\uparrow\uparrow\downarrow\rangle + |\uparrow\downarrow\uparrow\rangle + |\downarrow\uparrow\uparrow\rangle}{\sqrt{3}}$ as a $D = 2$ MPS.

Answers ① $\left[\left(\frac{1}{\sqrt{2}}, 0\right), \left(0, -\frac{1}{\sqrt{2}}\right)\right], \left[\begin{pmatrix}0\\1\end{pmatrix}, \begin{pmatrix}1\\0\end{pmatrix}\right]$

② $\left[\left(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, 0\right)\ \left(0, 0, \frac{1}{\sqrt{3}}\right)\right], \left[\begin{pmatrix}1 & 0 \\ & 1\end{pmatrix}, \begin{pmatrix}0 \\ 1 \\ 0\end{pmatrix}\right],$

$\left[\begin{pmatrix}0\\1\end{pmatrix}, \begin{pmatrix}1\\0\\0\end{pmatrix}\right]$

③ $\underline{N=3}$ all states $\rightarrow 2 \cdot 2^N - 1$ real parameters

$= 15$

$D=2$ MPS $\rightarrow 2 \cdot (2 \cdot D + 2 \cdot D^2 \cdot (N-2) + 2 \cdot D) - 1$

$= 4(2D + D^2(N-2)) - 1$

$\cancel{= 2(4 + 8 + 4) - 1 = 31}$ $= 4(4+4) - 1 = 31$

$\underline{N=4}$ all states $\rightarrow 31$

$D=2$ MPS $\rightarrow 4(4 + 4 \cdot 2) = 47$

Seems like $D \leq 2$ is enough

④ $a|\uparrow\uparrow\uparrow\rangle + b|\uparrow\uparrow\downarrow\rangle + c|\downarrow\uparrow\uparrow\rangle + d|\downarrow\uparrow\downarrow\rangle$
$+ e|\uparrow\downarrow\uparrow\rangle + f|\uparrow\downarrow\downarrow\rangle + g|\downarrow\downarrow\uparrow\rangle + h|\downarrow\downarrow\downarrow\rangle$

Note: it will not work out so cleanly for larger N!

So MPS is

$\left[\left(\frac{1}{\sqrt{3}}\ 0\right), \left(0\ \frac{1}{\sqrt{3}}\right)\right], \left[\begin{pmatrix}0&1\\1&0\end{pmatrix}, \begin{pmatrix}1&0\\0&0\end{pmatrix}\right], \left[\begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix}\right]$

Unfortunately, the naive counting argument really doesn't work, because the parameters are not really independent. Many different configurations correspond to the same state. This is called "gauge freedom"

Counting argument says D is too small $\Downarrow$ D is too small, but "$\Uparrow$" doesn't work

$\underline{\text{eg}}$ $\frac{|\uparrow\uparrow\rangle + |\downarrow\downarrow\rangle}{\sqrt{2}}$ could be $\left(\frac{1}{\sqrt{2}}\ 0\right)\left(0\ \frac{1}{\sqrt{2}}\right), \begin{pmatrix}1\\0\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix}$

or

$\left(0\ \frac{1}{\sqrt{2}}\right)\left(\frac{1}{\sqrt{2}}\ 0\right), \begin{pmatrix}0\\1\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix}$

(or many others)

For example, for $N=4$, you actually need in general matrices of sizes

$1\times 2,\ 2\times 4,\ 4\times 2,\ 2\times 1$

It's better to ~~think~~ of these dimensions as living "between" the matrices.

$$\begin{array}{ccccccccc} & A_1 & & A_2 & & A_3 & & A_4 & \\ & B_1 & & B_2 & & B_3 & & B_4 & \\ \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow \\ D=1 & & D=2 & & D=4 & & D=2 & & D=1 \end{array}$$

The D you need is the minimum of the dimension of the space to the left and the space to the right

so need $D=4$ b/c 1st 2 spins have total dimension 4, last 2 also have 4.

~~These numbers D are called the "bond dimensions" of the MPS.~~

The reason for this: look at $\begin{Bmatrix} A_2 \\ B_2 \end{Bmatrix}$ $\xleftarrow{2\text{-}d}$ $\xrightarrow{4\text{-}d}$

gives correspondence between left and right parts of the system:
If left is in $|\uparrow\rangle$, then in $|\psi\rangle$, what is the state of right?
This a transformation from a 2-dimensional space to 4d.
In linear algebra, that's a 2×4 matrix.

eg

$$|W\rangle = \frac{|\uparrow\uparrow\downarrow\rangle + |\uparrow\downarrow\uparrow\rangle + |\downarrow\uparrow\uparrow\rangle}{\sqrt{3}}$$

$$\left[(1\ 0)\ (0\ 1)\right],\ \left[\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}\right],\ \left[\begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix}\right]$$

This is a function:

if 2nd spin is $|\uparrow\rangle$, then if the total state is $|\psi\rangle$, how does 3rd spin state correspond to 1st spin state

In this case $|\uparrow\rangle_3 \longmapsto |\downarrow\rangle_1$
$|\downarrow\rangle_3 \longmapsto |\uparrow\rangle_1$

likewise B_2 says $|\uparrow\rangle_3 \longmapsto |\uparrow\rangle_1$
$|\downarrow\rangle_3 \longmapsto 0$ ← ie $\downarrow_2\downarrow_3$ is not allowed

Ok, now we have some idea of what an MPS is.

Let's see how it's useful.

Step 1 act operator on a site

Consider a ~~state~~ like

$$(A_1|\uparrow\rangle + B_1|\downarrow\rangle) \otimes (A_2|\uparrow\rangle + B_2|\downarrow\rangle) \otimes \ldots \otimes (A_N|\uparrow\rangle + B_N|\downarrow\rangle)$$

And let's apply σ_x to spin 2.

All we need to do is swap A_2 & B_2: $B_2|\uparrow\rangle + A_2|\downarrow\rangle$

So if we were on a computer and storing $|\psi\rangle$, we can choose to store only the A_i, B_i and never to compute all the coefficients.

Then applying an operator to site i only requires changing 2 matrices, we don't need to change 2^N coefficients!

(A) But, you should have an objection now.

We just showed that the center matrices are $2^{N/2} \times 2^{N/2}$.

This is still really bad!

Sol: Let's assume that we only care about states where a small D (≤ 100 let's say) is sufficient.

In the last class we'll discuss if this is reasonable or not

There's one more super important step: to stop separating A and B.

We can think of A and B as elts in a vector:

site i state is $\begin{pmatrix} A_i \\ B_i \end{pmatrix}$ ← vector of matrices

Does this make sense? Let's try acting some operators.

σ_i^x acts on $A_i |\uparrow\rangle + B_i |\downarrow\rangle$ like

$$A_i |\downarrow\rangle + B_i |\uparrow\rangle \quad \text{so} \quad \begin{pmatrix} A_i \\ B_i \end{pmatrix} \mapsto \begin{pmatrix} B_i \\ A_i \end{pmatrix}$$

σ_i^z acts like $A_i |\uparrow\rangle + B_i |\downarrow\rangle \longmapsto A_i |\uparrow\rangle - B_i |\downarrow\rangle$

$$\begin{pmatrix} A_i \\ B_i \end{pmatrix} \mapsto \begin{pmatrix} A_i \\ -B_i \end{pmatrix}$$

σ_i^y acts like $A_i |\uparrow\rangle + B_i |\downarrow\rangle \longmapsto A_i \cdot i |\downarrow\rangle + B_i \cdot -i \cdot |\uparrow\rangle$

$$\begin{pmatrix} A_i \\ B_i \end{pmatrix} \mapsto \begin{pmatrix} -iB_i \\ iA_i \end{pmatrix}$$

On the vector $\begin{pmatrix} A_i \\ B_i \end{pmatrix}$, these actions are given by

$$\sigma^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad \sigma^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma^y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

in other words, just the normal matrices!

So this vector notation really does make sense.

So our state on site i looks like

insert argument ~~that~~ all matrix mult. is just matrix · vector

$$\begin{array}{c} \uparrow \\ 2 \\ \downarrow \end{array} \begin{pmatrix} D_{left} \times D_{right} \\ \\ D_{left} \times D_{right} \end{pmatrix}$$

Instead of a vector of matrices, let's try to think of it as a rank-3 tensor: $2 \times D_{left} \times D_{right}$

~~In this form, we can easily write a computer program to act an operator in this state.~~

~~Let $T \equiv 2 \times D_{left} \times D_{right}$, act σ^x on this site using~~

This feels like a meaningless way of rewriting, but it's actually quite useful to change your perspective. To see why, consider 2 different operations we might do.

① apply O to site i

O is a 2×2 matrix $\begin{pmatrix} o_{11} & o_{12} \\ o_{21} & o_{22} \end{pmatrix}$

new state on site i is $\begin{pmatrix} o_{11} A_i + o_{12} B_i \\ o_{21} A_i + o_{22} B_i \end{pmatrix}$

Let's rename $A_i \to A_i^0$, $B_i \to A_i^1$

and let A_i^0 have elements a_{jk}^0 j up to D_{left}

A_i^1 have elements a_{jk}^1 k up to D_{right}

In the $2\times D_L \times D_R$ tensor notation, we get

$$\{\tilde{A}_i\}_{nlm} = \sum_{k=1,2} o_{nk}\, a_{lm}^k$$

Now, we could have done

$\{\tilde{A}_i\}_n = \sum_{k=1,2} o_{nk} A^k$ where again A^0 is the coeff of $|\uparrow\rangle$

B^0 is the coeff of $|\downarrow\rangle$

But the rank-3 tensor form is better for:

② Find state of 2 sites

There are now 4 basis states, $|\uparrow\uparrow\rangle, |\uparrow\downarrow\rangle, |\downarrow\uparrow\rangle, |\downarrow\downarrow\rangle$, w/ coeffs $A_i^0 A_{i+1}^0$, $A_i^0 A_{j+1}^1$, $A_i^1 A_{i+1}^0$, $A_i^1 A_{i+1}^1$

We can put these into a 2×2 matrix of matrices

$$\begin{array}{c|cc} & |\uparrow\rangle_{i+1} & |\downarrow\rangle_{i+1} \\ \hline |\uparrow\rangle_i & A^0A^0 & A^0A^1 \\ |\downarrow\rangle_i & A^1A^0 & A^1A^1 \end{array}$$

$\underline{\underline{\text{or}}}$ a $2\times 2\times D_L \times D_R$, rank-4 tensor.

To calculate the elements of this tensor from the separate site tensors, do

$$l \in \{1, \dots, D_L\}$$

$$\{\tilde{A}\}_{s1, s2, l, m} = \sum_k A^{s1}_{lk} A^{s2}_{km} \quad m \in \{1, \dots, D_R\}$$

spin on i spin on $i+1$

element of rank-3 tensor

$k \in \{1, \dots, D_{center}\}$

In the higher-rank tensor notation, both ① and ② just look like matrix multiplication with some extra indices.

Fortunately for us, that is actually implemented in python

np. linalg. tensordot

Will see the programming side of things next time.

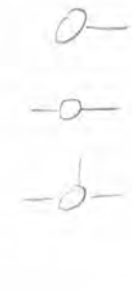
One last topic for today: graphical representation

We can draw a rank-N tensor like this:

rank-1 :

rank-2 :

rank-3 :

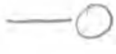


~~This picture should be regarded as a set of instructions.~~

~~You get elements of v by~~

Connecting two lines indicates matrix multiplication.

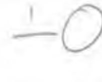
eg if  is the matrix $\begin{pmatrix} o_{11} & o_{12} \\ o_{21} & o_{22} \end{pmatrix}$

 is the vector $\begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$

Then

 is the vector $\begin{pmatrix} o_{11} & o_{12} \\ o_{21} & o_{22} \end{pmatrix}\begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$

To extend this to "tensordot" we can think of it this way:

 $= (o_{ij}) \leftarrow$ [parens mean "the whole matrix that has these elts"]

 $= (v_k)$

 $= \left(\sum_j \sum_k \delta_{jk} \; o_{ij} \, v_k \right)$

We sum over both the indices that have disappeared, and δ_{jk} forces them to be equal!

$= \left(\sum_j o_{ij} \, v_j \right)$

This picture is easily extended to higher rank:

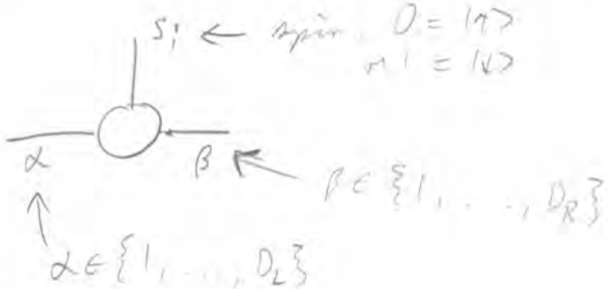
eg

i, j, m element is

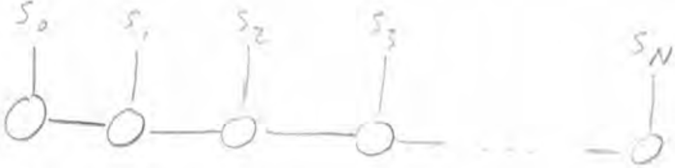
$\sum_k A_{ijk} B_{km}$

$k = l$ because we connected the line.

So in this notation, our state on site i look like



The whole state $|\psi\rangle$ is

$|\psi\rangle =$ 

This is a $2 \times 2 \times 2 \times \ldots \times 2$, rank-$N$ tensor

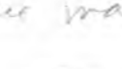
→ In total it has 2^N elements, the coeffs of the 2^N basis states.

→ if we ever calculate these 2^N elts., we lose all benefits of MPS, so don't do that!

But we can be clever, and the graphical notation will help.

Next time, we will see how this works.

$\boxed{\text{Exercises}}$ ① Show for a square matrix A that

$\mathrm{Tr}(A) =$ 

② Draw $\mathrm{Tr}(AB)$ and $\mathrm{Tr}(BA)$ for matrices A and B. Are these quantities equal?

③ Draw $\mathrm{Tr}(ABC)$, $\mathrm{Tr}(ACB)$, and $\mathrm{Tr}(BAC)$ for matrices A, B, C.
Which of these are equal?