

张量网络

林雨轩

June-July 2025

1 基本想法

多体问题中，我们所考虑的量子态往往具有复杂的表达方式。比如说 N 个自旋 $\frac{1}{2}$ ，如果完整表达需要用一个 2^N 维的向量。需要注意并不是 N 个，因为多体量子态往往具有复杂的纠缠关系。

以二体问题为例， $|\psi\rangle = a_1|00\rangle + a_2|01\rangle + a_3|10\rangle + a_4|11\rangle$ ，也可用向量 $(a_1, a_2, a_3, a_4)^T$ 来表示 $|\psi\rangle$ 。如果要表达算符的话，则需使用 $2^N \times 2^N$ 的矩阵。通常我们关心的都是哈密顿量 H 的本征值和本征态，而对于如此庞大的矩阵直接进行本征值和本征态的求解的复杂度（时间和空间）都将非常之高。

我们期待找到简化问题的方法。首先我们需要说明问题是可简化的。

一般而言，我们的目标是求解各种形式的 H 的本征值与本征态，而往往这些本征态都具有一定的对称性可以直观看出，意味着我们关心的态通常都占不满整个 2^N 维空间，因此我们相信可以利用对称性简化态的表达。

以上是从抽象层面进行的考虑，下面给出一种形式上的化简思路，以希望从中获得启发。

考虑如下例子：

$$\text{tr} \left(\begin{pmatrix} a_{11} & a_{12} \\ a_{13} & a_{14} \end{pmatrix} \begin{pmatrix} a_{21} & a_{22} \\ a_{23} & a_{24} \end{pmatrix} \begin{pmatrix} a_{31} & a_{32} \\ a_{33} & a_{34} \end{pmatrix} \right)$$

$$= a_{11}a_{21}a_{31} + a_{12}a_{23}a_{31} + a_{11}a_{22}a_{33} + a_{12}a_{24}a_{33} + a_{13}a_{21}a_{32} + a_{13}a_{23}a_{32} + a_{13}a_{22}a_{34} + a_{14}a_{24}a_{34}$$

通常而言， $\begin{pmatrix} a_{n1} & a_{n2} \\ a_{n3} & a_{n4} \end{pmatrix}$ 具有一致的结构，从以上例子可以看出，利用矩阵乘法的表达显然更为简洁，尤其是粒子数 N 很大的时候。需要注意这里的 tr 得到的并不一定要是一个数，而可以是一个态，比如说令 $a_{ni} = |i^n\rangle$ ，其中 n 代表粒子编号，则 $a_{11}a_{21}a_{31} = |111\rangle$ 。

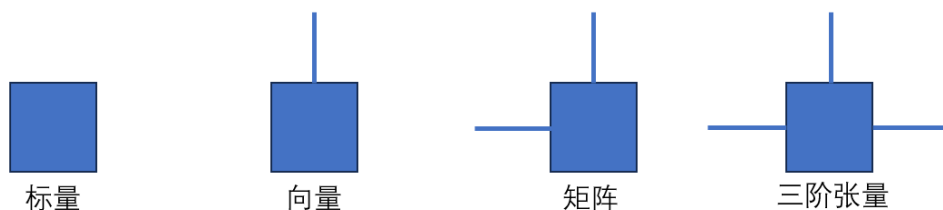
从以上的例子中我们得到启发，利用矩阵乘法（可以进一步推广到张量乘法），则可以简洁而完整的表述出一些态加和的结果。事实上在一维伊辛模型计算配分函数的时候，我们引入了转移矩阵进行化简计算，正是利用了这样的思想。

总结来说，基本思想有两点。其一是某些项能够写在一起（说明系数相同或者具有某种规律）是因为体系的对称性；而如何优雅地表述这些项的加和则需要巧妙地设计矩阵（张量）的乘法。

2 符号表示

2.1 张量表示

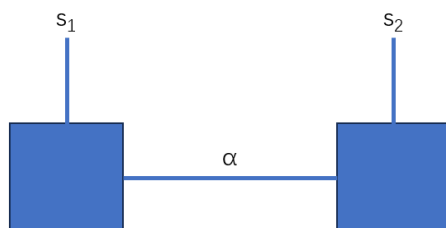
我们使用如下符号化的方式来简化表达形式：



容易看出其中脚的数目即为张量的阶数。以上为自由的脚，其实只起到了标定自由度的作用，比如说没有任何约束的 N 个自旋 $\frac{1}{2}$ 可以用下图表示：



上图可以是一个 N 阶张量，每一阶的维度为 2，同样也可以将其压平成为一个 2^N 维的向量。不过上述表示由于没有任何结构，并不能带来实际的化简。真正的简化来源于张量乘法，我们用下图定义：



上图表示一个 $s_1 \times \alpha$ 和一个 $\alpha \times s_2$ 的矩阵的乘积，得到的结果是一个 $s_1 \times s_2$ 的矩阵，其中的键 α 相当于矩阵乘法中的哑指标。

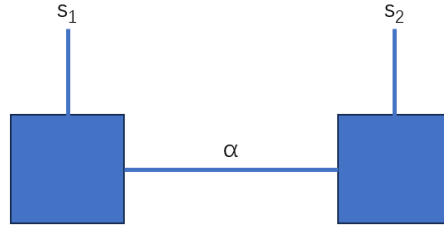
2.2 键

其实脚也可以当作是一端自由的键。从以上例子中我们容易总结出键的合并规律。连接相同两端（自由端视为同一端）的键总是可以合并，合并后张量阶数下降，但是键的重数为原先键重数的乘积。

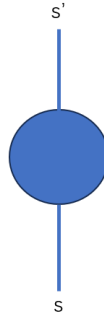
需要注意的是如上张量里面放的都是数，我们还没有赋予其物理意义。从物理上说，利用以上的符号既可以表达态也可以表达算符，当然也可以表示一个数（内积或平均值）。一般而言竖直方向的键带来物理含义（ket 和 bra），水平方向的键表征的是求和规律（也可以总结为某种量子数）。

2.2.1 竖直方向的键

$$\text{表达态: } |\psi\rangle = \sum_{s_1, s_2} \sum_{\alpha} A_{\alpha}^{s_1} A_{\alpha}^{s_2} |s_1\rangle \otimes |s_2\rangle$$



$$\text{表达算符: } \hat{O} = \sum_{s, s'} O^{s, s'} |s'\rangle \langle s|$$



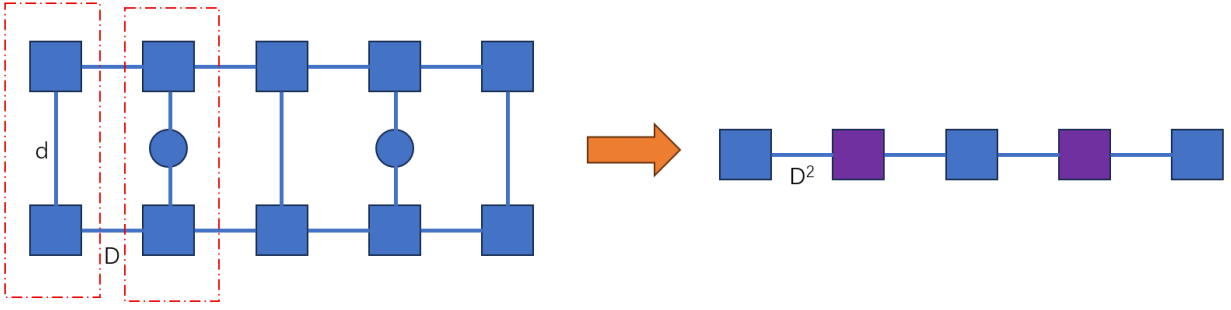
我们统一记上脚（上端自由）为 $ket |s\rangle$ ，下脚为 $bra \langle s|$ ，这样可以统一地理解上述两种表达。

在赋予了张量网络物理意义之后，我们可以放心地把竖直方向自由的键（脚）也进行求和，因为此时不同脚的基矢不同，并不会造成歧义。之后的理解中都可以带上物理含义去理解竖直方向的键，并默认对自由键进行求和。

专门区分开竖直方向的键的原因在于，由于带有物理意义，在实际表示中我们不把可以选择不这个方向的键展开为新的一个维度，而是统一带上基矢并求和（因为我们总是要对竖直方向的键求和，而张量乘法满足分配律，因此先把元素求和也是可行的）。把这样的竖直键称为自由键，因为它的求和方式不需要通过矩阵乘法的横纵坐标匹配，而是直接加和。通过之后的例子可以更具体地理解。

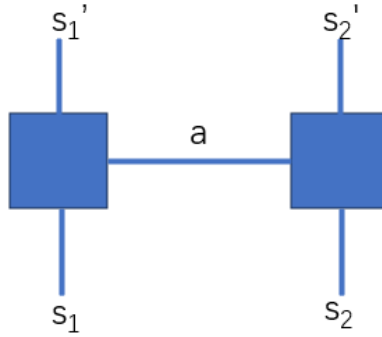
当然也可以表达数，或者说某个计算结果：

$$\langle \psi | \sigma_i \sigma_j | \psi \rangle = \sum_{s'_1 \dots s'_N} \sum_{\alpha'_1 \dots \alpha'_N} A_{\alpha'_1}^{s'_1} A_{\alpha'_1 \alpha'_2}^{s'_2} \dots A_{\alpha'_{N-1}}^{s'_N} \langle s'_1 s'_2 \dots s'_N | \sigma_i \sigma_j \sum_{s_1 \dots s_N} \sum_{\alpha_1 \dots \alpha_N} A_{\alpha_1}^{s_1} A_{\alpha_1 \alpha_2}^{s_2} \dots A_{\alpha_{N-1}}^{s_N} | s_1 s_2 \dots s_N \rangle$$

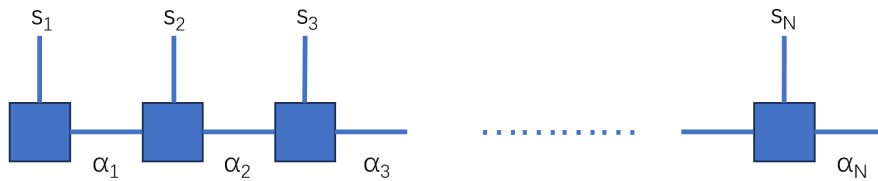


2.2.2 水平方向的键

水平方向的键是用于求和（通过矩阵乘法体现指标匹配关系）的，表征的是某种对称性。考虑算符 $\sigma_1 \cdot \sigma_2$ ，形式化表达如下图所示，水平键指标为 $a = x, y, z$ 。

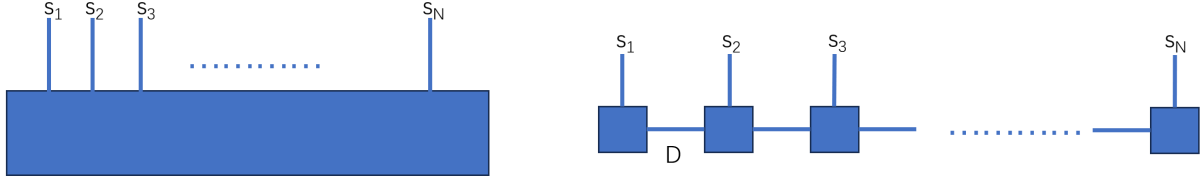


所有水平方向的键最后都要用于求和，按理说是不会出现所谓自由键的。不过有些时候为了更清晰地展示计算过程，我们可能也会形式地在水平方向上出现一端自由的键。此时该自由键是不进行求和的，而是作为一个新的量子数来进行表示。参考如下例子，竖直方向自由键默认求和，而水平方向自由键默认给定，定义为新量子数，用于与后续部分组合：



定义 $|\alpha_i\rangle = \sum_{\alpha_{i-1}, s_i} A_{\alpha_{i-1}\alpha_i}^{s_i} |\alpha_{i-1}\rangle |s_i\rangle$ 为新的量子态，表达的含义为一些自旋基矢的组合，体现系统的某种对称性。

3 矩阵乘积态 Matrix Product State (MPS)



左图中，我们需要 2^N 个参数来表达，参数规模随着 N 指数上升，计算开销巨大。

而右图中，量子态表述为 $|\psi\rangle = \sum_{s_1 \dots s_N} \sum_{\alpha_1 \dots \alpha_{N-1}} A_{\alpha_1}^{s_1} A_{\alpha_1 \alpha_2}^{s_2} \dots A_{\alpha_{N-1}}^{s_N} |s_1 s_2 \dots s_N\rangle$ ，其参数规模为 $N \cdot d \cdot D^2$ ，随着 N 线性增长，计算开销明显降低。

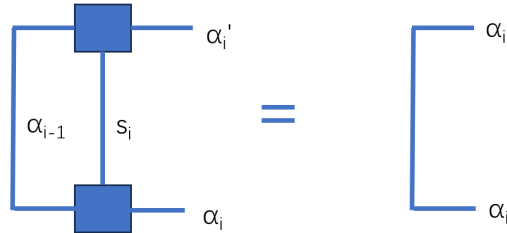
本节目的在于研究这种参数压缩方法的原理与使用方法。

3.1 重整化群思想

实际上之前所述的水平方向键所对应的新的态 $|\alpha_i\rangle = \sum_{\alpha_{i-1}, s_i} A_{\alpha_{i-1} \alpha_i}^{s_i} |\alpha_{i-1}\rangle |s_i\rangle$ 即为所谓的重整化过程。我们所做的是将局部的自由度 s_i 通过若干投影矩阵 A_{α}^s 变为新的有效自由度 α 。

现在我们试图找到这一重整化过程带来的约束条件。自然地，我们希望新的态 $|\alpha\rangle$ 之间是相互正交的，即 $\langle \alpha'_i | \alpha_i \rangle = \delta_{\alpha'_i \alpha_i}$ 。

符号化表达如下所示：



写成矩阵乘积形式：

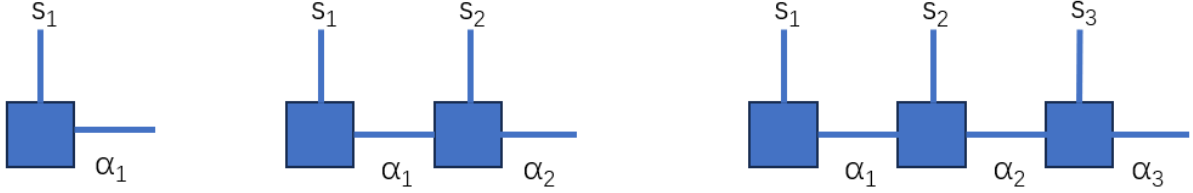
$$\delta_{\alpha'_i \alpha_i} = \sum_{\alpha'_{i-1}, s'_i} A_{\alpha'_{i-1} \alpha'_i}^{s'_i} \sum_{\alpha_{i-1}, s_i} A_{\alpha_{i-1} \alpha_i}^{s_i} \langle \alpha'_{i-1}, s'_i | \alpha_{i-1}, s_i \rangle = \sum_{\alpha_{i-1}, s_i} A_{\alpha_{i-1} \alpha'_i}^{s_i*} A_{\alpha_{i-1} \alpha_i}^{s_i}$$

如果把 s_i, α_{i-1} 合并在一起作为矩阵的行指标， α_i 作为矩阵的列指标，即 A 成为一个 $(Dd) \times D$ 的矩阵，则有：

$$A^\dagger A = I$$

需要注意这并不意味着 $AA^\dagger = I$ 。

下面考虑一个更加具体的例子，考虑 N 体哈密顿量 $H = \sum_{i=1}^{N-1} \sigma_i \cdot \sigma_{i+1}$ 。
我们希望寻找不同本征值对应的可以写成 MPS 形式的本征态，用如下方式表达：



最右端的自由键用来区分不同的本征态。同时我们期待递推地求解本征态，因此假设 N 体量子态的前 $N-1$ 个粒子自然是前 $N-1$ 体哈密顿量的本征态，这样我们每次只需求解 $A_{\alpha_{N-1}\alpha_N}^{s_N}$ 。

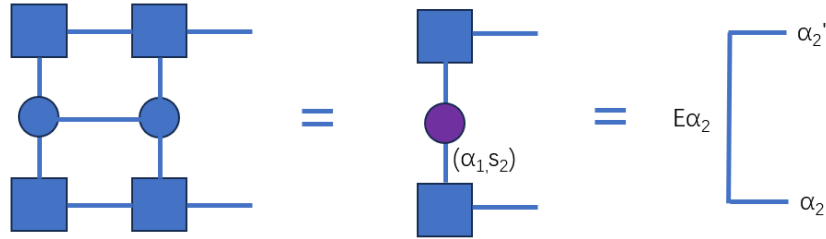
对于单体问题而言该哈密顿量不带来任何约束，因此自然有 $A_{\alpha_1}^{s_1} = \delta_{s_1\alpha_1}$ 。

对于二体问题，本征态需要满足条件： $\sigma_1 \cdot \sigma_2 |\alpha_2\rangle = E_{\alpha_2} |\alpha_2\rangle$, $\langle \alpha'_2 | \alpha_2 \rangle = \delta_{\alpha'_2\alpha_2}$ 。

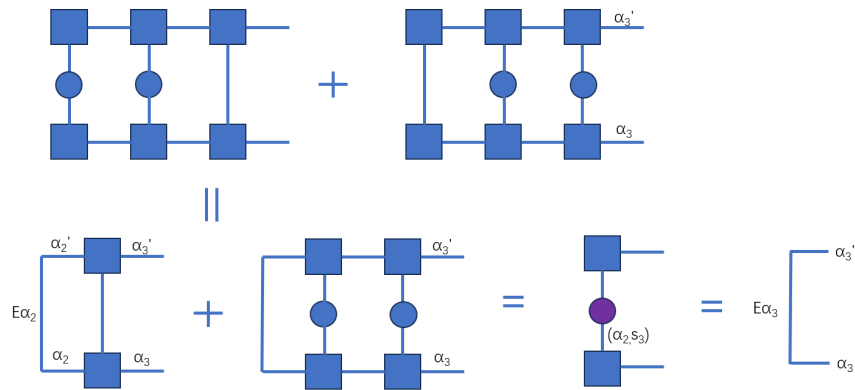
$$|\alpha_2\rangle = \sum_{s_1, s_2, \alpha_1} A_{\alpha_1\alpha_2}^{s_2} A_{\alpha_1}^{s_1} |s_2\alpha_1\rangle = \sum_{s_2, \alpha_1} A_{(\alpha_1, s_2)}^{\alpha_2} |s_2\alpha_1\rangle$$

我们把 $|s_2\alpha_1\rangle$ 作为正交基矢，想要求解向量 $A_{(\alpha_1, s_2)}^{\alpha_2}$ ，其实就是求哈密顿量矩阵 $\langle s'_2\alpha'_1 | \hat{H} | s_2\alpha_1 \rangle$ 的本征向量。

用图形化的方式即：



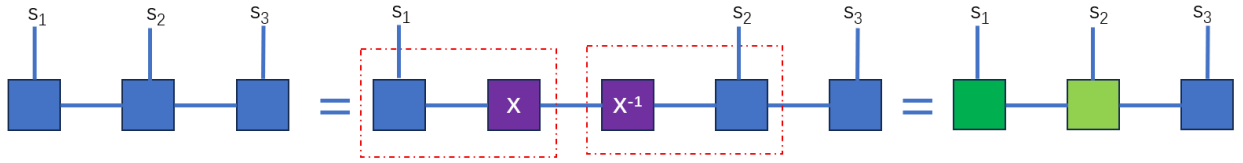
对于三体问题，直接使用图形化的方式求解：



参照以上过程可以推广到 N 体情形。事实上我们仔细思考以上过程，会发现其实和直接对哈密顿量在 2^N 维的基上的矩阵求解本征值和本征态没有区别，因为 $N-1$ 体哈密顿量的本征向量总是构成 2^{N-1} 维空间上的一组正交基。也就是说不进行任何截断的话， α_N 的重数将是 2^N ，并没有起到任何简化效果。只有将键重数始终控制在某个常数时 MPS 态才是简单的，因此接下来我们的任务就是要找到一种好的近似方法进行截断。

3.2 正则形式

容易想到 MPS 的形式并不是唯一的，我们总是可以插入一些新的矩阵 X, X^{-1} 并进行相应的划分：



要想得到标准化的形式（正则形式），需要对 MPS 的组成部分的矩阵有一定要求。常用的方法是进行 SVD。

线性代数的知识告诉我们，一个 $m \times n$ 的矩阵 M 总是可以进行 SVD (Singular Value Decomposition)，具体地：

$$M = U\Lambda V^\dagger$$

我们直接使用紧致形式， U 是 $m \times r$ 的等距矩阵，($U^\dagger U = I$)， Λ 是 $r \times r$ 对角元大于 0 的对角阵， V^\dagger 是 $r \times n$ 的等距矩阵， $V^\dagger V = I_{r \times r}$ 。注意等距矩阵与酉矩阵的区别，其并非方阵，对于上述情形有 $UU^\dagger = P, P^2 = P$ ，即 UU^\dagger 是一个投影矩阵。

SVD 的重要应用就是可以进行量子态的 Schmidt 分解，并衡量其纠缠程度，当然所针对的是二量子系统，即量子态是纯态且可以表述为双正交基的叠加形式。

$|\psi\rangle = \sum_{\alpha, \beta} A_{\alpha\beta} |\alpha_A\rangle \otimes |\beta_B\rangle$ ，其中 $A_{\alpha\beta}$ 是一个 $m \times n$ 的矩阵，可以进行 SVD。

$$|\psi\rangle = \sum_{\alpha, \beta, \gamma} U_{\alpha\gamma} \Lambda_\gamma V_{\gamma\beta}^\dagger |\alpha\rangle_A \otimes |\beta\rangle_B = \sum_\gamma \Lambda_\gamma (\sum_\alpha U_{\alpha\gamma} |\alpha\rangle_A) \otimes (\sum_\beta V_{\gamma\beta}^\dagger |\beta\rangle_B)$$

记 $|\phi_\gamma\rangle_A = \sum_\alpha U_{\alpha\gamma} |\alpha\rangle_A$ ， $|\phi_\gamma\rangle_B = \sum_\beta V_{\gamma\beta}^\dagger |\beta\rangle_B$ 。

通过 $U^\dagger U = V^\dagger V = I$ 可以证得 $|\phi_\gamma\rangle_A, |\phi_\gamma\rangle_B$ 的正交性：

$$\langle \phi_{\gamma'} | \phi_\gamma \rangle_A = \sum_{\alpha, \alpha'} U_{\gamma'\alpha'}^\dagger U_{\alpha\gamma} \delta_{\alpha\alpha'} = \sum_\alpha U_{\gamma'\alpha}^\dagger U_{\alpha\gamma} = (U^\dagger U)_{\gamma'\gamma} = \delta_{\gamma'\gamma}。$$

于是有 $|\psi\rangle = \sum_\gamma \Lambda_\gamma |\phi_\gamma\rangle_A \otimes |\phi_\gamma\rangle_B$ 。我们成功地把 $m \times n$ 个正交态整合到了 r 个正交的态上。

如果 $r = 1$ 则体系无纠缠，否则有纠缠，利用纠缠熵衡量体系纠缠程度。

$$\text{约化密度矩阵 } \rho_A = \text{tr}(|\psi\rangle \langle \psi|)_B = \sum_\gamma \Lambda_\gamma^2 |\phi_\gamma\rangle_A \langle \phi_\gamma|_A。$$

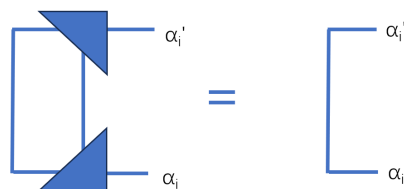
纠缠熵 $S = -\text{tr}(\rho_A \ln \rho_A) \leq \ln r$ ，最后的估计值来源于最大纠缠情况 $\rho_A = \text{diag}(\frac{1}{r}, \frac{1}{r}, \dots, \frac{1}{r})$ 。

考察正则化过程的时间复杂度，如果想要完整求解必然会得到与矩阵规模 $m \times n$ 直接相关的开销，因此如果矩阵太大则依然无法进行分解，不过我们并不需要完整的奇异值谱，而是仅保留重要的几项，即所谓的截断。如果使用迭代算法，比如 Lanczos 迭代，我们的确可以在不大的时间复杂度（与原始矩阵大小无关）下得到最大的几个特征值。必须指出，这里我们能够直接对奇异值谱进行截断的原因在于我们实际上进行了量子态的正交分解，并选取了最大的几个本征态。而如果只在某个局部进行这样的分解并截断，并不是对于整个量子态的 Schmid 分解，因此这样近似的结果并不好。原则上，我们总是要在正则形式上做截断。

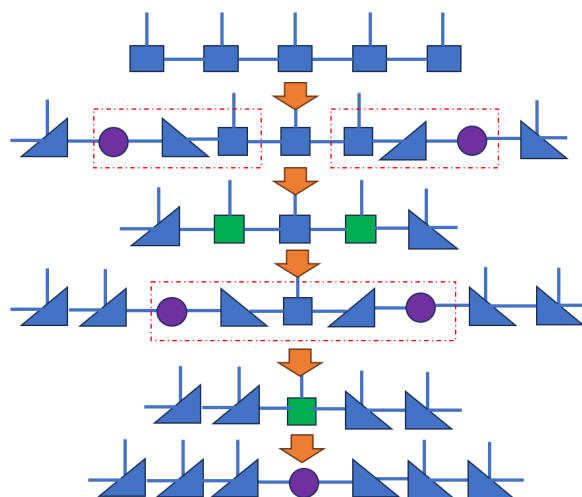
下面用图形化的方式表述正则化过程：



这里利用直角三角形指代等距矩阵，直角边短，因此是出口方向（不是与其厄米转置相乘时缩并的指标），其天然满足如下关系：



总是可以连续地进行正则化过程，直至得到最终的中心正交形式：



总系统被分割为了 $\sum_{\gamma} \Lambda_{\gamma} |L_{\gamma}\rangle \otimes |R_{\gamma}\rangle$ 的形式。

3.3 面积律

考虑一个高度纠缠的态，总共有 $2L$ 个自旋 $\frac{1}{2}$ ，将其分为左右两半，矩阵的维度为 2^L ，纠缠熵 $S \sim \ln(2^L) \sim L$ 。而对于 MPS 而言，纠缠熵 $S \sim \ln(D_{max})$ ，与系统尺寸无关。之所以叫做面积律是因为以上规律可以推广为系统纠缠熵与系统面积成正比，一维系统的面积为常数，体积为系统尺寸。

我们反过来利用上面的结论，如果想要得到 MPS，则必然要求量子态的纠缠熵与系统尺寸无关，即体系必然处在一个低纠缠的状态。我们想知道什么样的哈密顿量能够使体系的纠缠程度较低。

此处我们直接给出两种判断，详细的证明见参考论文，我们姑且认为结论正确并给出一些物理上的理解。

第一是体系的相互作用是局域的，哈密顿量是一些局域哈密顿量的组合，通常而言可能是相邻的两个粒子之间。局域的相互作用带来的纠缠程度低是一件可以理解的事，有时我们还能得到指数衰减的关联函数（关联长度有限），此时只有系统的边界处（表面）受到外界环境的影响，某种意义上说也是面积律的一种直观体现。

第二是体系在热力学极限下具有能隙，即基态与第一激发态之间有有限的能级差。量子场论的结论告诉我们，关联长度 $\xi \sim \frac{1}{\Delta}$ ，其中 Δ 为能隙。我们可以形式上理解以上规律，激发量子涨落的时间尺度是 $\frac{\hbar}{\Delta}$ ，激发传递的速度 v 是一个有限的值，则激发能够影响的区域（即关联长度）应该反比于能隙。当然如何判断体系是否具有能隙同样也是一件困难的事，Haldane 的结论指出，具有 $SU(n)$ 对称性的自旋系统，若自旋为半整数则无能隙，自旋为整数则有能隙。

总结来说，可以用 MPS 表示的系统应该满足要求：局域相互作用（最好是最近邻），热力学极限下具有能隙。

3.4 MPS 示例

我们来看两个能够用 MPS 形式表出的例子来理解之前提到的低纠缠概念以及 MPS 表示的便利性。注意这里我们并不从物理上考虑，而是直接给出某些量子态的一般形式以及 MPS 形式，从数学的角度上验证二者是等价的。

3.4.1 GHZ 态

$|\psi\rangle = |00\dots 0\rangle + |11\dots 1\rangle$ ，容易知道这是一个低纠缠的态，虽然它已经可以很简单地表述，但我们仍然希望给出 MPS 形式，以具体理解 MPS 的得出方式。

我们期待找到 $g_1 g_2 \dots g_N = \begin{pmatrix} |00\dots 0\rangle & 0 \\ 0 & |11\dots 1\rangle \end{pmatrix}$ ，则 $|\psi\rangle = \text{tr}(g_1 g_2 \dots g_N)$ 。

容易想到 $g_1 = \begin{pmatrix} |0\rangle_1 & 0 \\ 0 & |1\rangle_1 \end{pmatrix}$ ， $g_2 = \begin{pmatrix} |0\rangle_2 & 0 \\ 0 & |1\rangle_2 \end{pmatrix}$ ， $g_1 g_2 = \begin{pmatrix} |0\rangle_1 \otimes |0\rangle_2 & 0 \\ 0 & |1\rangle_1 \otimes |1\rangle_2 \end{pmatrix}$ 。

于是 $|\psi\rangle = \begin{pmatrix} |0\rangle_1 & |1\rangle_1 \end{pmatrix} \begin{pmatrix} |0\rangle_2 & 0 \\ 0 & |1\rangle_2 \end{pmatrix} \begin{pmatrix} |0\rangle_3 & 0 \\ 0 & |1\rangle_3 \end{pmatrix} \dots \begin{pmatrix} |0\rangle_{N-1} & 0 \\ 0 & |1\rangle_{N-1} \end{pmatrix} \begin{pmatrix} |0\rangle_N \\ |1\rangle_N \end{pmatrix}$ 。

不过以上写法容易产生歧义，因为实际上使用了乘法分配率将对于每个自旋的求和合并写在了一个矩阵里，具体地：

$$\begin{aligned}
|\psi\rangle &= \sum_{s_1 \dots s_n} \sum_{\alpha_1 \dots \alpha_{N-1}} A_{\alpha_1}^{s_1} A_{\alpha_1 \alpha_2}^{s_2} \dots A_{\alpha_{N-1}}^{s_N} |s_1 s_2 \dots s_N\rangle \\
&= \sum_{\alpha_1 \dots \alpha_{N-1}} (A_{\alpha_1}^{s_1=0} |0\rangle_1 + A_{\alpha_1}^{s_1=1} |1\rangle_1) (A_{\alpha_1 \alpha_2}^{s_2=0} |0\rangle_2 + A_{\alpha_1 \alpha_2}^{s_2=1} |1\rangle_2) \dots \\
&= \left(\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} |0\rangle_1 + \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} |1\rangle_1 \right) \left(\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} |0\rangle_2 + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} |1\rangle_2 \right) \dots \\
&= \begin{pmatrix} |0\rangle_1 & |1\rangle_1 \end{pmatrix} \begin{pmatrix} |0\rangle_2 & 0 \\ 0 & |1\rangle_2 \end{pmatrix} \begin{pmatrix} |0\rangle_3 & 0 \\ 0 & |1\rangle_3 \end{pmatrix} \dots \begin{pmatrix} |0\rangle_{N-1} & 0 \\ 0 & |1\rangle_{N-1} \end{pmatrix} \begin{pmatrix} |0\rangle_N \\ |1\rangle_N \end{pmatrix}
\end{aligned}$$

3.4.2 Kitaev's Majorana 链

$$|\psi\rangle_{\text{even}} = (1 + a_1^\dagger a_2^\dagger + a_1^\dagger a_3^\dagger + \dots + a_1^\dagger a_2^\dagger a_3^\dagger a_4^\dagger + \dots) |0\rangle,$$

$$|\psi\rangle_{\text{odd}} = (a_1^\dagger + a_2^\dagger \dots + a_1^\dagger a_2^\dagger a_3^\dagger + a_2^\dagger a_3^\dagger a_4^\dagger + \dots) |0\rangle,$$

其中 $|0\rangle$ 表示真空态，偶数态表示总有偶数个粒子被激发，奇数态表示总有奇数个粒子被激发。

想依靠枚举表达出整个态是很复杂的，但是利用 MPS 可以清晰地得到结果。

我们想到可以利用 $\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, $\sigma_x^2 = I$ ，奇数个 σ_x 相乘得到的矩阵总是 σ_x ，偶数个 σ_x 相乘得到的矩阵总是 I 。对于奇数态和偶数态只需要一个取非对角元（取其中之一即可，两个非对角元相等）另一个取对角元（两个对角元相逢）。需要注意的是约束只针对于激发态 $|1\rangle$ ，因此 $|0\rangle$ 所对应的矩阵直接就是单位矩阵。

$$\begin{aligned}
|\psi\rangle_{\text{even}} &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \left(\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} |0\rangle_1 + \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} |1\rangle_1 \right) \dots \left(\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} |0\rangle_N + \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} |1\rangle_N \right) \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\
&= \begin{pmatrix} |0\rangle_1 & |1\rangle_1 \end{pmatrix} \begin{pmatrix} |0\rangle_2 & |1\rangle_2 \\ |1\rangle_2 & |0\rangle_2 \end{pmatrix} \dots \begin{pmatrix} |0\rangle_{N-1} & |1\rangle_{N-1} \\ |1\rangle_{N-1} & |0\rangle_{N-1} \end{pmatrix} \begin{pmatrix} |0\rangle_N \\ |1\rangle_N \end{pmatrix} \\
|\psi\rangle_{\text{odd}} &= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \left(\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} |0\rangle_1 + \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} |1\rangle_1 \right) \dots \left(\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} |0\rangle_N + \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} |1\rangle_N \right) \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\
&= \begin{pmatrix} |1\rangle_1 & |0\rangle_1 \end{pmatrix} \begin{pmatrix} |0\rangle_2 & |1\rangle_2 \\ |1\rangle_2 & |0\rangle_2 \end{pmatrix} \dots \begin{pmatrix} |0\rangle_{N-1} & |1\rangle_{N-1} \\ |1\rangle_{N-1} & |0\rangle_{N-1} \end{pmatrix} \begin{pmatrix} |0\rangle_N \\ |1\rangle_N \end{pmatrix}
\end{aligned}$$

因为能写出 MPS 形式，我们可以断言 Kitaev's Majorana 链是低纠缠的。

3.5 AKLT 模型

以上的两个构建 MPS 的例子是从纯数学的角度出发，猜出了如何写成 MPS，并不涉及物理（没有给出是哪个哈密顿量的本征态）。接下来给出的 MPS 不仅是严格的结果，且我们能够知道它是某一个哈密顿量的唯一的基态。

3.5.1 AKLT 态的构建

考虑自旋为 1 的一维系统，其哈密顿量为 $H = \sum \cos \theta (\mathbf{S}_i \cdot \mathbf{S}_{i+1}) + \sin \theta (\mathbf{S}_i \cdot \mathbf{S}_{i+1})^2$ 。

当 $\theta \in (-\frac{\pi}{4}, \frac{\pi}{4})$ 时称为 Haldane 相, 其具有能隙且关联函数指数衰减。特别地, 在 θ 处于某个角度时, 基态可以被解析求解, 称为 AKLT 态。

两个自旋 1 耦合, 总自旋可以为 $0, 1, 2: 1 \otimes 1 = 0 \oplus 1 \oplus 2$ 。

考虑一种特殊的多体量子态, 满足相邻两个粒子的总自旋只能为 0, 1。引入投影算符 $P_{i,i+1}^{s_t=2}$, 表示投影到粒子 i 与粒子 $i+1$ 的总自旋为 2 的子空间。对于上述特殊的态, 容易知道其为算符 $\sum_i P_{i,i+1}^{s_t=2}$ (半正定) 的本征态, 且本征值为 0, 说明这是基态。现在使用自旋算符 $\mathbf{S}_i \cdot \mathbf{S}_{i+1}$ 来表达投影算符 $P_{i,i+1}^{s_t=2}$ 。

$$\begin{aligned}\mathbf{S}_i \cdot \mathbf{S}_{i+1} &= \frac{1}{2}((\mathbf{S}_i + \mathbf{S}_{i+1})^2 - \mathbf{S}_i^2 - \mathbf{S}_{i+1}^2) = \frac{1}{2}(\mathbf{S}_i^2 + \mathbf{S}_{i+1}^2 - \mathbf{S}_{i+1}^2) \\ &= \frac{1}{2}(P_0 \cdot 0 \cdot (0+1) + P_1 \cdot 1 \cdot (1+1) + P_2 \cdot 2 \cdot (2+1) - 1 \cdot 2 \cdot 2) = P_1 + 3P_2 - 2,\end{aligned}$$

同时投影算符满足 $P_a P_b = P_a \delta_{ab}$, $P_0 + P_1 + P_2 = 1$ 。

于是 $(\mathbf{S}_i \cdot \mathbf{S}_{i+1})^2 = P_1 + 9P_2 + 4 - 4P_1 - 12P_2 = 4 - 3P_1 - 3P_2$ 。

有 $3\mathbf{S}_i \cdot \mathbf{S}_{i+1} + (\mathbf{S}_i \cdot \mathbf{S}_{i+1})^2 = 6P_2 - 2 \Rightarrow P_2 = \frac{1}{2}\mathbf{S}_i \cdot \mathbf{S}_{i+1} + \frac{1}{6}(\mathbf{S}_i \cdot \mathbf{S}_{i+1})^2 + \frac{1}{3}$ 。

也就是说 $H = \sum_i \frac{1}{2}\mathbf{S}_i \cdot \mathbf{S}_{i+1} + \frac{1}{6}(\mathbf{S}_i \cdot \mathbf{S}_{i+1})^2 + \frac{1}{3}$ 的基态即为我们前述所说的相邻自旋总和不为 2 的态, 亦即 $\tan \theta = \frac{1}{3}$ 的态。

现在希望解析地求解这个态, 需要我们用另一种方式表述相邻自旋总和不为 2 这一条件。

考虑把每个自旋 1 拆成两个自旋 $\frac{1}{2}$, 则两个自旋 1 的和即为 $\frac{1}{2} \otimes \frac{1}{2} \otimes \frac{1}{2} \otimes \frac{1}{2}$ 。如果限定中间两个自旋 $\frac{1}{2}$ 的和为 0, 则总自旋必然不为 2。

用图形化的方式表达如下, 圆圈代表真实的自旋, 水平的键代表约束条件 (两个自旋 $\frac{1}{2}$ 之和为 0):



定义 $P_i = \sum_{s_i} \sum_{\alpha_i, \beta_i} B_{\alpha_i, \beta_i}^{s_i} |s_i\rangle \langle \alpha_i, \beta_i|$, $I_{i,i+1} = \sum_{\beta_i, \alpha_{i+1}} R_{\beta_i, \alpha_{i+1}} |\beta_i\rangle \langle \alpha_{i+1}|$ 。

则我们的目标态可以表述为 $|\psi\rangle_{AKLT} = (\otimes P_i)(\otimes I_{i,i+1})$, 考虑周期性边界条件:

$$\begin{aligned}|\psi\rangle_{AKLT} &= \sum_{\{s\}} \sum_{\{\alpha, \beta\}} B_{\alpha_1, \beta_1}^{s_1} B_{\alpha_2, \beta_2}^{s_2} \dots B_{\alpha_N, \beta_N}^{s_N} |s_1 s_2 \dots s_N\rangle \langle \alpha_1 \beta_1 \alpha_2 \beta_2 \dots \alpha_N \beta_N| \times \\ &\quad \sum_{\{\alpha', \beta'\}} R_{\beta'_1, \alpha'_2} R_{\beta'_2, \alpha'_3} \dots R_{\beta'_N, \alpha'_1} |\alpha'_1 \beta'_1 \alpha'_2 \beta'_2 \dots \alpha'_N \beta'_N\rangle \\ &= \sum_{\{s\}} \sum_{\{\alpha, \beta\}} B_{\alpha_1, \beta_1}^{s_1} B_{\alpha_2, \beta_2}^{s_2} \dots B_{\alpha_N, \beta_N}^{s_N} R_{\beta_1, \alpha_2} R_{\beta_2, \alpha_3} \dots R_{\beta_N, \alpha_1} |s_1 s_2 \dots s_N\rangle \\ &= \sum_{\{s\}} \text{tr}(B^{s_1} R B^{s_2} R \dots B^{s_N} R) = \text{tr}(A_1 A_2 \dots A_N)\end{aligned}$$

$$I = |01\rangle - |10\rangle, R = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, P = |1\rangle \langle 11| + \frac{1}{\sqrt{2}} |0\rangle (\langle 01| + \langle 10|) + |-1\rangle \langle 00|,$$

$$B^{s=-1} = \begin{pmatrix} \sqrt{2} & 0 \\ 0 & 0 \end{pmatrix}, A^{s=-1} = B^{s=-1} R = \begin{pmatrix} 0 & \sqrt{2} \\ 0 & 0 \end{pmatrix},$$

$$B^{s=0} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, A^{s=0} = B^{s=0} R = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix},$$

$$B^{s=1} = \begin{pmatrix} 0 & 0 \\ 0 & \sqrt{2} \end{pmatrix}, A^{s=1} = B^{s=1} R = \begin{pmatrix} 0 & 0 \\ -\sqrt{2} & 0 \end{pmatrix},$$

$$A_i = \begin{pmatrix} -|0\rangle_i & \sqrt{2}|-1\rangle_i \\ -\sqrt{2}|1\rangle_i & |0\rangle_i \end{pmatrix},$$

$$A_i A_{i+1} = \begin{pmatrix} |0\rangle_i |0\rangle_{i+1} - 2|-1\rangle_i |1\rangle_{i+1} & -\sqrt{2}|0\rangle_i |-1\rangle_{i+1} + \sqrt{2}|-1\rangle_i |0\rangle_{i+1} \\ -\sqrt{2}|0\rangle_i |1\rangle_{i+1} + \sqrt{2}|1\rangle_i |0\rangle_{i+1} & |0\rangle_i |0\rangle_{i+1} - 2|1\rangle_i |-1\rangle_{i+1} \end{pmatrix},$$

容易看出两个矩阵的乘积确实没有出现总自旋为 2 的态 $|1\rangle_i |1\rangle_{i+1}, |-1\rangle_i |-1\rangle_{i+1}$ 。

仔细观察，我们可以直接给出关于 AKLT 态的一些模式特征：

注意到 $|1\rangle \rightarrow \sigma_+, |0\rangle \rightarrow \sigma_z, |-1\rangle \rightarrow \sigma_-, \sigma_+^2 = \sigma_-^2 = 0$ ，因此有贡献的矩阵乘积序列必然会是 σ_-, σ_+ 交错出现（中间可以插入任意个数的 σ_z ）。亦即可能得态必然是 $|1\rangle, |-1\rangle$ 交错出现（中间可以插入任意个数的 $|0\rangle$ ）。

一个可能的状态形如：1 0 0 -1 0 1 0 0 0 -1 0 1 -1 0 1，满足删去 0 之后 1, -1 交替出现。

另一个有趣的结论是 $\langle S_i^z \prod_{k=i+1}^{j-1} e^{i\pi S_k^z} S_j^z \rangle = -\frac{4}{9}$ 。

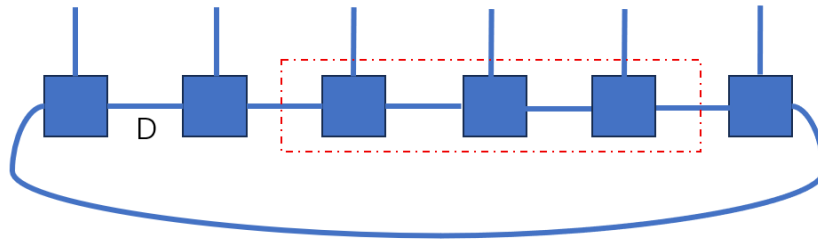
证明如下：对于中间的连乘部分， ± 1 均贡献 -1, 0 贡献 1。假设选定的 i, j 中有 0，则贡献为 0。如果选定的 i, j 同号，则中间必然有奇数个 ± 1 ，因此总贡献为 -1（包括 i, j ）；如果选定的 i, j 异号，则中间有偶数个 ± 1 ，因此总贡献为 1（包括 i, j ）。综上，只要 i, j 均不为 0，则贡献 -1，因此均值即为 -1 与 i, j 均不为 0 的概率乘积 $-1 \times \frac{2}{3} \times \frac{2}{3}$ 。这里每一个粒子出现三种态的概率是相等的。

由于我们已经成功地构建了 MPS，且键的重数很小（=2），因此可以放心地做想进行的运算。计算可得关联函数 $S_i^z S_j^z \sim e^{-\frac{|i-j|}{\xi}}, \xi = \frac{1}{\ln 3}$ 。

3.5.2 MPS 单射性

上一节我们已经成功构造出了 AKLT 态的 MPS 形式，也证明了它是哈密顿量 $H = \sum_i \frac{1}{2} \mathbf{S}_i \cdot \mathbf{S}_{i+1} + \frac{1}{6} (\mathbf{S}_i \cdot \mathbf{S}_{i+1})^2$ 的基态，现在我们想要说明 AKLT 态是该哈密顿量的唯一的基态，且该哈密顿量具有能隙（即在热力学极限下第一激发态和基态能量不连续），并进而推广到更一般形式的 MPS，说明满足何种条件一定可以找到哈密顿量，使得该 MPS 是目标哈密顿量的唯一且具有能隙的基态。

首先给出 MPS 单射性的定义，以便于后续的表述，考虑一般的 MPS：



可以表述为 $|\psi\rangle = \sum_{\alpha, \beta} |\phi_{\alpha\beta}\rangle \otimes |\tilde{\phi}_{\alpha\beta}\rangle$ ，其中前半部分表示在红色框内的部分，后半部分表示不在红色框内的部分。

假设红色框内包含了 L 个粒子，MPS 带来的简化就是将原先的 d^L 个态整合到了 D^2 个态上（显然有 $d^L > D^2$ ）。我们一定能够找到算符 h 使得对于所有的 D^2 个态而言均满足 $h|\phi_{\alpha\beta}\rangle = 0$ ， h 作用于这 L 个粒子上（利用投影到 D^2 个态以外的投影算子组成）。则该 MPS 一定是哈密顿量 $H = \sum_i h_i$ 的基态。这种先确定态再找哈密顿量使得目标态一定是基态的方式正是我们上一节的操作背后的合理性所在，用此方法找到的哈密顿量称为父哈密顿量。

需要注意的是这 D^2 个态不一定是相互独立的。具体来看，我们得到这 D^2 个态的方式是通过 L 个 $D \times D$ 的矩阵乘法，每个矩阵元素代表一个可能的态，不相互独立通常来源于矩阵中某些元素始终为 0。我们定义使得这 D^2 个态相互独立所需要的最少粒子数 L 为单射长度 M 。

以 AKLT 为例，当 $L=1$ 时， $A_i = \begin{pmatrix} -|0\rangle_i & \sqrt{2}|-1\rangle_i \\ -\sqrt{2}|1\rangle_i & |0\rangle_i \end{pmatrix}$ ，四个元素并不相互独立，当 $L=2$ 时， $A_i A_{i+1} = \begin{pmatrix} |0\rangle_i |0\rangle_{i+1} - 2|-1\rangle_i |1\rangle_{i+1} & -\sqrt{2}|0\rangle_i |-1\rangle_{i+1} + \sqrt{2}|-1\rangle_i |0\rangle_{i+1} \\ -\sqrt{2}|0\rangle_i |1\rangle_{i+1} + \sqrt{2}|1\rangle_i |0\rangle_{i+1} & |0\rangle_i |0\rangle_{i+1} - 2|1\rangle_i |-1\rangle_{i+1} \end{pmatrix}$ ，四个元素相互独立，因此 $M=2$ 。

找到单射长度 M 的意义在于，我们有结论，对于作用在 $M+1$ 个粒子上的父哈密顿量，该 MPS 一定是这个父哈密顿量的唯一的具有能隙的基态。这个结论可以严格证明，我们先假定他的正确性。

不过这个结论无法直接应用于 AKLT 模型上，因为实际上我们需要说明对于作用于 2 个粒子上的父哈密顿量 AKLT 态依然是唯一基态，这需要检查所谓的交集条件：

$$V_1 = \{|\psi^\perp\rangle_{12} \otimes |s\rangle_3\}, V_2 = \{|s\rangle_1 \otimes |\psi^\perp\rangle_{23}\}, V_3 = \{|\psi^\perp\rangle_{123}\} \text{ 满足 } V_1 \cap V_2 = V_3.$$

这实际上等价于 $\ker(P_{12} + P_{23}) = \ker(P_{123})$ ，说明两个二体投影算符的核空间相同，都是我们求得的 MPS。

当然并非所有的 MPS 都一定满足单射性，以 GHZ 态为例， $g_1 g_2 \dots g_N = \begin{pmatrix} |00\dots 0\rangle & 0 \\ 0 & |11\dots 1\rangle \end{pmatrix}$ ，始终只有 2 个相互独立的矩阵元。

不过非单射的 MPS 通常可以由单射的 MPS 组合而生成，比如说 GHZ 态可以直接分割为 $|00\dots 0\rangle$ 和 $|11\dots 1\rangle$ 。我们可以反过来理解这种分割的可行性：

$|\phi\rangle = \text{tr}(g_1 g_2 \dots g_N)$, $|\tilde{\phi}\rangle = \text{tr}(\tilde{g}_1 \tilde{g}_2 \dots \tilde{g}_N)$, g_i, \tilde{g}_i 分别是 $D_1 \times D_1, D_2 \times D_2$ 矩阵，这两个态都是单射的。

$$|\psi\rangle = |\phi\rangle + |\tilde{\phi}\rangle = \text{tr}\left(\begin{pmatrix} g_1 & 0 \\ 0 & \tilde{g}_1 \end{pmatrix} \begin{pmatrix} g_2 & 0 \\ 0 & \tilde{g}_2 \end{pmatrix} \dots \begin{pmatrix} g_N & 0 \\ 0 & \tilde{g}_N \end{pmatrix}\right),$$

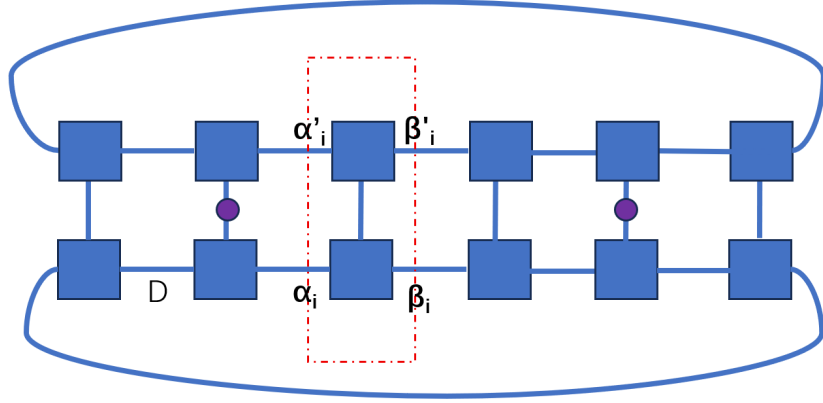
显然这个态不是单射的： $D_1^2 + D_2^2 < (D_1 + D_2)^2$ 。

3.5.3 转移矩阵

想要检查某个 MPS 是否是单射的，最直接的想法是多乘几项，看看各个矩阵元是否相互独立，但显然较为麻烦。另一种简单的方式是通过转移矩阵进行判断。转移矩阵为 $T = \sum_s A_{\alpha'\beta'}^s \otimes A_{\alpha\beta}^s$ 。即转移矩阵是 MPS 矩阵与其共轭的张量积。如果这个转移矩阵的最大本征值是非简并的，说明 MPS 单射；否则 MPS 不是单射。

这里要特别说明一下 T 的计算规则，因为此前我们所有计算都只涉及到态，而没有涉及到平均值。引入转移矩阵的目的是利用转移矩阵的乘法再一次简化求和运算，这里选用张量积的形式进行合并是

有道理的：



用转移矩阵 T 表示红框中的整体。我们想用矩阵乘法的横纵坐标乘积 $T_1 T_2 \dots T_N$ 代替求和 $\sum_{\alpha, \alpha'} \sum_{\beta, \beta'}$ ，注意组合方式，要求我们将 α, α' 分别作为前一个转移矩阵的纵坐标（对于前一个矩阵来讲其实是 $\beta_{i-1}, \beta'_{i-1}$ ）和这个矩阵的横坐标。也就是说在得到转移矩阵时必然要求我们把 α, α' 作为行指标， β, β' 作为列指标。而我们已经得到的矩阵 $A_{\alpha\beta}^s, A_{\alpha'\beta'}^s$ 是分别以 α, β 作为行指标和列指标的，因此必须通过直积的方式得到转移矩阵而非其他合并方式。

检验一下之前的结论，对于 AKLT 态，转移矩阵为 $\begin{pmatrix} 1 & 0 & 0 & 2 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 2 & 0 & 0 & 1 \end{pmatrix}$ ，其特征值为 $3, -1, -1, -1$ ，

最大特征值简并，可以单射。对于 GHZ 态，转移矩阵为 $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ ，其特征值为 $1, 1, 0, 0$ ，最大

特征值简并，不可单射。

有了转移矩阵之后我们可以完成之前对于关联函数的计算，同时可以说明结论，满足单射性的 MPS 具有指数衰减的关联函数：

记中间有圆圈（表示作用算符 O_i ）的转移矩阵为 T^O ，

则 $\langle O_i \rangle = \frac{\langle \psi | O_i | \psi \rangle}{\langle \psi | \psi \rangle} = \frac{\text{tr}(T^{i-1} T^O T^{N-i})}{\text{tr}(T^N)} = \frac{\text{tr}(T^{N-1} T^O)}{\text{tr}(T^N)}$ ，其中 $T = \sum \lambda_n |R_n\rangle \langle L_n|$ ($|\lambda_1| > |\lambda_2| \geq \dots$)。

这里使用到了矩阵求迹的性质，对于循环置换而言，矩阵的迹不改变。即 $\text{tr}(ABC) = \text{tr}(BCA) \neq \text{tr}(BAC)$ 。

取极限有 $\langle O_i \rangle = \lim_{N \rightarrow \infty} \frac{\text{tr}(T^{N-1} T^O)}{\text{tr}(T^N)} = \frac{\langle L_1 | T^O | R_1 \rangle}{\lambda_1}$ 。

$\langle O_i O_j \rangle = \lim_{N \rightarrow \infty} \frac{\text{tr}(T^{i-1} T^O T^{j-i-1} T^O T^{N-j})}{\text{tr}(T^N)} = \frac{\langle L_1 | T^O T^{j-i-1} T^O | R_1 \rangle}{\lambda_1^{j-i+1}}$ 。

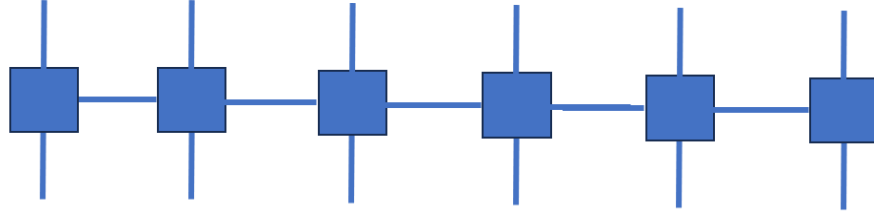
展开到第二阶， $T^{j-i-1} = \lambda_1^{j-i-1} |R_1\rangle \langle L_1| + \lambda_2^{j-i-1} |R_2\rangle \langle L_2|$ 。（注意各个本征向量之间是正交的）。

于是 $\langle O_i O_j \rangle = \frac{\langle L_1 | T^O | R_1 \rangle \langle L_1 | T^O | R_1 \rangle}{\lambda_1^2} + \frac{\langle L_1 | T^O | R_2 \rangle \langle L_2 | T^O | R_1 \rangle}{\lambda_1^{j-i+1}} \lambda_2^{j-i-1}$ 。

可以得到关联函数 $\langle O_i O_j \rangle - \langle O_i \rangle \langle O_j \rangle \sim |\frac{\lambda_2}{\lambda_1}|^{|j-i|} = e^{-\frac{|j-i|}{\xi}}, \xi = \ln|\frac{\lambda_1}{\lambda_2}|$ 。

4 矩阵乘积算符 Matrix Product Operator (MPO)

回顾一下从一个复杂的态到一个 MPS 的过程，我们通过引入了一些虚拟的键克服了指数增涨的问题（核心在于利用奇异值分解进行截断）。同样地，对于一个算符我们也可以进行类似的操作，得到如下形式的算符：



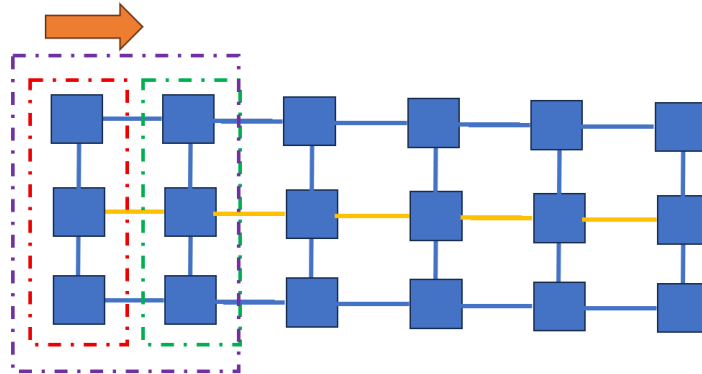
MPO 形式的算符有局域算符用数量较少的键组合（相乘）而成，可以表征一些非局域算符，或者是一些局域算符的和。

考虑总自旋算符 $\sum S_i^z$ ，相把它表述成一些局域项的乘积形式（而非求和），从数学上可以直接构造得到：

$$\begin{pmatrix} I & 0 \\ S_i^z & I \end{pmatrix} \begin{pmatrix} I & 0 \\ S_j^z & I \end{pmatrix} = \begin{pmatrix} I & 0 \\ S_i^z + S_j^z & I \end{pmatrix},$$

$$\text{于是 } \sum S_i^z = \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} I & 0 \\ S_1^z & I \end{pmatrix} \begin{pmatrix} I & 0 \\ S_2^z & I \end{pmatrix} \cdots \begin{pmatrix} I & 0 \\ S_N^z & I \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

得到 MPO 的好处在于进行平均值计算时我们不需要在计算很多遍（对局域算符算平均值然后再求和），而是只需要计算一个 MPO 形式的计算：

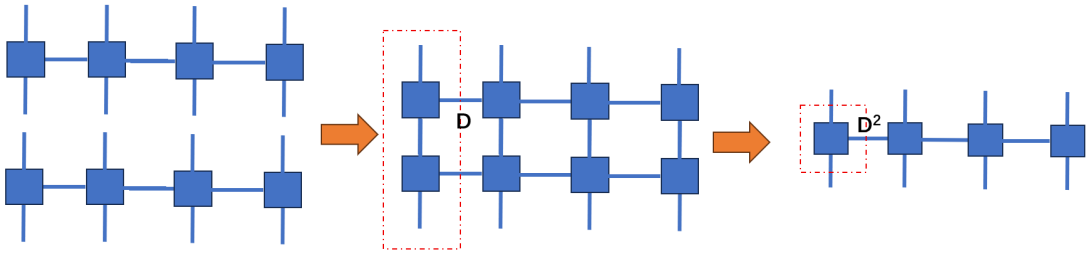


代码上，我们可以设计收缩函数，每次输入红色框和绿色框，将二者进行收缩计算得到紫色框，红

色框和紫色框是同类型的三阶张量，因此可以依次从左向右进行收缩。相比于计算局域算符再求和的方式，我们付出的计算代价是中间那些黄色键的收缩，但是节省了整体收缩的次数，使得计算效率得到提升。

接下来举出更多的例子来展现 MPO 带来的简化：

考虑子算算符的傅里叶变换， $S_q^z = \frac{1}{\sqrt{N}} \sum e^{-iqj} S_j^z$ ，要求解均值 $\langle S_q^z S_{-q}^z \rangle$ ，无需进行 N^2 次计算，通过 MPO 可以进行程序性的化简。先把 S_q^z, S_{-q}^z 分别化成 MPO 形式，二者的乘积实际上就是两个 MPO 的垂直方向的连接，理解这一点可以通过垂直方向的物理意义去看，上方的脚是 ket，下方的脚是 bra，算符的乘积无非是写到一起，被连起来的 bra 和 ket 总是会消去。当然这么做的代价是中间的键重数提高了，但是整体的计算效率得到提升。



海森堡模型中的哈密顿量同样可以写成 MPO 形式：

$$\begin{aligned}
 H = \sum \mathbf{S}_i \cdot \mathbf{S}_{i+1} &= \sum \frac{1}{2} (S_i^+ S_{i+1}^- + S_i^- S_{i+1}^+) + S_i^z S_{i+1}^z, \\
 &= \begin{pmatrix} I & 0 & 0 & 0 & 0 \\ S_i^+ & 0 & 0 & 0 & 0 \\ S_i^- & 0 & 0 & 0 & 0 \\ S_i^z & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} S_i^- & \frac{1}{2} S_i^+ & S_i^z & I \end{pmatrix} \begin{pmatrix} I & 0 & 0 & 0 & 0 \\ S_j^+ & 0 & 0 & 0 & 0 \\ S_j^- & 0 & 0 & 0 & 0 \\ S_j^z & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} S_j^- & \frac{1}{2} S_j^+ & S_j^z & I \end{pmatrix} \begin{pmatrix} I & 0 & 0 & 0 & 0 \\ S_k^+ & 0 & 0 & 0 & 0 \\ S_k^- & 0 & 0 & 0 & 0 \\ S_k^z & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} S_k^- & \frac{1}{2} S_k^+ & S_k^z & I \end{pmatrix} \\
 &= \begin{pmatrix} I & 0 & 0 & 0 & 0 \\ S_i^+ & 0 & 0 & 0 & 0 \\ S_i^- & 0 & 0 & 0 & 0 \\ S_i^z & 0 & 0 & 0 & 0 \\ \frac{1}{2} S_i^- S_j^+ + \frac{1}{2} S_i^+ S_j^- + S_i^z S_j^z & \frac{1}{2} S_j^- & \frac{1}{2} S_j^+ & S_j^z & I \end{pmatrix} \begin{pmatrix} I & 0 & 0 & 0 & 0 \\ S_k^+ & 0 & 0 & 0 & 0 \\ S_k^- & 0 & 0 & 0 & 0 \\ S_k^z & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} S_k^- & \frac{1}{2} S_k^+ & S_k^z & I \end{pmatrix} \\
 &= \begin{pmatrix} I & 0 & 0 & 0 & 0 \\ S_i^+ & 0 & 0 & 0 & 0 \\ S_i^- & 0 & 0 & 0 & 0 \\ S_i^z & 0 & 0 & 0 & 0 \\ \mathbf{S}_i \cdot \mathbf{S}_j + \mathbf{S}_j \cdot \mathbf{S}_k & \frac{1}{2} S_k^- & \frac{1}{2} S_k^+ & S_k^z & I \end{pmatrix}
 \end{aligned}$$

只需取左下角元素即可得到哈密顿量。

如果哈密顿量的形式较复杂，是不同种模型的共同叠加，比如说 $H = \sum \mathbf{S}_i \cdot \mathbf{S}_{i+1} + \mathbf{S}_i \cdot \mathbf{S}_{i+2}$ ，其 MPO 形式就会较为复杂。注意 MPO 形式是不唯一的，通过巧妙地设计可能可以使得键的重数较低，

一般的通用方法为 $H = H_1 + H_2$, 则 $O = \begin{pmatrix} O_1 & 0 \\ 0 & O_2 \end{pmatrix}$ 。我们需要理解的是这个 O 矩阵的横纵指标都是我们所创造的虚拟键, 因此只要让 1,2 两个部分不互相干扰即可, 最终再两端乘上特定矩阵以取出我们想要的矩阵元。这么做的代价是键的重数上升为 $D_1 \times D_2$ 。

5 密度矩阵重整化群 Density Matrix Renormalization Group(DMRG)

回顾之前的内容, 我们得到了一套可以简便地处理量子多体系统的方式。系统的量子态写成 MPS 形式, 哈密顿量写成 MPO 形式, 但是数学上能够严格得到并证明确实是某个哈密顿量的基态波函数的情形非常少。实际情况下, 我们通常面对的是一个哈密顿量, 我们可以想办法将其写成 MPO 形式, 然后我们想要找到其基态能量或者基态波函数。这时候我们可以使用变分法, 试探波函数取成 MPS 形式, 这时候得益于 MPS 和 MPO 形式的简洁性, $E = \frac{\langle \psi | H | \psi \rangle}{\langle \psi | \psi \rangle}$ 是简单的, 因此不断优化可以找到基态能量和波函数。这正是 DMRG 最基本的想法。

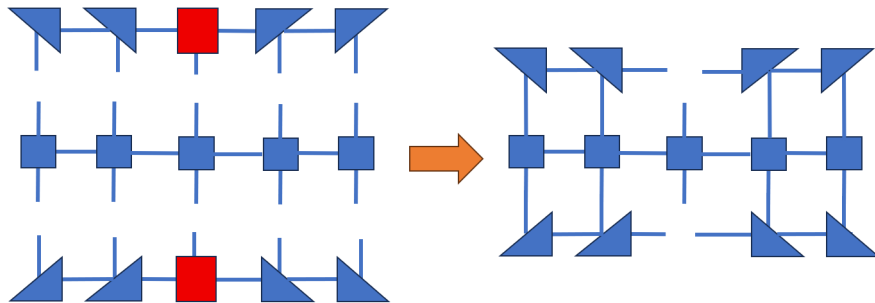
5.1 单点 DMRG

确定了变分法的思路, 参数设置也比较明显, 无非就是矩阵 $\{A_{\alpha_i \beta_i}^{s_i}\}$, 现在的问题是如何去逐步优化参数。一种比较自然的想法是逐点去优化, 反复扫几遍按理说就能够得到最佳参数。下面来具体完善细节。

待优化的参数记为 $A_{\alpha_i \beta_i}^{s_i}$, 需要注意的是该参数是采用中心正则形式后的中心点参数。即量子态表述为 $|\psi\rangle = \sum_{s_i, \alpha_i, \beta_i} A_{\alpha_i \beta_i}^{s_i} |L_{\alpha_i}\rangle \otimes |s_i\rangle \otimes |R_{\beta_i}\rangle$, 其中 $|L_{\alpha_i}\rangle, |R_{\beta_i}\rangle$ 之间是相互正交的。现在我们有一组正交基 $|L_{\alpha_i}\rangle \otimes |s_i\rangle \otimes |R_{\beta_i}\rangle$, 共 $D^2 d$ 个, 需要求 $D^2 d$ 个系数使得量子态是已知哈密顿量的基态。因此我们只需得到哈密顿量在这组 $D^2 d$ 个正交基下的矩阵即可。

即定义矩阵 $H_{eff} = \langle L_{\alpha'_i} s'_i R_{\beta'_i} | H | L_{\alpha_i} s_i R_{\beta_i} \rangle$, 我们想要的是基态能量和本征态 $H_{eff} A = \lambda_{min} A$ 。总是可以使得本征态归一化 $A^\dagger A = 1$ 。

用图示的方法表示以上过程, 注意正则化是重要的, 因为我们必须要得到一组正交基:



右侧的六个端口三个展平为矩阵的一个维度, 即为 H_{eff} 矩阵。

需要注意的是我们只需要知道基态信息，因此并不需要通过把 H_{eff} 矩阵写出并储存的方式求解本征值，利用 Lanczos 算法可以更高效地得到基态，尤其是在键重数 D 较大时。

5.2 Lanczos 算法

Lanczos 算法的核心在于找到一组正交基，使得目标矩阵在这组正交基下是三对角的。具体如下：

令 $A_1, A_2, A_3 \dots$ 相互正交且都满足归一条件， A_1 为初始向量。

$$H_{eff}A_1 = \mu_1 A_1 + v_1 A_2, \mu_1 = A_1^\dagger H_{eff} A_1, v_1 = \|(H_{eff} - \mu_1)A_1\| = A_2^\dagger H_{eff} A_1;$$

$H_{eff}A_2 = v_1^* A_1 + \mu_2 A_2 + v_2 A_3$ ，注意这里 $v_1^* = A_1^\dagger H_{eff} A_2 = (A_2^\dagger H_{eff} A_1)^*$ ，其他的参数可以按照上述方式得出。

$H_{eff}A_3 = v_2^* A_2 + \mu_3 A_3 + v_3 A_4$ ，注意这里没有 A_1 项，是因为 $A_1^\dagger H_{eff} A_3 = (A_3^\dagger H_{eff} A_1)^* = 0$ ，以此类推，每一个递推式只涉及到至多 3 个正交基。

正交基组 $A_1, A_2, A_3 \dots$ 构成的空间叫做 Krylov 子空间，

$$\text{在 Krylov 子空间中, } H_{eff} = \begin{pmatrix} \mu_1 & v_1 & & & \\ v_1^* & \mu_2 & v_2 & & \\ & v_2^* & \mu_3 & v_3 & \\ & & & \ddots & \ddots & \ddots \end{pmatrix}.$$

值得说明的是，在第一次扫描各个格点时，初始向量距离基态较远，因此需要在较大的 Krylov 子空间中进行求解。而在第一遍扫描之后，初始向量距离基态较近，因此在生成 Krylov 子空间上的 H_{eff} 时， v_1 是一个小量，同时，此后得到的 $v_2, v_3 \dots$ 将越来越小，一旦缩小到机器精度时，我们就可以直接进行截断。通常而言可能后续的扫描所需的 krylov 子空间维数只要 3 到 5 阶即能满足要求。

在整个计算过程当中，我们调用 H_{eff} 都是通过直接进行张量计算 $H_{eff}A_i$ ，而不需要像传统的本征值求解方法那样求解所有的矩阵元，从而减小了计算开销。

5.3 两点 DMRG

两点 DMRG 有些时候更为有用，因为其具有更多的可调控性，我们先来关注具体过程，最后总结与单点 DMRG 之间的优劣：

两点 DMRG 就是每次同时优化两个格点所对应的矩阵，即 $|\psi\rangle = \sum_{s_i, s_{i+1}, \alpha_i, \beta_i} A_{\alpha_i, \beta_i}^{s_i, s_{i+1}} |L_{\alpha_i}\rangle \otimes |s_i, s_{i+1}\rangle \otimes |R_{\beta_i}\rangle$ ，可以看出此时的 H_{eff} 是一个 $d^2 D^2 \times d^2 D^2$ 的矩阵。

与单点 DMRG 显著不同之处在于，每一次得到优化后的矩阵 $A_{\alpha_i, \beta_i}^{s_i, s_{i+1}}$ 之后，如何进行保存以进行下两个格点的优化。此时需要再次用到前面所说的 SVD 分解。我们将矩阵写成 $A_{(s_i, \alpha_i)(s_{i+1}, \beta_i)}$ 的形式，即 $dD \times dD$ ，进行 SVD 分解后有效特征值的个数不超过 dD 个，但是也有可能超过 D 个，这意味着横向键的重数发生了变化，图形化的过程如下：



也就是说在优化的过程中 MPS 的横向的键重数是存在持续增加的可能的（每次进行都 $\times d$ ），当然我们也可以通过截断技术，只选择奇异值谱中的前几个来控制键的重数。总而言之，两点 DMRG 的优化过程中可以不断调整键重数以达到更高的计算效率，比如说在第一次扫描时如果距离基态较远，可以扩大键重数，即 H_{eff} 矩阵的维数从而缩小进行 Lanczos 迭代的次数，以更快地达到基态。

两点 DMRG 的另一好处在于利用更大的参数调节范围，可以摆脱局部极小值点，使计算结果更准确。注意我们的变分过程始终是在搜索局部的最小本征值，因而很有可能得不到正确结果。当然其缺点也较为明显，就是随着键的重数增长，计算开销也会相应增大。

5.4 小结

以上我们介绍了张量网络方法处理问题所需要的基本工具和核心方法，整体而言偏向数学工具，可以算作为后续解决物理问题的一个准备。在进入物理部分之前，对该方法的适用性再次做一个小结。张量网络方法克服指数墙的核心在于在 MPS 和 MPO 上处理问题，MPO 的键重数一般不会很大，但是 MPS 作为变分法的优化态，想要得到精确结果实际上需要较大的键重数（ $\sim 10^3 \sim 10^4$ ）。能用常数大小的键代替指数增长的键的核心在于体系的纠缠状态是低的，这通常要求哈密顿量是局域的，物理上，大概可以这么想，如果说相互作用总是局域的，某一个粒子与外界的纠缠不那么大，因此与外界连接所需要的键重数应该也不会很大（至少不应该指数增长）。

此外，我们以上的讨论均针对于一维系统，如果是二维系统，该方法的有效性实际上会明显降低。因为我们总是希望把体系的状态排成一维的，并且要求在这条一维的链上有局域的相互作用。而对于一个二维连接的体系，无论如何将粒子强行排成一维链，总是无法避免引入非局域的相互作用。同时对于更一般的多体系统，比如说量子化学中的某些构型，可能根本不存在局域相互作用的概念，强行把各个原子排成一条链之后相互作用就是乱七八糟地作用在某些格点处，这时候可能就需要更多的键重数才能进行充分近似，该方法的有效性也就不那么明显了。

6 费米子系统

之前的处理中，我们只是把系统当作了一些有相互作用的自旋粒子，忽略了粒子的全同性带来的约束条件。在实际物理情形中，我们更关注电子的行为，本节的目的就在于建立起自旋和费米子之间的映射关系，需要用到著名的 Jordan-Wigner 变换，在讲述过程中将会穿插张量网络的语言以熟悉这样的处理方式。

6.1 二次量子化

处理全同粒子问题时，按照常规做法，波函数形式很复杂， $\psi_B = \frac{1}{\sqrt{2}}(\psi_1(x_1)\psi_2(x_2) + \psi_2(x_1)\psi_1(x_2))$ ， $\psi_F = \frac{1}{\sqrt{2}}(\psi_1(x_1)\psi_2(x_2) - \psi_2(x_1)\psi_1(x_2))$ ，如果把波函数全部展开处理复杂程度不堪设想。因此我们考虑另一种描述量子态的方式。

用 $|n_1 n_2 \dots n_k\rangle$ 描述量子态，其中 $\begin{cases} n_k = 0, 1 & \text{Fermion} \\ n_k = 0, 1, 2 \dots & \text{Boson} \end{cases}$ ，产生和湮灭算符记为 c_i^\dagger, c_i 。只要规定好产生和湮灭算符的运算方式，就可以完全表述出费米子和玻色子的全同性质，无需展开

波函数，使得表述得到了极大的化简。

需要注意的是关于粒子数与模式数的理解。粒子数是各个模式占据数量的总和，这意味这总的模式数可以远多于粒子数。同时粒子数也不再是一个我们需要预设满足的要求，而是成为了后续计算中的一个条件。现在开始处理玻色子和费米子的产生湮灭算符需要满足的条件：

$$\begin{aligned}
& \text{玻色子: } c_i^\dagger |n_1, n_2, \dots, n_i \dots\rangle = \sqrt{n_i + 1} |n_1, n_2, \dots, n_i + 1 \dots\rangle, \\
& c_i |n_1, n_2, \dots, n_i \dots\rangle = \sqrt{n_i} |n_1, n_2, \dots, n_i - 1 \dots\rangle, c_i |0\rangle = 0, \\
& c_i^\dagger c_j^\dagger |n_1, n_2, \dots, n_i \dots\rangle = \sqrt{n_i + 1} \sqrt{n_j + 1} |n_1, n_2, \dots, n_i + 1 \dots, n_j + 1 \dots\rangle, \\
& c_i c_i^\dagger |n_1, n_2, \dots, n_i \dots\rangle = (n_i + 1) |n_1, n_2, \dots, n_i \dots\rangle, c_i^\dagger c_i |n_1, n_2, \dots, n_i \dots\rangle = n_i |n_1, n_2, \dots, n_i \dots\rangle, \\
& \Rightarrow [c_i, c_j] = 0, [c_i^\dagger, c_j^\dagger] = 0, [c_i, c_j^\dagger] = \delta_{ij}. \\
& \text{费米子: } c_i^\dagger |n_{i+1}, n_{i+2}, \dots\rangle = |n_i, n_{i+1}, n_{i+2}, \dots\rangle, \\
& c_i |n_{i+1}, n_{i+2}, \dots\rangle = 0, c_i |n_i, n_{i+1}, n_{i+2}, \dots\rangle = |n_{i+1}, n_{i+2}, \dots\rangle,
\end{aligned}$$

对于 $|n_i, n_j, n_k\rangle$ 的理解是这样的，虽然默认 $n_i = n_j = n_k = 1$ ，但是标号仍是重要的，实际上它代表的是 $\psi_i(x_1)\psi_j(x_2)\psi_k(x_3)$ 反对称化后的态，因此 $|n_i, n_j, n_k\rangle = -|n_j, n_i, n_k\rangle$ 。

而产生湮灭算符总是作用在 $|\rangle$ 中的第一个位置，满足 $c_i |n_j, n_i\rangle = -c_i |n_i, n_j\rangle = -|n_j\rangle$ 。

由此一来反对易性才得到满足，我们希望将其总结成算符的对易或反对易关系形式。

$$\begin{aligned}
& c_i^\dagger c_j^\dagger |n_k, n_l, n_m \dots\rangle = |n_i, n_j, n_k, n_l, n_m \dots\rangle, \\
& c_j^\dagger c_i^\dagger |n_k, n_l, n_m \dots\rangle = |n_j, n_i, n_k, n_l, n_m \dots\rangle = -|n_i, n_j, n_k, n_l, n_m \dots\rangle, \\
& \Rightarrow c_i^\dagger c_j^\dagger = -c_j^\dagger c_i^\dagger, \{c_i^\dagger, c_j^\dagger\} = 0, \{c_i, c_j\} = 0 \text{ (湮灭算符同理可得)}, \\
& i \neq j: c_i c_j^\dagger |n_i, n_k, n_l, n_m \dots\rangle = c_i |n_j, n_i, n_k, n_l, n_m \dots\rangle = -|n_j, n_k, n_l, n_m \dots\rangle, \\
& c_j^\dagger c_i |n_i, n_k, n_l, n_m \dots\rangle = |n_j, n_k, n_l, n_m \dots\rangle \Rightarrow c_i c_j^\dagger + c_j^\dagger c_i = 0, \\
& i = j, c_i c_i^\dagger |n_i, n_k, n_l, n_m \dots\rangle = 0, c_i^\dagger c_i |n_i, n_k, n_l, n_m \dots\rangle = |n_i, n_k, n_l, n_m \dots\rangle, \\
& \Rightarrow c_i c_i^\dagger + c_i^\dagger c_i = 1, \Rightarrow \{c_i, c_j^\dagger\} = \delta_{ij}.
\end{aligned}$$

对于玻色子和费米子共同而言，可以定义占据数算符 $n_i = c_i^\dagger c_i$ ，总粒子数 $N = \sum n_i = \sum c_i^\dagger c_i$ 。

6.2 Jordan Wigner 变换

Jordan Wigner 变换的核心在于实现了自旋和费米子之间的映射，从数学上说这是一种巧妙的变换关系，利用自旋算符构造出了满足费米子反对易关系的产生湮灭算符。但仅从这个角度看，忽视了其中的物理内涵，本节试图展示这种映射关系背后的物理意义。

我们希望找到满足反对易关系的算符： $\{c_i^\dagger, c_j^\dagger\} = 0, \{c_i, c_j\} = 0, \{c_i, c_j^\dagger\} = \delta_{ij}$ 。

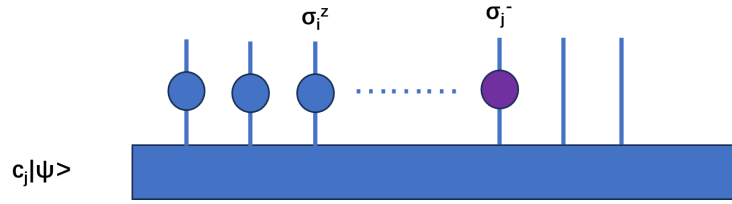
我们想通过自旋升降算符构造出费米子的产生湮灭算符，自旋升降算符满足关系：

$$\begin{aligned}
& [S_i^+, S_j^+] = 0, [S_i^-, S_j^-] = 0, [S_i^+, S_j^-] = 0, (i \neq j), \\
& \{S_i^+, S_i^+\} = 0, \{S_i^-, S_i^-\} = 0, \{S_i^-, S_i^+\} = 1,
\end{aligned}$$

亦即对于单个格点而言，自旋升降算符满足反对易关系，但是对于不同格点而言相差了一个正负号。我们想要做的就是通过某一种构造来调整这个正负号。正面去构造算符是困难的，我们将从另一个角度去思考。

我们假设每一个粒子对应于一个费米子（或者说理解为一种费米子模式），这个假设其实并没有看起来的那么自然，之后将解释原因。

$|\downarrow\rangle \longleftrightarrow |0\rangle, |\uparrow\rangle \longleftrightarrow |1\rangle, |s_1 s_2 \dots s_N\rangle \longleftrightarrow |n_1 n_2 \dots n_N\rangle, n_k$ 可以为 1,0。
 $c_j |n_1 n_2 \dots n_N\rangle = c_j (c_1^\dagger)^{n_1} (c_2^\dagger)^{n_2} \dots (c_N^\dagger)^{n_N} |0\rangle = \prod_{i=1}^{j-1} (-1)^{n_i} (c_1^\dagger)^{n_1} (c_2^\dagger)^{n_2} \dots c_j (c_j^\dagger)^{n_j} \dots (c_N^\dagger)^{n_N} |0\rangle$
 $= \prod_{i=1}^{j-1} (-1)^{n_i} \delta_{n_j, 1} |n_1 n_2 \dots n_j - 1 \dots n_N\rangle,$
 $(-1)^{n_i} = |0\rangle_i \langle 0|_i - |1\rangle_i \langle 1|_i \longleftrightarrow |\downarrow\rangle_i \langle \downarrow|_i - |\uparrow\rangle_i \langle \uparrow|_i = -\sigma_i^z, |0\rangle_j \langle 1|_j \leftrightarrow |\downarrow\rangle_j \langle \uparrow|_j = \sigma_j^-,$
 于是 $c_j |n_1 n_2 \dots n_N\rangle \longleftrightarrow \prod_{i=1}^{j-1} (-\sigma_i^z) (\sigma_j^-) |s_1 s_2 \dots s_N\rangle。$
 $\Rightarrow c_j = \prod_{i=1}^{j-1} (-\sigma_i^z) (\sigma_j^-), c_j^\dagger = \prod_{i=1}^{j-1} (-\sigma_i^z) (\sigma_j^+), c_j^\dagger c_j = \sigma_j^+ \sigma_j^- = \frac{1}{2}(\sigma_j^z + 1)。$
 最后一个式子说明费米子总占据数和自旋向上的粒子数量（而不是粒子总数）相同。
 用图形化的方式表达如下：



现在我们回答一个问题，凭什么一个自旋就对应于一个费米子？为什么没有更多的占据模式？

要回答这个问题，其实需要想明白我们所构造出的这个费米子究竟是什么，或者说系统的模式数量等于粒子数意味着什么。这实际上和系统的哈密顿量有关。考虑海森堡模型 $H = \sum -\mathbf{S}_i \cdot \mathbf{S}_{i+1}$ ，能量的不同激发态取决于相邻自旋是平行还是反平行的，反平行意味着高能级，平行意味着低能级，这二者对应了一个模式的占据与不占据。

考虑基态和第一激发态，应该全局只有一处从平行变成了反平行，但是实际上翻转的自旋数量却不是 1： $\downarrow\downarrow\downarrow\downarrow\downarrow \dots \rightarrow \downarrow\downarrow\downarrow\uparrow\uparrow \dots$ 。

所以说实际上系统的不同模式，对应于自旋链在不同位置处发生了扭转。不过这里所谓的模式，虽然满足费米子的泡利不相容原理，却不满足交换反对称性，但是好在通过扭结我们已经成功的将系统的某种模式和自旋链中的某个粒子对应了起来。现在考虑的是如何让这种扭结模式成为费米子，即交换两次扭结的顺序会带来 -1 的相位。很容易想到我们可以通过自旋的方向来决定引入的相位是 +1 还是 -1，关键在于先扭 A 粒子后扭 B 粒子与先扭 B 粒子后扭 A 粒子之间要差上 -1，这说明相位贡献只能由某一侧（不妨设为前方）的自旋贡献，考虑 B 在 A 后，则 A 前方的相位贡献不变，B 前方的相位贡献前后总是相反的，因为存在 A 扭与不扭的差别。当然，我们也不需要真的在发生扭结处把后续所有的自旋都翻转（因为我们做的都是虚拟的操作，肯定越简洁越好），我们只需要标记出是在哪个位置放生了平行和反平行的突变即可。

综上，虽然说 Jordan-Wigner 变换很巧妙地构建了自旋和费米子的映射，但是这个映射的前提在于系统的模式数和粒子数相同，意味着对于哈密顿量有要求，需要是最近邻作用才行。

6.3 费米系统的哈密顿量

将自旋映射到费米子上之后，我们后续针对该费米子体系处理问题，因此算符也都需要使用费米子产生湮灭算符表出。(在解释清楚了费米子和自旋链之间的对应关系之后，我们可以直接把体系抽象成 N 个费米子模式的问题，而不需要再关注原先的自旋链)。

一个关键的结论在于费米子算符所表达的哈密顿量如果是局域的，对于原始的自旋链来说也应该是局域的。典型的哈密顿量为 $H = -t \sum (c_i^\dagger c_{i+1} + c_{i+1}^\dagger c_i) + V \sum n_i n_{i+1}$ ，前一项体现的是两个相邻的格点之间一个费米子的转移，后一项体现的是同时占据两个相邻格点。

需要注意的是一般来说费米子占据总数是守恒的，比如说一般费米子的占据数等于粒子总数，在自旋-费米子映射中，费米子总数等于总自旋向上的粒子数，也是一个守恒量，因此对于哈密顿量是有要求的，不能随意改变占据数。

代入之前得到的 Jordan Wigner 变换， $c_i^\dagger c_{i+1} = \sigma_i^+ (-\sigma_i^z) \sigma_{i+1}^- = \sigma_i^+ \sigma_{i+1}^-$ ，同样是局域作用。

$n_i n_{i+1} = \frac{1}{4}(\sigma_i^z + 1)(\sigma_{i+1}^z + 1)$ ，同样也是局域作用。不难发现，适当的搭配参数 t, V ，即成为了常见的 XY 模型等，是可以解析求解的。

另一种常用的模型称为 Hubbard 模型，其哈密顿量为 $H = -t \sum_{j, \sigma=\uparrow, \downarrow} (c_{j\sigma}^\dagger c_{j+1\sigma} + c_{j+1\sigma}^\dagger c_{j\sigma}) + V \sum n_{j\uparrow} n_{j\downarrow}$ ，与之前的不同之处在于，每个格点处可以有四种状态， $|0\rangle, |\uparrow\rangle, |\downarrow\rangle, |\uparrow\downarrow\rangle$ ，通过合理的映射关系我们同样可以建立费米子和自旋之间的映射，该模型同样可以解析求解。

6.4 自由费米子

6.4.1 自由费米子的解

自由费米子系统指的并不是哈密顿量中无相互作用项，而是只含有二次型式，因此总可以对角化为某种形式的单粒子，从而把体系变成自由的。

$H = \sum c_j^\dagger T_{jl} c_l$ ，令 $c^\dagger = (c_1^\dagger, c_2^\dagger \dots c_N^\dagger)$ ， $c = (c_1, c_2 \dots c_N)^T$ ，进行对角化操作，

$$H = c^\dagger T c = c^\dagger U^\dagger (U T U^\dagger) U c = d^\dagger \epsilon d = \sum \epsilon_m d_m^\dagger d_m,$$

注意本征值可能有负数，将本征值按照从小到大排序，其中有 M 个为负值，

因此基态为： $|\psi\rangle = \prod_{m=1}^M d_m^\dagger |0\rangle$ 。

考虑关联矩阵 $G_{lj} = \langle \psi | c_j^\dagger c_l | \psi \rangle$ ，

$(UGU^\dagger)_{m,m'} = \sum_{j,l} U_{mj} \langle \psi | c_j^\dagger c_l | \psi \rangle U_{lm'}^\dagger = \langle \psi | d_m^\dagger d_{m'} | \psi \rangle$ ，这是一个对角矩阵，

对于基态 $|\psi\rangle = \prod_{m=1}^M d_m^\dagger |0\rangle$ ， $UGU^\dagger = \begin{pmatrix} I_M & 0 \\ 0 & 0 \end{pmatrix}$ ， $\Rightarrow G = U^\dagger \begin{pmatrix} I_M & 0 \\ 0 & 0 \end{pmatrix} U$ 。

现在我们研究自由费米子系统的纠缠情况。研究纠缠的一般程序为将系统分为 A, B 两个子系统，然后对 B 系统求偏迹，从而获得在 A 系统下的约化密度矩阵。通常来说对于一个纯态，对某一子系统得到的约化密度矩阵不再是投影算子，即不是纯态，意味着两个系统之间存在纠缠。我们总是可以通过分析约化密度矩阵以分析系统的纠缠情况。

在此情形中，我们可以使用之前求得的关联矩阵来得到约化密度矩阵。下面来看具体细节，我们主要研究基态的纠缠情况。

取 $l, j \in A, G_{lj}^A = \langle \psi | c_j^\dagger c_l | \psi \rangle = \text{tr}(|\psi\rangle \langle \psi| c_j^\dagger c_l) = \text{tr}_{A,B}(\rho c_j^\dagger c_l) = \text{tr}_A(\text{tr}_B(\rho c_j^\dagger c_l)) = \text{tr}_A(\rho_A c_j^\dagger c_l)$ 。

这里符号上稍微有点歧义， c_j^\dagger, c_l 在子系统 B 上的部分都是 I，因此在最后的表达式中可以把这些 I 去掉，仅视为在子系统 A 上的算符。我们希望由此反解出约化密度矩阵 ρ_A 。

对关联矩阵进行对角化操作， $G = V^\dagger \Lambda V, \Lambda_p \delta_{pp'} = \sum_{l,j} V_{pl} \text{tr}_A(\rho_A c_j^\dagger c_l) V_{jp'}^\dagger = \text{tr}_A(\rho_A f_p^\dagger f_{p'})$ 。

f 所代表的是一组相互正交的模式，每个模式有占据与不占据两种状态，以上我们完成了解耦过程，约化密度矩阵可以写为：

$$\rho_A = \prod (\Lambda_p f_p^\dagger f_p + (1 - \Lambda_p) f_p f_p^\dagger) = \bigotimes \begin{pmatrix} \Lambda_p & 0 \\ 0 & 1 - \Lambda_p \end{pmatrix}。$$

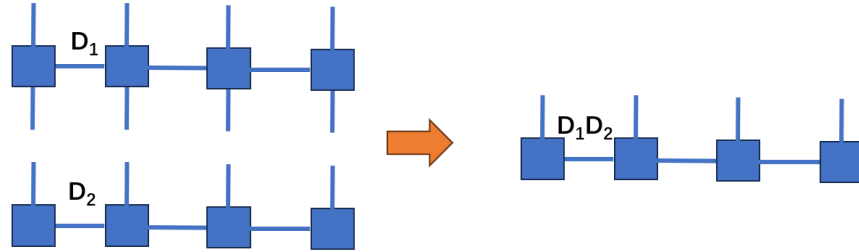
需要注意的是系统的纠缠特征与我们选择何种表象是没有关系的，更换表象只是为了使得密度矩阵的形式更简单以便于求得纠缠特性。在这里，子系统下不同 f 模式之间的耦合不带来纠缠，真正带来纠缠的是总系统的基态在子系统表述下，各个 f 模式究竟是占据还是不占据是不清楚的，只知道一个概率，换言之，如果 $\Lambda_p = 0, 1$ ，则该模式是否占据是确定的，因此不会对总的纠缠带来贡献。同理我们也可以看出纠缠最大的情况时 $\Lambda_p = \frac{1}{2}$ ，容易看出纠缠熵 $S \sim -\text{tr}(\rho_A \ln \rho_A) \sim \ln N$ 。

更具体地，考虑 2^{N_A} 个基态 $|\alpha\rangle = |n_1 n_2 \dots n_{N_A}\rangle = (f_1^\dagger)^{n_1} (f_2^\dagger)^{n_2} \dots (f_{N_A}^\dagger)^{n_{N_A}} |0\rangle$ ，相互正交，我们相当于完成了体系的 Schimidt 分解 $|\psi\rangle = \sum |\alpha\rangle_A \otimes |\alpha\rangle_B$ 。由此我们找到了横向连接的键（虽然重数很大）。下一步自然是要进行分解，我们可以选择使用之前的 SVD 方法进行分解，也可以直接基于求得的约化密度矩阵，由于已经对角化，直接截断掉较小的值（意味着这些模式的贡献很小）。

建立了一个键之后，我们可以不断二分建立更多的键，如果键的重数过大则直接对于约化密度矩阵进行截断即可，理论上来说，通过这种方法我们可以得到基态的 MPS 形式。

6.4.2 MPO-MPS 作用

张量网络处理问题时，通常需要计算 MPO 在 MPS 作用后得到的结果：



如果不进行截断，在计算过程中键的重数会不断上升。截断的方式前面已经讲过，只需要从左到右依次进行 SVD 分解并截取较小的奇异值即可，最终得到的应该是一连串正交矩阵的乘积。

采用这种方式也可以构造出一些态的 MPS 形式，前提是已经得到一些解析计算的结果。以自由费米子情形为例，上一节中我们已经得到了基态的表达式 $|\psi\rangle = \prod_{m=1}^M d_m^\dagger |0\rangle$ 。上一节中给出了通过关联矩阵对角化求解约化密度矩阵，再不断分割子系统得到 Schimidt 分解构造 MPS 的方法，另一种思路是通过 DMRG，利用变分法直接求解基态。这里给出第三种方法，这种方法需要利用到一些数学上的特征和设计。

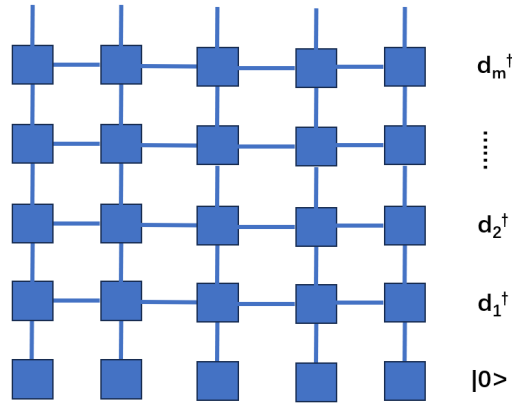
考虑 $d_m^\dagger = \sum_{l=1}^N A_{ml} c_l^\dagger$ ，这种求和形式转化成矩阵乘积我们之前已经得到过，

$$d_m^\dagger = \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} I & 0 \\ A_{m1}c_1^\dagger & I \end{pmatrix} \begin{pmatrix} I & 0 \\ A_{m2}c_2^\dagger & I \end{pmatrix} \cdots \begin{pmatrix} I & 0 \\ A_{mN}c_N^\dagger & I \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

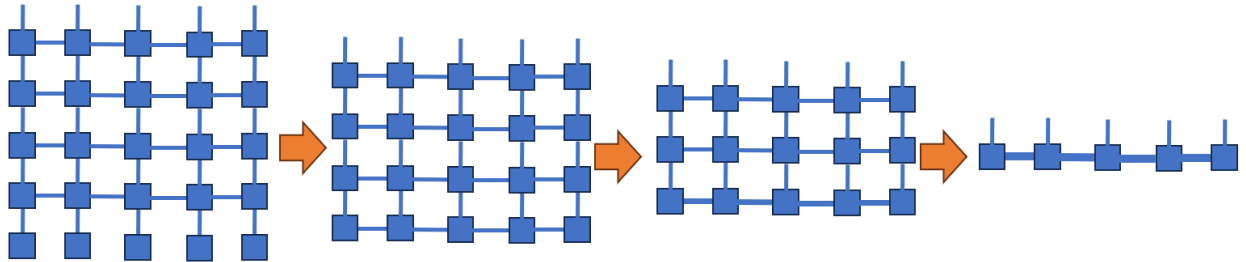
我们希望得到每个矩阵只使用对应点的自旋算符的形式：

$$\begin{aligned} & \begin{pmatrix} I & 0 \\ A_{m1}\sigma_1^+ & -\sigma_1^z \end{pmatrix} \begin{pmatrix} I & 0 \\ A_{m2}\sigma_2^+ & -\sigma_2^z \end{pmatrix} \begin{pmatrix} I & 0 \\ A_{m3}\sigma_3^+ & -\sigma_3^z \end{pmatrix} \\ &= \begin{pmatrix} I & 0 \\ A_{m1}\sigma_1^+ + A_{m2}(-\sigma_1^z)\sigma_2^+ & (-\sigma_1^z)(-\sigma_2^z) \end{pmatrix} \begin{pmatrix} I & 0 \\ A_{m3}\sigma_3^+ & -\sigma_3^z \end{pmatrix} \\ &= \begin{pmatrix} I & 0 \\ A_{m1}\sigma_1^+ + A_{m2}(-\sigma_1^z)\sigma_2^+ + A_{m3}(-\sigma_1^z)(-\sigma_2^z)\sigma_3^+ & (-\sigma_1^z)(-\sigma_2^z)(-\sigma_3^z) \end{pmatrix} \\ & d_m^\dagger = \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} I & 0 \\ A_{m1}\sigma_1^+ & -\sigma_1^z \end{pmatrix} \begin{pmatrix} I & 0 \\ A_{m2}\sigma_2^+ & -\sigma_2^z \end{pmatrix} \cdots \begin{pmatrix} I & 0 \\ A_{mN}\sigma_N^+ & -\sigma_N^z \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \end{aligned}$$

于是基态可以用下图方式得到：



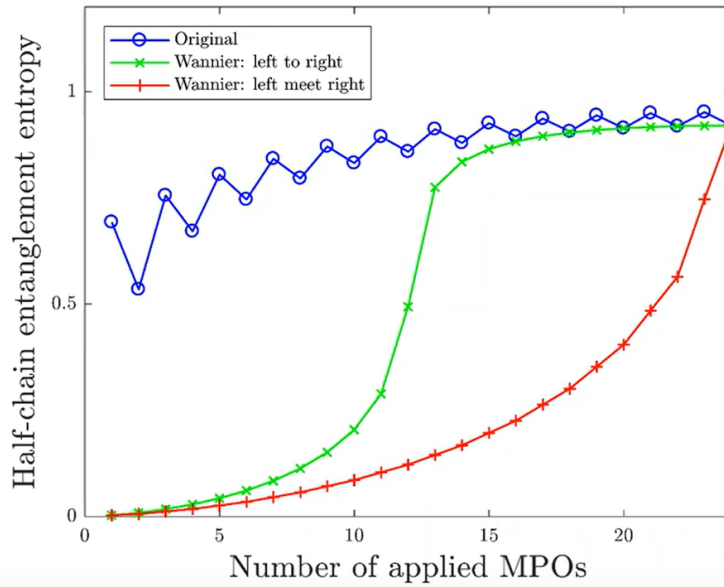
这不过是不断地将 MPO 作用在 MPS 上的过程，从下至上进行收缩，每一次键的重数乘 2，不进行截断的话，收缩到最顶端时键的重数为 2^M 。



值得指出的是，由于 $\{d_m^\dagger, d_{m'}^\dagger\} = 0$ ，因此交换作用顺序得到的 MPS 最多相差负号，因此我们可

以任意调换 MPO 的作用顺序。

在以上过程中，键的重数指数增长，意味着必须要进行截断才能保持计算效率，但是截断必然会引入截断误差，截断误差在收缩过程中不断积累，因此如何缓解截断误差的积累是值得考虑的问题。从物理上去思考某一次 MPO 作用到 MPS 上的截断误差，其来源通常是非局域（键重数本来很大，许多本征值有效）的项被我们强行截断。换句话说，如果中间过程导致的纠缠过多，截断的影响会很大，即每一次截断的过程带来的误差和当前的纠缠程度是相关的。换言之，尽管通过不同的方式进行作用，最终体系都会达到相同的纠缠程度，但是在逐步完成作用的过程中，如果一开始就达到了较多的纠缠状态，后续过程积累的截断误差就会很多；如果在作用过程中纠缠程度是缓慢升高的，则只有最后几次积累的误差会带来影响。



以上图为例，蓝线和绿线较快地达到了高纠缠的状态，而红线达到高纠缠的进程平缓，因此积累的误差最小。在实际进行了截断近似的过程中，利用红线的方法的误差应该更低。

现在我们说明如何进行不同的 MPO 作用：

$|\psi\rangle = \prod_{m=1}^M d_m^\dagger |0\rangle = \prod_{r=1}^M f_r^\dagger |0\rangle$, $f_r^\dagger = \sum_{m=1}^M U_{rm} d_m^\dagger$, 我们总是可以任意组合 d_m^\dagger , 由于费米子的泡利不相容原理，展开右端后必然只会余下 $d_{\sigma(1)}^\dagger d_{\sigma(2)}^\dagger \dots d_{\sigma(M)}^\dagger |0\rangle$ 项，

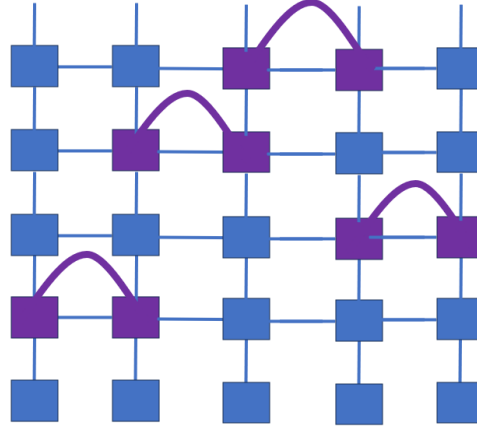
$$\begin{aligned} \text{具体地, } d_1^\dagger d_2^\dagger \dots d_M^\dagger &= \sum_{\sigma} U_{1\sigma(1)} U_{2\sigma(2)} \dots U_{M\sigma(M)} d_{\sigma(1)}^\dagger d_{\sigma(2)}^\dagger \dots d_{\sigma(M)}^\dagger \\ &= \sum_{\sigma} (-1)^{\text{sgn}(\sigma)} U_{1\sigma(1)} U_{2\sigma(2)} \dots U_{M\sigma(M)} d_1^\dagger d_2^\dagger \dots d_M^\dagger = \det(U) d_1^\dagger d_2^\dagger \dots d_M^\dagger, \end{aligned}$$

因此只需要 U 是一个酉矩阵即可满足条件。

直接使用 d_m^\dagger , 避免不了其由各个 c_i^\dagger 线性组合构成，通常不是局域作用，我们希望得到一组 f_r^\dagger , 尽可能满足局域化条件。由于我们能够得到的新轨道总是在 d_m^\dagger 构成的子空间中，必然无法是完全局域的 (c_i^\dagger)，因此我们的目标只能是找到 d_m^\dagger 子空间中最局域的一组轨道。

想要找到局域轨道，自然想到位置算符 $\sum l c_l^\dagger c_l$ (前面的 1 为了区别不同位置的费米子) 的本征态就

是各个费米子单独占据的态。当然由于我们只能在子空间内求解，必然无法做到这么局域，但是我们有理由相信在子空间内求解得到的轨道，应该是非常接近局域的。具体的，定义矩阵 $\tilde{X}_{mn} = \langle 0 | d_n X d_m^\dagger | 0 \rangle$ ，将其对角化， $\Lambda_X = U \tilde{X} U^\dagger$ ，此处的 U 即为之前待求的 U 。这样找到的轨道称之为 Wannier 轨道。



此时再进行 MPO-MPS 作用，每一次作用的 f_r^\dagger 算符都是局域的，其 MPO 展开的很多部分都是之间相连的键而没有多少信息（节点处系数矩阵是 I ），态的纠缠程度上升平缓。由此我们可以理解上面那张图里曲线的走势。需要说明的是图中给出了从左到右与一左一右两种不同的作用方式，我们发现从左到右作用时纠缠熵在中间位置有一个显著的上升，这其实也是好理解的，因为我们纠缠熵的测量总是从中间位置处，测量两端的纠缠情况。一旦靠近中点处的 MPO 作用后，左右两侧之间的纠缠会显著上升，而对于一左一右的情况，纠缠一开始都只在左右两个子系统内部，因此总的纠缠熵上升缓慢。

7 含时演化

我们期待解决含时演化问题， $|\psi(t)\rangle = e^{-iHt} |\psi\rangle$, $|\psi(\tau)\rangle = e^{-\tau H} |\psi\rangle$ ，后一式称为虚时间演化，应用范围更加广泛，比如说可以代表热算符 $e^{-\beta H}$ 。容易看出对于虚含时演化，当 $\tau \rightarrow \infty$ 时，得到的实际上是基态（非简并），把初始态展开到本征态上即证明其他项的贡献是指数衰减的。

一般而言解决含时演化问题，我们需要求解本征态，然后将初态分解到各个本征态上，再带上不同的相位再求和。这个过程非常麻烦。但是如果我们可以把算符 $e^{-iHt}, e^{-\tau H}$ 直接写成 MPO 形式，对于任意简单的 MPS 形式初态，我们只需要做收缩操作即可。本节的目的在于寻找把 e 指数的哈密顿量算符 $e^{-iHt}, e^{-\tau H}$ 写成 MPO 形式的方法。

7.1 一维自旋系统

$$H = \sum h_i h_{i+1} = (h_{12} + h_{34} + \dots) + (h_{23} + h_{45} + \dots) = H_{\text{odd}} + H_{\text{even}},$$

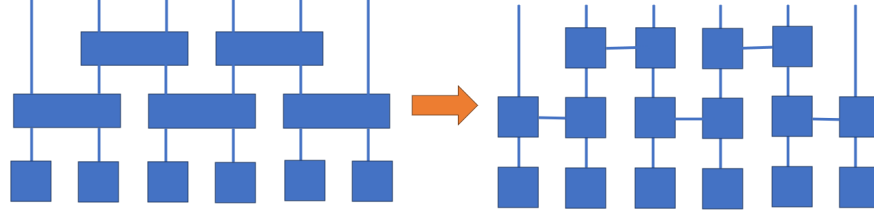
因此 $e^{-H\tau} = e^{-H_{\text{odd}}\tau - H_{\text{even}}\tau}$ ，需要注意的是该式子无法直接拆开，

由 BH 公式，仅当 $[H_{\text{odd}}, H_{\text{even}}] = 0$ 时， $e^{-H\tau} = e^{-H_{\text{odd}}\tau} e^{-H_{\text{even}}\tau}$ 。

7.1.1 对易哈密顿量

$[H_{odd}, H_{even}] = 0$, 则有 $\forall j, [h_{j,j+1}, h_{j+1,j+2}] = 0$, 这通常是无法满足的, 不过我们先考虑简单情形, 此时 $e^{-H_{odd}\tau} = e^{-h_{12}\tau} e^{-h_{34}\tau} e^{-h_{56}\tau} \dots, e^{-H_{even}\tau} = e^{-h_{23}\tau} e^{-h_{45}\tau} e^{-h_{67}\tau} \dots$,

用图形化的方式计算, 我们很容易得到 MPO 作用于 MPS 的形式:



该过程只需对每个二元块进行一次 SVD 分解, 然后把中间的对角阵吸收进入某一侧即可构成右侧 MPO 形式。

7.1.2 Trotter-Suzuki 分解

一般而言, 哈密顿量并不满足如上对易关系, 我们没有办法直接通过拆分的方式构造 MPO。

可以进行如下操作, $\delta\tau = \frac{\tau}{N}, e^{-H_{odd}\tau - H_{even}\tau} = (e^{-H_{odd}\delta\tau - H_{even}\delta\tau})^N$,

由 BH 公式, 强行拆分的误差项应该正比于两个指数的对易子,

$e^{-H_{odd}\delta\tau - H_{even}\delta\tau} = e^{-H_{odd}\delta\tau} e^{-H_{even}\delta\tau} + O(\delta\tau^2)$, 误差是二阶小量,

因此 $e^{-H\tau} = \lim_{N \rightarrow \infty} (e^{-iH_{odd}\delta\tau} e^{-iH_{even}\delta\tau})^N$ 。

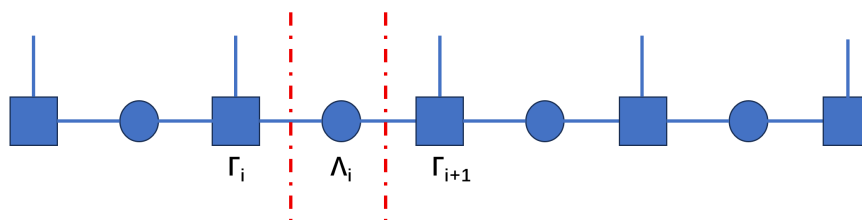
需要说明的是, 对于没有能隙的系统, 只有严格趋于无穷次作用才能得到正确的基态, 如果强行截断的话误差较明显; 但是对于有能隙的系统, 基态和激发态之间有明显的能量差, 通过较多次的作用总是可以认为得到了足够好的基态, 具体地, 只要每一步能捕获的能量变化小于能隙, 则说明我们得到了充分好的基态。

7.2 时间演化块消减算法 Time-evolving Block Decimation TEBD

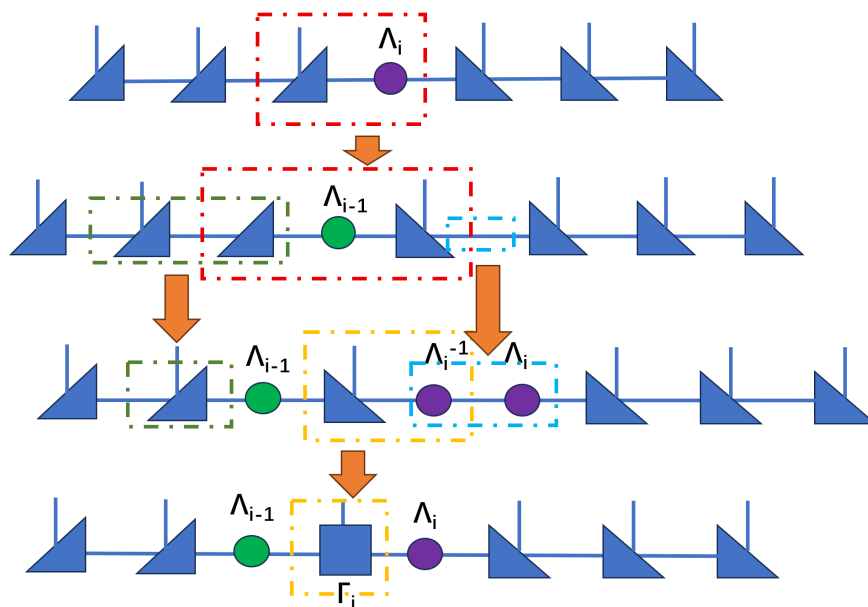
前一节中我们已经有了解决时间演化问题的思路, 对于不对易的哈密顿量, 使用 Trotter-Suzuki 分解, 只要作用足够多次, 总能够使得最初的 MPS 达到想要的状态。作用的过程不过是在不断在张量网络上进行收缩的过程, 现在我们具体思考应该使用何种形式进行维护每一次收缩之后的 MPS。

7.2.1 $\Gamma - \Lambda$ 形式 MPS

我们希望得到如下形式的 MPS, 每一个圆圈代表对角矩阵, 任意一个圆圈的两侧都是正交的, 即我们从任意位置处都能够得到量子态的 Schmidt 分解:



这种 $\Gamma - \Lambda$ 形式 MPS 实际可以从中心正交形式的 MPS 得到：



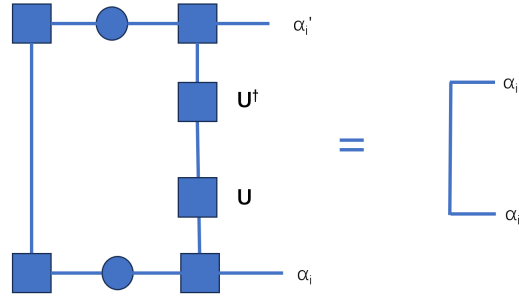
该流程图的含义为相同颜色的虚线框视为同一个整体，进行收缩或是 SVD 分解操作。

我们检查一下引入的绿色和紫色圆圈两侧是否都是正交矩阵。对于绿色圆圈，从第二步可以看出其左右两侧都是正交矩阵，因此后续步骤仍然保持了这一点；对于紫色圆圈，我们在第一步中就已经知道了其左右两侧都为正交矩阵，因此尽管最后我们得到的紫色矩阵是后插入的，但是其左右的总的结构应该与第一步时完全相同，都是正交的。

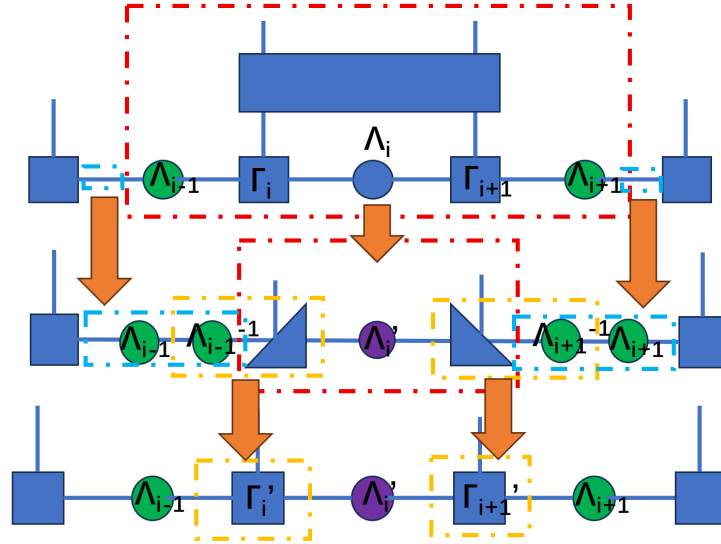
通过这种方式我们可以构造得到 $\Gamma - \Lambda$ 形式 MPS, 需要注意的是我们需要在作用 MPO 的过程中始终维持 $\Gamma - \Lambda$ 形式。

这种形式的 MPS 可以用于利用经典计算机模拟量子线路。量子线路中我们常用的操作可以分为单比特操作和双比特操作，并且这些操作都是酉操作。下面具体来看在 $\Gamma - \Lambda$ 形式 MPS 上执行这两种操作的过程。

单比特操作是简单的，只要执行收缩即可自动满足正交条件：



对于双比特操作，需要按照如下方式更新 $\Gamma - \Lambda$ 形式 MPS:



该流程图的含义为相同颜色的虚线框视为同一个整体，进行收缩或是 SVD 分解操作。

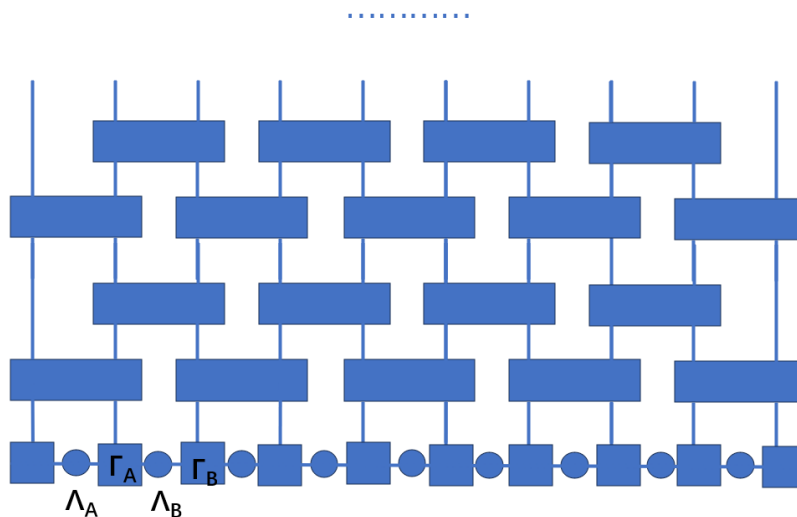
按照如上图顺序更新 $\Lambda_i, \Gamma_i, \Gamma_{i+1}$ ，需要注意的是这个过程中有键的增加，中间的 Γ_i' 的重数从 $D \rightarrow dD$ ，因此我们有必要在进行 SVD 时引入截断。

同时我们检查是否保持 $\Gamma - \Lambda$ 形式的性质，即从任意圆圈处割开均可以进行量子态的 Schmidt 分解。绿色圆圈保持不变，且其一侧完全没变，因此绿色圆圈的两端总是保持正交性质。紫色圆圈的两端相当于在正交矩阵的基础上乘上 SVD 分解得到的正交矩阵，仍然为正交矩阵（注意填入的两个绿色圆圈实际上是 I ，没有贡献），因此同样保持住了正交形式。从中我们也可以看出这么操作的巧妙之处，如果第一次分割不包含绿色圆圈，则无法保持 $\Gamma - \Lambda$ 形式正交特性。

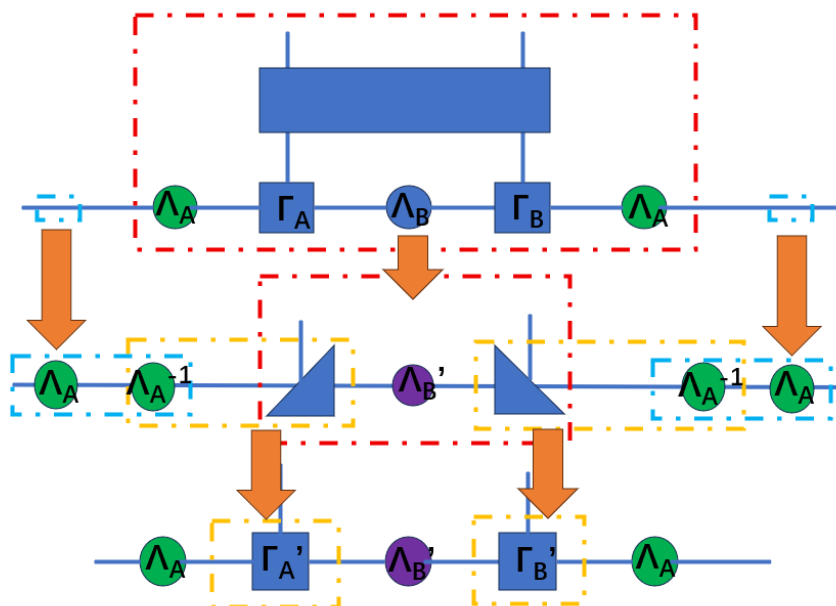
在 $\Gamma - \Lambda$ 形式 MPS 上维护量子线路是简洁而高效的。注意在单比特门作用下键重数不变，但是双比特门作用下键的重数会上升，这从物理上是好理解的，因为双比特门带来了纠缠而单比特门没有。同时为了保持 MPS 形式的高效性，键重数需要进行合理的截断，也就是说我们依旧只能维护低纠缠状态，这是限制一切张量网络方法的核心。

7.3 无限长链演化 iTEBD

$\Gamma - \Lambda$ 形式 MPS 的一个很大的好处在于具有平移不变性，不同于之前两点形式的 DMRG，那时我们寻求的是逐点优化，这种具有平移不变性的形式非常利于我们研究热力学极限下的情况。

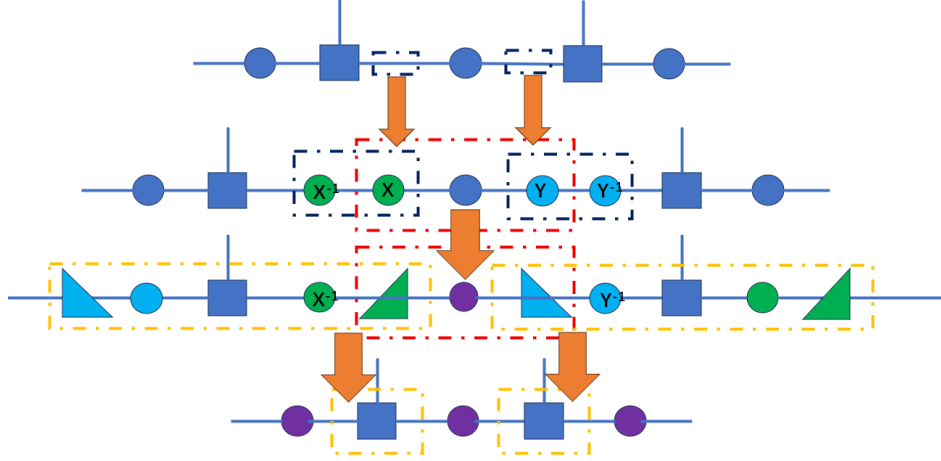


考虑使用 Trotter-Suzuki 分解，只要作用的次数足够使得 MPS 收敛，我们就可以说求得了基态。现在看具体维护过程，对于热力学极限，即无限长链的情况（仍然考虑哈密顿量只有最近邻作用），系统具有平移不变性，使用 $\Gamma - \Lambda$ 形式 MPS 我们实际上只需要更新 4 个矩阵参数，其核心更新过程如下所示：

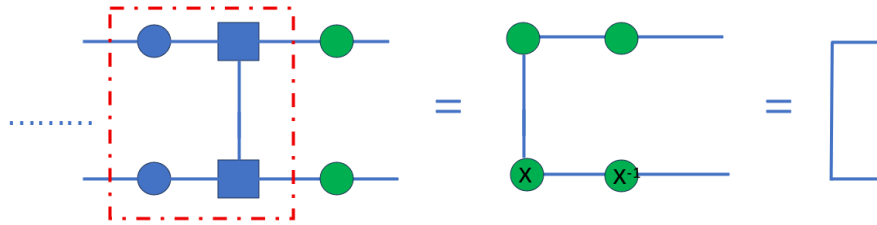


每一次更新过程中只进行了 $\{\Gamma_A, \Gamma_B, \Lambda_A, \Lambda_B\} \rightarrow \{\Gamma'_A, \Gamma'_B, \Lambda_A, \Lambda'_B\}$ 的改变，注意每一层的作用要滑动错位一下，即每一次 A 和 B 的地位要互换一下。这种求解基态的方式极其简洁，只需进行非常局部（4 个矩阵）的更新。

以上演化方式实际上存在一定风险。对于酉演化，演化过程原则上不会破坏 $\Gamma - \Lambda$ 形式的正交性，但是由于我们引入了截断，因此截断误差的积累很可能导致正交性被破坏；对于非酉演化，比如之前提到的虚时间演化，正交性天然无法满足。因此我们需要某种技术调整当前的 $\{\Gamma_A, \Gamma_B, \Lambda_A, \Lambda_B\}$ ，使其满足正交形式。



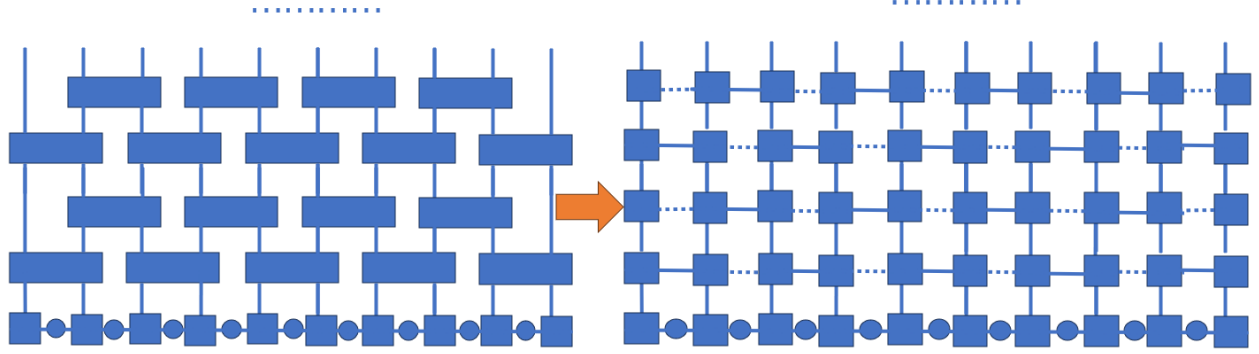
基本想法是插入特定的 $X, X^{-1}; Y, Y^{-1}$ ，使得改进后的 Γ', Λ' 满足正交性。对于 $X, X^{-1}; Y, Y^{-1}$ 有如下要求：



基本想法是找到由初始的 Γ, Λ 构成的转移矩阵 ($D^2 \times D^2$)，其一边由 bra, ket 指标共同组成。然后求解该转移矩阵的贡献最大的特征值（一个长度为 D^2 ）的向量，再将其写成 $D \times D$ 的矩阵形式，把这个矩阵拆分成 XX^\dagger 的形式，由此计算得到我们所需的 X ，对于 Y 同理。通过这种构造可得，上图第二行中绿色 X 左侧部分均满足正交条件，因此再乘上后续 SVD 分解的正交矩阵依然是正交的。

7.4 时间密度矩阵重整化群 t-DMRG

考虑有限长的系统，没有了平移不变性，问题变得复杂了许多。为了方便起见，我们将时间演化过程写成 MPO-MPS 作用形式：

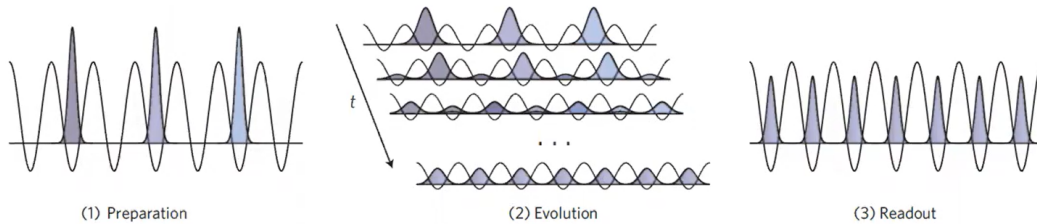


其中虚线代表强行定义的维度为 1 的键。看起来我们好像可以直接做 MPO-MPS 作用进行收缩，但实际上这是有问题的。具体来说，我们在进行收缩时需要从左到右重新维护 MPS 形式，即不断进行 SVD 分解，这时键的重数是指数增长的，进行截断的合理性（或者说保真度）依赖于当前步骤的 MPO 作用是局域的。但是显然这里每一个 MPO 都是两两之间的作用，是非常不局域的。因此直接进行收缩和截断的误差会很大。

考虑类似 DMRG 的方法得到 MPO-MPS 作用，即我们先设一个重数已经固定好的 MPS，称为 $|\psi_D\rangle$ ，然后不断去优化它。优化的标准是我们想要让设定的 MPS 和 MPO-MPS 作用之后的态 $|\psi'\rangle = W|\psi\rangle$ 尽可能相近。即 $\max(\langle\psi_D|\psi'\rangle\langle\psi'|\psi_D\rangle)$ 。类似于 DMRG 问题，总是可以定义有效哈密顿量 $H_{eff} = |\psi'\rangle\langle\psi'|$ ，其最大本征值对应的本征态即为所求的 $|\psi_D\rangle$ 。

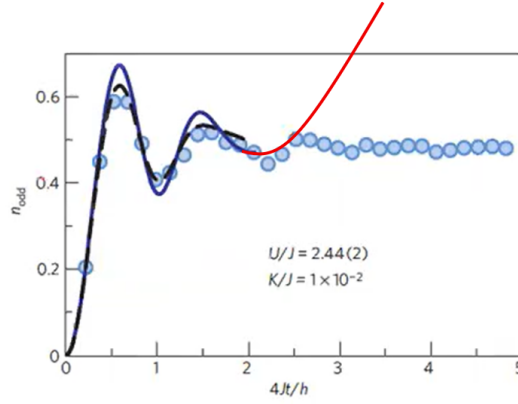
对于以上方法的有效性（保真度），需要依赖于物理规律。根据之前的理论，键重数 $D \sim e^S$ ，但是一般而言，系统的纠缠熵 $S \sim \alpha t$ ，这意味着长时间的演化必然导致有限重数的 MPS 无法描述。换言之，张量网络的方法只适用于短时间的演化过程。

从具体的例子上我们可以更清楚地认识到这一点。（这是德国慕尼黑某个组的工作）。考虑玻色系统在如下哈密顿量的演化，初态只在偶数格子处占据，偶数格子全部空出。我们可以实验测量奇数格子的占据总数，并和利用前述 t-DMRG 方法得到的模拟结果进行比较。



$$\hat{H} = \sum_j \left[-J (\hat{a}_j^\dagger \hat{a}_{j+1} + \text{h.c.}) + \frac{U}{2} \hat{n}_j (\hat{n}_j - 1) + \frac{K}{2} \hat{n}_j^2 \right]$$

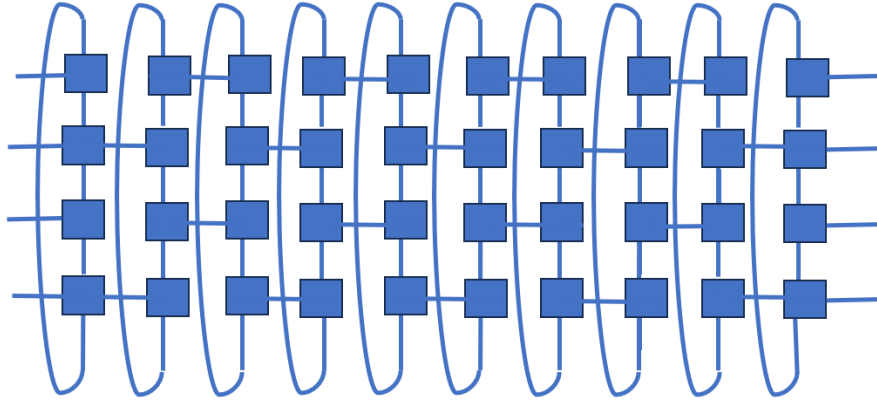
实验结果和理论结果的对比如下。圆点代表实验值，实线代表 t-DMRG 方法模拟值。原图在某处截断是因为后续就发散了，即 t-DMRG 方法失效，这充分说明了该方法只能用于短时间的演化。



人们也寻找过其他方法试图解决该问题。比如说考虑海森堡绘景，试图把算符进行演化并写成 MPO 形式，不过最后的结果依然不成功。个人认为，如果系统的纠缠熵的确随时间线性增加的话，似乎从原则上来说就不可能通过张量网络的方法进行很好的模拟。再精巧的设计只是更加充分地利用了现有的空间去表达一个纠缠程度更高的态，只能做到优化能模拟的时间长度这一点，并不能够从根本上实现任意长时间的态演化。

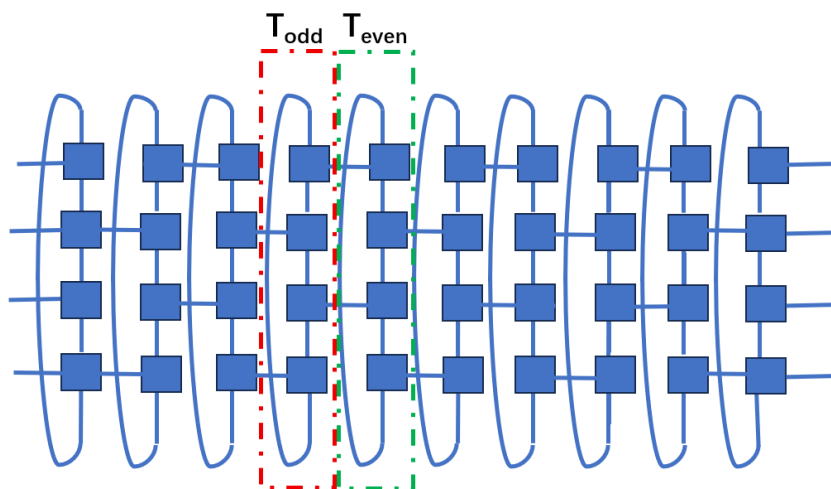
7.5 虚时间演化

虚时间演化的一个重要作用在于计算配分函数 $Z = \text{tr}(e^{-\beta H}) = \sum \langle s_1 s_2 \dots s_N | e^{-\beta H} | s_1 s_2 \dots s_N \rangle$ ，我们同样可以将这个配分函数看作是 MPO-MPS 作用的结果。我们已经将哈密顿量展开成了矩阵乘积形式：



我们自然可以选择从下至上收缩的形式计算该 MPO 的迹，这里给出另一种视角，可以更加巧妙而简洁地解决这一问题。

竖着来看，把每一列当作一个 MPS，对于热力学极限情形 ($N \rightarrow \infty$ ，有无穷多列)，实际上只有两种类型的 MPS，用转移矩阵表示，因此配分函数的表达式可以改写为： $Z = \text{tr}((T_{\text{odd}} T_{\text{even}})^{\frac{N}{2}}) = \lambda_{\text{max}}^{\frac{N}{2}}$ 。



需要注意的是，此时的转移矩阵 T 并不一定是酉矩阵，且其形式可能很复杂，因为纵向列的长度取决于演化的时间长度。

8 对称性

前面我们已经介绍了张量网络的基本方法和具体的使用场景，我们了解到对于一个 MPS 而言，如何使用有限的键重数尽可能相似地表示一个量子态将是算法优劣的核心影响因素。本节我们的目的就是先从理论的角度出发，研究如何充分地利用这些键重数。

8.1 MPS 形式中的对称性

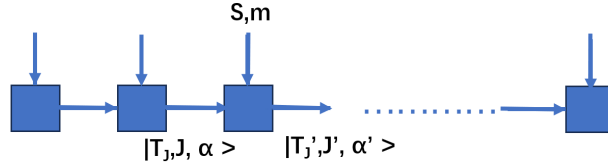
多体系统总是满足一定的对称性。比如海森堡模型 $H = \sum \mathbf{S}_i \cdot \mathbf{S}_{i+1}$ ，满足 $SU(2)$ 对称性， $[H, \sum S_i^a] = 0, (a = x, y, z)$ 。这告诉我们沿着每个方向的总自旋是守恒的，亦即我们在构造或者说演化 MPS 时需要始终满足这一条件。

我们思考如何做到这一点。最直接的方式就是让这些守恒量（总自旋， z 方向的总自旋）成为量子数，这样就避免了把不同总自旋的态混在一起的可能。事实上对于自旋耦合问题，我们正是这么做的。以自旋 $\frac{1}{2}$ 的耦合为例：

$$\frac{1}{2} \otimes \frac{1}{2} = 0 \oplus 1;$$

$$\frac{1}{2} \otimes \frac{1}{2} \otimes \frac{1}{2} = (0 \oplus 1) \otimes \frac{1}{2} = \frac{1}{2} \oplus \frac{1}{2} \oplus \frac{3}{2};$$

从中我们发现简并的情况出现，意味着仅适用总自旋和总 z 方向自旋无法完整表述一个量子态的来源，因此我们还需要另外的量子数，不难看出这个量子数是为了区分相同的总自旋 J ，因此我们将其称为 T_J 。有了这三个量子数，我们可以描述矩阵乘积态：



于是我们维护的矩阵（张量）可以写成：

$$A_{(T_{J'}, J', \alpha'); (T_J, J, \alpha)}^{S, m} = \langle T_{J'}, J', \alpha' | T_J, J, \alpha \rangle | S, m \rangle = T_{(T_{J'}, J'); (T_J, J)} \langle J', \alpha' | J, \alpha, S, m \rangle。$$

后一部分实际上是 CG 系数，是固定不变的，前一部分是结构张量，是可变的。换言之，体系的对称性由后一部分决定，而前一部分 T 表示的是能进行变分的参数，即体系真正的自由度所在。

自旋守恒体现的是 $SU(2)$ 对称性，我们还可以关注其他对称性。

粒子总数守恒，此时系统满足 $SU(1)$ 对称性，此时 $A_{(t_m, m); (t'_{m'}, m')}^n = T_{t_m, t_{m'}} \delta_{m+n, m'}$ ；

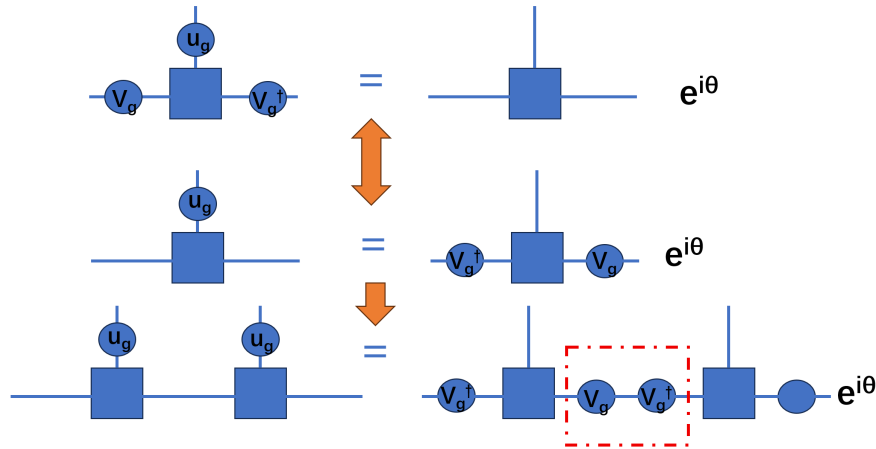
费米子宇称守恒（费米子数的奇偶性不变），此时系统满足 Z_2 对称性，此时 $A_{(t_m, m); (t'_{m'}, m')}^n = T_{t_m, t_{m'}} \delta_{m+n, m'} (mod\ 2)。$

$$\text{具体来说，此时 } A^0 = \begin{pmatrix} 0 & 1 \\ T_1 & 0 \\ 1 & 0 & T_2 \end{pmatrix}, A^1 = \begin{pmatrix} 0 & 1 \\ 0 & T_3 \\ T_4 & 0 \end{pmatrix}，\text{即对称性决定了 } A \text{ 矩阵只在某些区域非零。}$$

在

以上分析比较抽象，可以当作是对于系统对称性的一种理解。下面介绍一个实际有用的结论，以便于之后利用对称性分析问题。

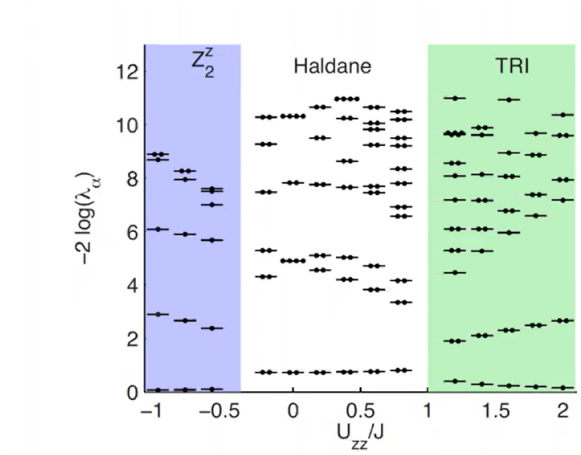
对于满足平移不变性的体系，其对称性可以表述为 $U_g |\psi\rangle = e^{i\theta_g} |\psi\rangle, U_g = \otimes u_g$ ，MPS 满足该对称性完全等价于：



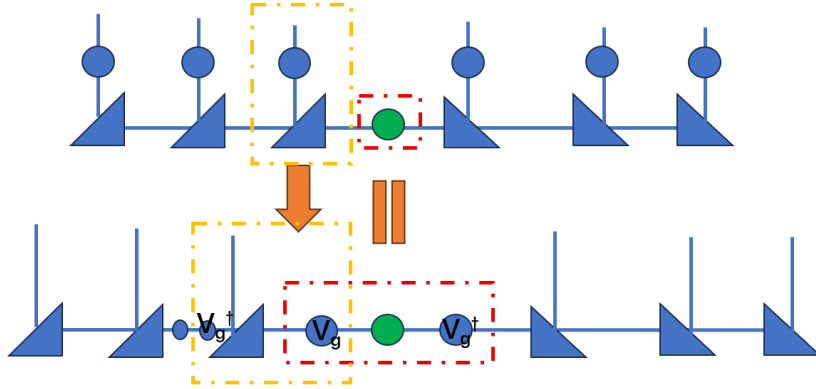
从上图最后一行可以理解其该结论的必要性，对充分性的证明还需更严谨的数学论证，这里暂且不做展开，我们只需要知道该结论完全正确即可。需要注意的是图中的等号右侧总是可以引入相位，这个相位对于后续的分析是重要的。

8.2 对称性保护拓扑相

了解了系统对称性之后的作用在于我们可以对于系统本征态（主要是基态）的性质做出一些解释和判断。以哈密顿量 $H = J \sum \mathbf{S}_j \cdot \mathbf{S}_{j+1} + U_{zz} \sum (S_j^z)^2$ 为例。容易看出当 $\frac{U_{zz}}{J} \gg 1$ 时，由后一项主导，容易看出后一项的基态显然是 $|00 \dots 0\rangle$ ，并且是一个单重态（非简并）。更多的本征态分布情况如下图所示：



容易看出在 $\frac{U_{zz}}{J} = 1$ 处应该发生了一个相变，Haldane 相的基态总是二重简并的（图上最低能级上两个小圆点），我们试图用体系对称性来解释这个现象。



把 MPS 写成中心正交形式，要想满足对称性，可以利用上一节最后给出的结论。从图中不难看出必须要使得 V_g 满足： $V_g \Lambda V_g^\dagger = \Lambda \Leftrightarrow [V_g, \Lambda] = 0$ 。注意这里的 V_g 总是可以引入任意的相位。

当然以上条件并不是 V_g 需要满足的唯一条件。用群论的语言来说，对任意 $u_g \in G$ ，上述条件均应该得到满足。考虑群元素的乘法条件，则容易看出 $V_g V_h = e^{i\phi(g,h,gh)} V_{gh}$ 。注意上节最后得到的等价关系使得我们最后引入了一个相位因子。

现在我们看该哈密顿量，容易看出其具有 $Z_2 \times Z_2$ 对称性，即系统沿着 Z 或者 X 方向旋转 180° 不变。因此我们可以得到如下条件： $[V_X, \Lambda] = [V_Z, \Lambda] = 0, V_X V_Z = \pm V_Z V_X$ 。注意最后不确定的正负号

实则体现的正是以上第二个关系的相位因子。

事实上，以上条件中的正负号可以看做是体系的一种序参量，当取正号时，量子态无简并；取负号时，量子态存在简并。接下来具体说明这一点：

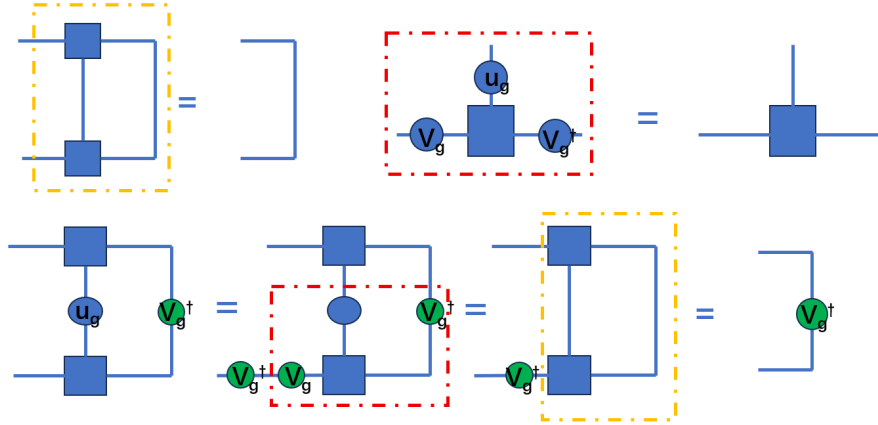
$$[V_Z, \Lambda] = 0 \Rightarrow \begin{cases} \Lambda |\psi\rangle = \Lambda_\alpha |\psi\rangle \\ V_Z |\psi\rangle = e^{i\theta} |\psi\rangle \end{cases},$$

$[V_X, \Lambda] = 0 \Rightarrow \Lambda(V_X |\psi\rangle) = V_X \Lambda |\psi\rangle = \Lambda_\alpha (V_X |\psi\rangle)$ ，即 $V_X |\psi\rangle$ 也是本征值为 Λ_α 的本征态。

如果 $\langle \psi | V_X | \psi \rangle = 0$ ，即可说明体系存在简并。

插入 $V_Z^\dagger V_Z$ ， $\langle \psi | V_X | \psi \rangle = \langle \psi | V_Z^\dagger V_Z V_X | \psi \rangle = \pm (\langle \psi | V_Z^\dagger V_X (V_Z |\psi\rangle)) = \pm \langle \psi | V_X | \psi \rangle$ ，于是取负号时体系存在简并。

为了验证究竟以上关系中序参量究竟是正还是负，我们需要求得 V_g ，可以通过如下方式求解：

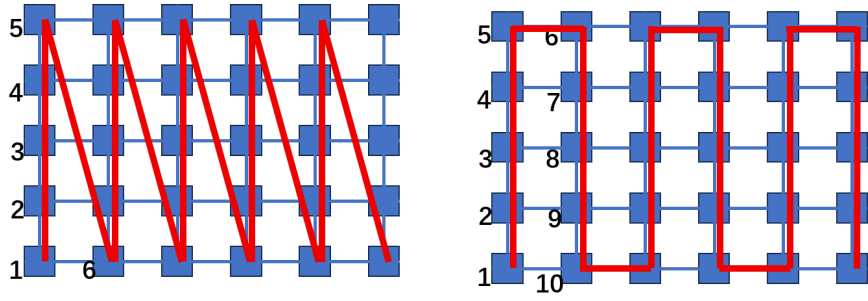


利用正交性条件 and 对称性条件，我们可以得到所求的 V_g^\dagger ，写成向量形式的话，即为插入算符 u_g 后的转移矩阵的最大特征向量。综上，我们可以检查得到对于 Haldane 相，即 $0 < \frac{U_{zz}}{J} < 1$ ， $V_Z V_X = -V_X V_Z$ ；当 $\frac{U_{zz}}{J} > 1$ ， $V_Z V_X = V_X V_Z$ 。

9 投影纠缠对态 Projected Entangled Pair State PEPS

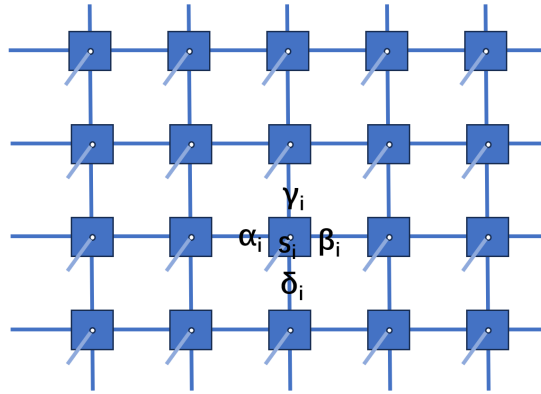
9.1 二维系统

前面所讲的 MPS，MPO 都是针对于一维体系的，如果想求解二维体系，我们必须想办法把该二维体系映射到某个一维的链上去。这么做会导致一些问题，使得 MPS 的一些良好性质被破坏。首先，体系的局域作用会被映射成长程作用，比如说下图中 1,6 号之间或者 1,10 号之间的作用：



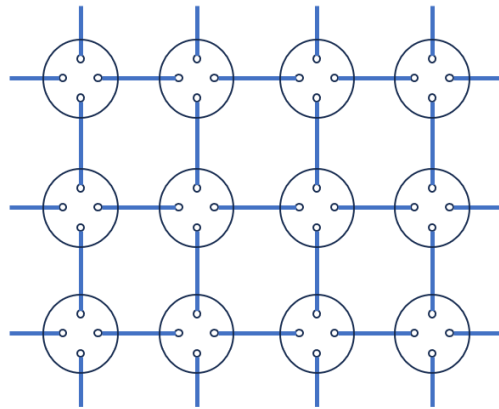
引入了长程作用是第一个弊端，另一个更严重的问题在于面积律。由面积律可知，该二维系统的熵 $S \sim L_y$ ，于是我们的键重数随着 y 方向是指数增加的 $D \sim e^S \sim e^{L_y}$ 。这意味着我们只能够处理 L_y 很小 (< 10) 的情形，并不能很好地解决问题。

于是我们可以转换思路，仿照 MPS，构造相应的结构，即 PEPS:



量子态可以表达为 $|\psi\rangle = \sum_s \sum_{\{\alpha, \beta, \gamma, \delta\}} A_{\alpha_i \beta_i \gamma_i \delta_i}^{s_i} |s_i\rangle$ 。

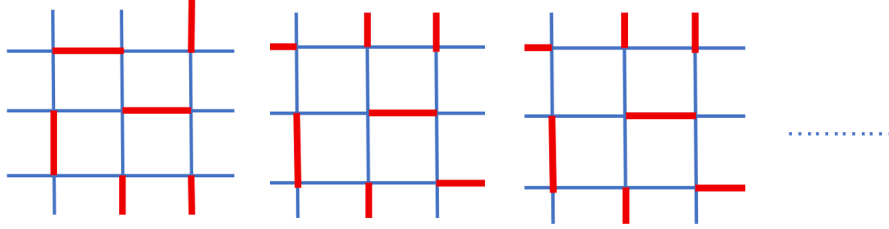
具体如何产生这些键呢？我们可以仿照 AKLT 模型，给出类似的二维 AKLT 模型：



把一个自旋 2 拆分成 4 个自旋 $\frac{1}{2}$, 并且要求用键相连的自旋 $\frac{1}{2}$ 反平行。可以规定投影算符 $P = \sum_{s_i} \sum_{\alpha_i \beta_i \gamma_i \delta_i} B_{\alpha_i \beta_i \gamma_i \delta_i}^{s_i} |s_i\rangle \langle \alpha_i \beta_i \gamma_i \delta_i|$, $I = \sum_{\delta \gamma \text{ or } \alpha \beta} R_{\delta \gamma \text{ or } \alpha \beta} |\delta \gamma \text{ or } \alpha \beta\rangle$, $\Rightarrow |\psi\rangle = (\otimes P)(\otimes I)$ 。

具体而言, 无约束的 $2 \otimes 2 = 0 \oplus 1 \oplus 2 \oplus 3 \oplus 4$, 而 AKLT 模型下, 由于两个自旋 $\frac{1}{2}$ 反平行, 因此总自旋为 4 的态不存在, 于是总空间为 $0 \oplus 1 \oplus 2 \oplus 3$ 。

利用这种引入虚拟点, 并在虚拟点上给定约束条件再投影到真实物理空间上的方式, 我们可以得到更为复杂的态。比如说下图所示的 RVB 状态:



其含义为每个格点总是被一条红边作用, 每条红边表示连接的两端处于单重态 $|01\rangle - |10\rangle$ 。整个态是所有任意分布的红边对应的态的和, 可以想见表达这样一个态是非常复杂的, 但是我们使用如上投影技巧可以构造得到。

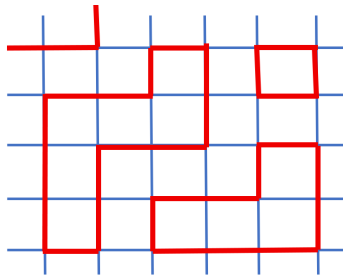
具体地, 每个点连接 4 条边, 可以引入四个虚拟点。虚拟点具有三种状态, $|0\rangle, |1\rangle, |2\rangle$, 蓝边对于状态 $|2\rangle$, 红边对应状态 $|0\rangle, |1\rangle$, 也对应了真实的物理状态 $|0\rangle, |1\rangle$ 。由此可以看出投影算符具有如下形式, 满足约束: 三蓝一红和红边对应状态与真实物理状态对应。

$$\Rightarrow P = |0\rangle (\langle 0222| + \langle 2022| + \langle 2202| + \langle 2220|) + |1\rangle (\langle 1222| + \langle 2122| + \langle 2212| + \langle 2221|)$$

此时描述边的约束。如果一个虚拟点为 $|2\rangle$, 则与之相连的虚拟点必然为 $|2\rangle$; 如果一个虚拟点为 $|0\rangle, |1\rangle$, 则与之相连的虚拟点必然处在相反状态。于是有 $I = |01\rangle - |10\rangle + |22\rangle$ 。

从而 $|\psi\rangle_{RVB} = (\otimes P)(\otimes I)$ 。可以由 PEPS 表述该状态。

另一个例子是如下所示的 RAL 状态, 其由各种不同构型的 AKLT 环叠加而成, 即每个点必须在一个满足 AKLT 条件的环上:



同样可以按照上述方法构造 PEPS 形式。具体而言, 每个点连接 4 条边对应 4 个虚拟节点, 应该保持两红两蓝。其中的两红和物理格点的映射即为三重态的映射, 物理格点的自旋为 1, 红边与蓝边满足的两端的关系与如上情形类似, 因此有:

$$P = |1\rangle (\langle 1122| + \langle 1212| + \dots) + |-1\rangle (\langle 0022| + \langle 0202| + \dots) + \frac{|0\rangle}{\sqrt{2}} (\langle 0122| + \langle 1022| + \dots),$$

$I = |01\rangle - |10\rangle + |22\rangle, \Rightarrow |\psi\rangle_{RAL} = (\otimes P)(\otimes I)$ 。可以由 PEPS 表述该状态。

不过尽管我们给出了 PEPS，并不代表其能够如 MPS 那样好用。其原因在于在 MPS 视角下的矩阵在这里全部变成了四阶张量，导致原先我们可以对矩阵进行的操作（SVD 分解，转移矩阵乘法）都变得很困难。

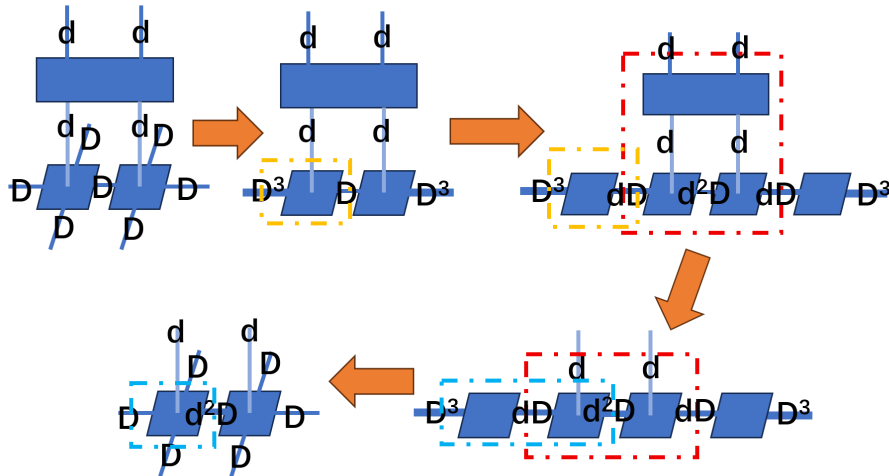
9.2 PEPS 的优化

张量网络方法的核心目标在于求解两个问题。其一是使用变分法求给定哈密顿量的基态，核心是优化问题；其二是计算可观测量，核心是收缩问题。本节研究如何对于 PEPS 进行这两个问题的求解。

本节先研究优化问题，即求解某个已知哈密顿量的基态。之前解决这个问题我们有两种主流思路，其一是用 DMRG，具体而言是把哈密顿量写成 MPO 形式后逐点优化张量，每次相当于求解一个等效哈密顿量的主导本征态；其二是无穷长时间的虚时间演化算法，每次计算小的时间间隔下对应的演化过程，相当于一次 MPO 作用于 MPS。把这两种方法前移到二维情形，显然对于逐点优化来说工作量的增加是巨大的，因为要优化的张量数量和大小都变得很大；而对于第二种方法而言，只是每一次的计算量增加了，感觉上来说应该会更易实现一些。下面具体来看第二种方法。

$$e^{-\tau H} = (e^{-\delta\tau H})^M, e^{-\delta\tau H} = e^{-\delta\tau h_1} e^{-\delta\tau h_2} e^{-\delta\tau h_3} e^{-\delta\tau h_4} + O(\delta\tau^2)。$$

仿照一维情形，先把哈密顿量拆分成互相对易的四个部分，每次只需维护其中一种。更具体地，分开过后的哈密顿量作用于两个格点之上，且当前步骤中在这两个格点上的作用是局域的，因此我们实际上只需要解决如下问题，具体方法如下：



先把三个方向的 D 合成一个维度 D^3 ，然后把黄色框进行 SVD 分解，注意把 d, D 边收缩乘矩阵的一边，因此得到的新的键的上限为 $\min(D^3, dD) = dD$ ；再对于中间整体部分（红色框）进行收缩计算得到一个 $d^2D \times d^2D$ 矩阵，并再次进行 SVD 分解；最后合并回到标准构型。可以看出一次操作过程中中间键的维度从 $D \rightarrow d^2D$ 是上升的，不过看起来在可接受范围内。

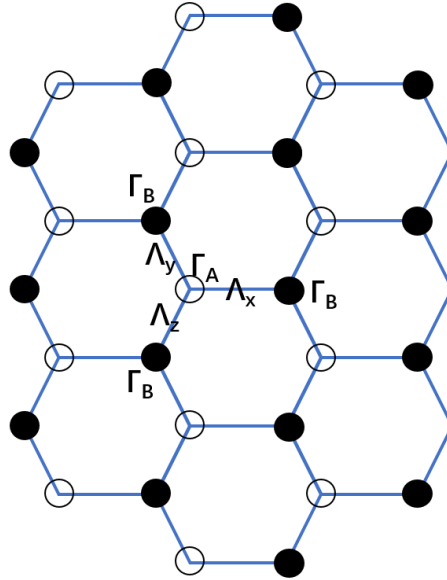
需要注意的是这里的截断与 MPS 中又有所不同，因为我们不再具有键中心正交形式。在 MPS 截断过程中我们丢弃较小奇异值的原因在于两侧都是正交矩阵，即实际上进行了态的 Schmidt 分解，然

而对于 PEPS, 无法这么做, 因此我们又得通过一次新的优化得到截断后的结果。

具体而言, 我们想要求得张量 A , 使得 $K = ||\psi_B(d^2D)\rangle - |\psi_A(D)\rangle||^2 = \langle\psi_A|\psi_A\rangle + \langle\psi_B|\psi_B\rangle - \langle\psi_A|\psi_B\rangle - \langle\psi_B|\psi_A\rangle$ 最小化。

将 \bar{A}, A 视作独立变量, 可以得到 $\frac{\partial K}{\partial \bar{A}_{\alpha_i \beta_i \gamma_i \delta_i}^{s_i}} = N_i A_i - W_i = 0$, 于是求解 $N_i A_i = W_i$ 即可得到截断后的近似张量。注意其中的矩阵 N_i, W_i 可以这么理解, $\langle\psi_A|\psi_A\rangle = A_i^\dagger N_i A_i, \langle\psi_A|\psi_B\rangle = A_i^\dagger W_i$, 都是关于张量 \bar{A}_i 的线性方程。其中 W_i 可以当作是在原张量网络中把 \bar{A}_i 部分拿掉, 剩下的脚组成一个指标所构成的向量; N_i 即把 $\bar{A}_i A_i$ 都拿掉, 各自剩下的脚分别组成矩阵的行和列构成的矩阵。

最后, 我们讨论一个最为可解的问题, 即无限大网络。仿照之前 iTEBD 的思路, 我们只需要维护很少的几个张量即可, 需要注意的是我们需要使用 $\Gamma - \Lambda$ 形式, 考虑无限大六边形网格:



每一次更新过程为 $\{\Gamma_A, \Gamma_B, \Lambda_x, \Lambda_y, \Lambda_z\} \rightarrow \{\Gamma'_A, \Gamma'_B, \Lambda'_x, \Lambda_y, \Lambda_z\}$, 流程与一维的 iTEBD 的更新过程几乎相同。得到的结果与使用其他主流方法的结果相比, 还是具有较高的可信度的:

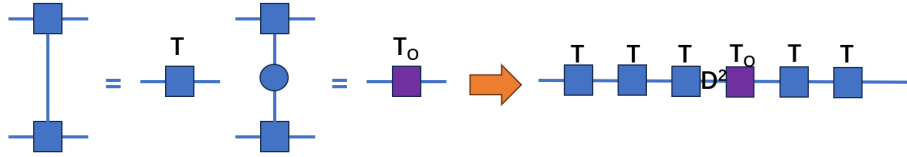
TABLE II. Comparison of our results with those obtained by other approaches for the ground state energy per site E and the staggered magnetization M of the Heisenberg model with $h = 0$.

Method	E	M
Spin wave [12]	-0.5489	0.24
Series expansion [13]	-0.5443	0.27
Monte Carlo [14]	-0.5450	0.22
Ours $D = 8$	-0.5506	0.21 ± 0.01

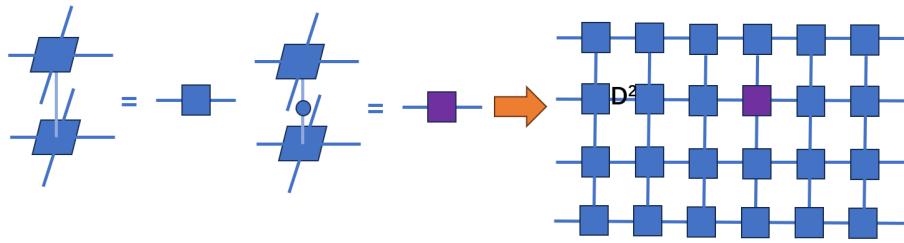
9.3 PEPS 的收缩

9.3.1 张量重整化群

我们现在来考虑收缩问题，最一般地，我们考虑计算 $\langle O_i \rangle$ 。对于一维问题，我们引入转移矩阵进行计算：



然而对于二维问题，转移矩阵变成了转移张量；最终得到的是一个二维的张量网络，想要进行收缩计算是很复杂的，因为我们不再具有一维那样显而易见的从左到右的计算形式：



为了解决这个复杂的收缩问题，我们引入张量重整化群（TRG）的方法。其核心思路是找到一种把几个小块合成为一个大块的方式，每个合成过程中进行截断，并且合成之后的张量排布方式与之前相似，因此可以持续合成下去。我们以二维蜂巢的伊辛模型为例，解释该过程：

首先我们需要把伊辛模型的哈密顿量用张量网络的方式表达出来。事实上这个过程正是我们一开始引入张量网络想法的起源。 $H = - \sum_{\{\sigma\}} \sigma_i \sigma_{i+1}$, $\sigma_i = \pm 1$ ，对于一维情形我们可以引入转移矩阵：

$$Z = e^{-\beta H} = e^{-\sigma_1 \sigma_2} \cdot e^{-\sigma_2 \sigma_3} \cdot e^{-\sigma_3 \sigma_4} \dots = \text{tr} \left(\begin{pmatrix} e^\beta & e^{-\beta} \\ e^{-\beta} & e^\beta \end{pmatrix} \begin{pmatrix} e^\beta & e^{-\beta} \\ e^{-\beta} & e^\beta \end{pmatrix} \dots \right) = \text{tr}(T^N),$$

有了转移矩阵能得到一维情形下的简单表达式，但是对于二维情况的推广还不够，因为一维情形下一个点只有两条边相连，而转移矩阵描述的是每条边，横纵坐标的一次相乘恰好对应了一个点被两条边连接。为了适应更高的连接数，我们应该设法把转移矩阵拆分，重新对应到每个点上去。事实上，只需要进行对角化即可。

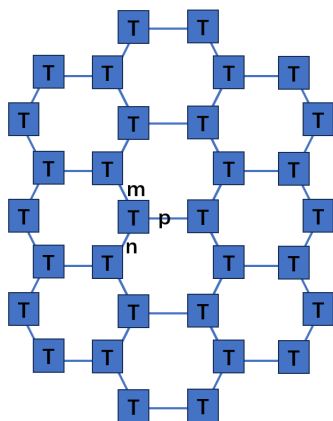
$$T = U \Lambda U^\dagger = e^\beta I + e^{-\beta} \sigma_x \Rightarrow \Lambda_\pm = e^\beta \pm e^{-\beta}, U = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

$$\text{进一步令 } \tilde{U} = U \sqrt{\Lambda}, \text{ 又 } U^\dagger = U^T, \Rightarrow T_{ij} = \sum_m \tilde{U}_{im} \tilde{U}_{jm}.$$

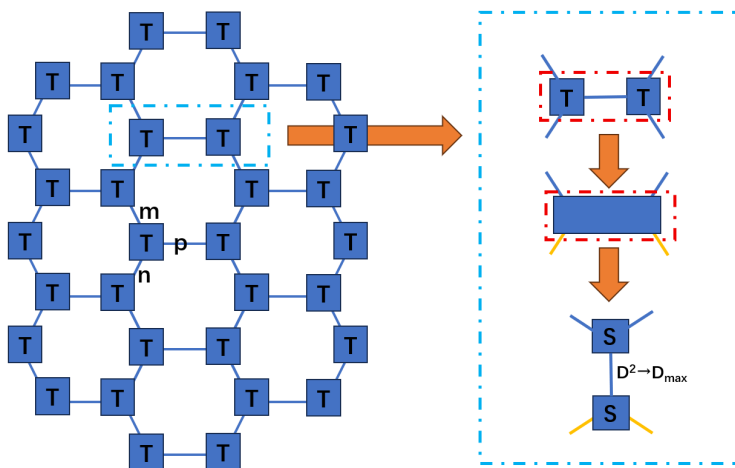
$$\text{如果按照边分类，得到的即为以上转移矩阵形式， } Z = \prod_{\{m\}} \sum_m \tilde{U}_{im} \tilde{U}_{jm};$$

同样地，我们可以按照点分类，得到 $Z = \prod_{\{i\}} \sum_m \tilde{U}_{im} \tilde{U}_{in}$ ，这是一个张量网络的形式，其中的求和 \sum_m 代表对每一个相同的边进行收缩操作，在每一个格点处有一个张量 $T_{mn}^i = \tilde{U}_{im} \tilde{U}_{in}$ 。

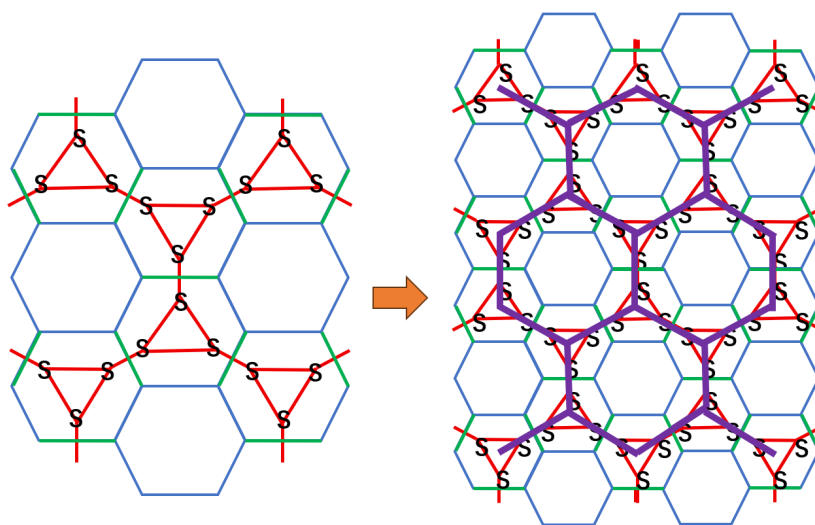
有了张量网络形式，很容易推广到蜂巢形式： $T_{mnp} = \tilde{U}_{im} \tilde{U}_{in} \tilde{U}_{ip}$ ，得到如下网络：



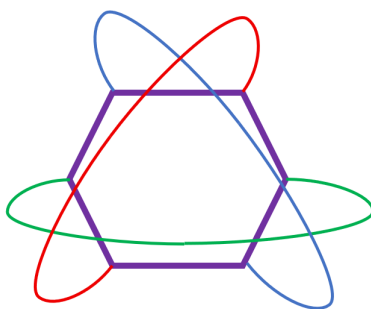
接下来是重整化群步骤：我们先做一次局部的收缩和 SVD 分解，注意这么做的代价是中间的键的重数升高了，因此在必要的时候需要进行截断，保持键的重数不超过 D_{max} ；



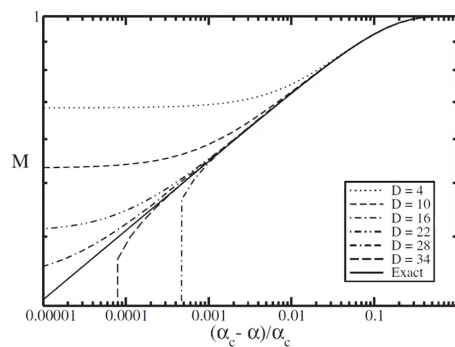
注意这里的数量关系，一个 T 与一个 S 相互对应，我们可以对整个蜂巢上所有的 T 进行类似的操作。按照下方左图方式，所有绿色边连接的 T 被改写为与之垂直的红色边上的 S ，有趣的是红色的 S 组成了三角形结构，我们可以进一步把这个三角形的 S 合并，形成一个更大的三阶张量（三个脚），于是得到了右图。右图中的紫色六边形网格中的每个顶点放置了一个新的张量 S' 。



我们可以观察到这一次过程中，最后得到的仍然是六边形网络，但是张量数量却只有初始时的 $\frac{1}{3}$ (因为 3 个 S 合并成一个 S')。于是我们可以继续在新的六边形网格上进行重整化操作。于是张量的数量以指数形式下降，最终达到我们所希望的形式，即一个单独的六边形网格，考虑边界条件，应该是对角位置相互连接：

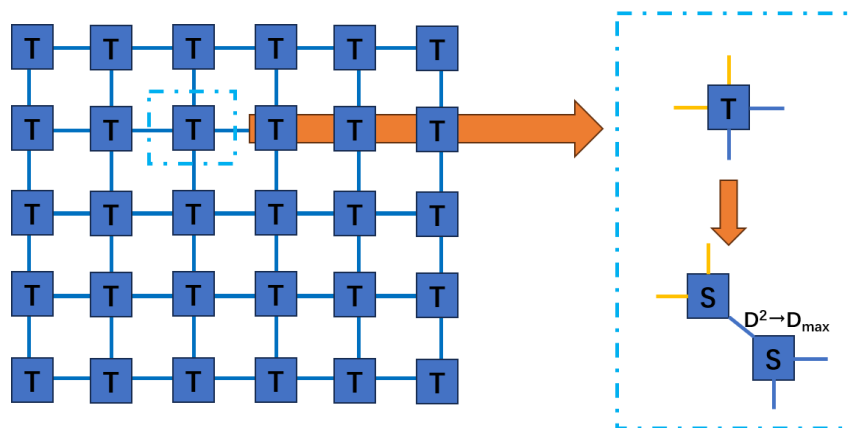


以下是文献中给出的计算结果，可以看到实际上想要得到比较好的模拟结果，需要的键重数是不

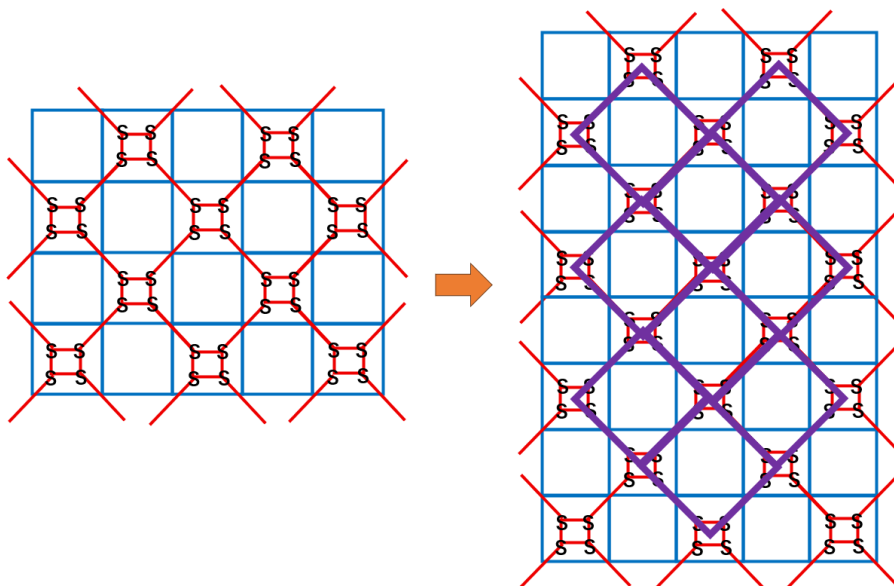


结果的不准确性实际上是好理解的，因为我们进行的截断过程都是局域的截断，并没有像先前那样采用正交形式（实际上是进行了 Schmidt 分解）后的截断。局域的截断显然引入的误差会大不少，当然这个问题目前还没有很好的解决受到。不过虽然我们还没有办法解决局域截断不准确的问题，但是至少我们找到了一种有规律的收缩方式。

对于正方形网格，我们也可以进行相似的过程：

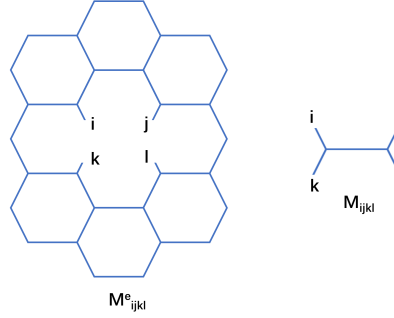


将每个格点的 T 按照如下左图顺序进行拆解，可以得到 S ，此时 4 个 S 合并成为 1 个 S' ，实现了右图的重整化过程。紫色格点处即为新的 S' ，仍然是正方形网格，可以继续进行收缩。所需的张量数量以指数形式下降，但是同时我们必须引入截断以防止键的重数过高：



9.3.2 二次重整化群 (SRG)

上一节的讨论中指出, 直接进行局部的截断的误差较大。想要进行更好的截断, 我们必须要知道环境张量的信息, 综合二者共同进行截断。假设我们现在拥有待优化张量 M_{ijkl} 和环境张量 M_{ijkl}^e , 如下图所示:



考虑对环境张量部分进行 SVD 分解: $M_{ij,kl}^e = \sum_m U_{ij,m}^e \Lambda_m^e V_{m,kl}^{e\dagger}$;

配分函数 $Z = \sum_{ijkl} M_{ij,kl} M_{kl,ij}^e = \sum_{ijkl} \sum_m M_{ij,kl} U_{kl,m}^e \Lambda_m^e V_{m,ij}^{e\dagger} = \sum_{ijkl} \sum_m \sqrt{\Lambda_m^e} V_{m,ij}^{e\dagger} M_{ij,kl} U_{kl,m}^e \sqrt{\Lambda_m^e}$,

$\tilde{M} = \sqrt{\Lambda^e} V^{e\dagger} M U^e \sqrt{\Lambda^e}$, $Z = \text{tr}(\tilde{M})$ 。

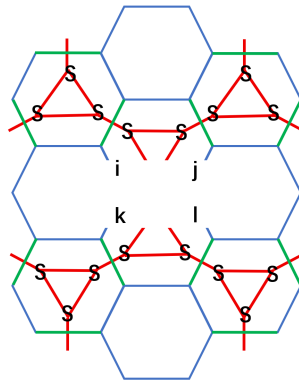
由此可知, 实际上对于 \tilde{M} 进行截断才是最合理的, 注意这里的 \tilde{M} 是一个 $D^2 \times D^2$ 的矩阵。

对 \tilde{M} 做 SVD 分解, $\tilde{M} = \tilde{U} \tilde{\Lambda} \tilde{V}^\dagger$, 并进行截断, $\tilde{\Lambda}$ 的维度为 D_{max} :

$M = S_a S_b = \tilde{\Lambda}^{-\frac{1}{2}} V^e \tilde{M} U^{e\dagger} \tilde{\Lambda}^{-\frac{1}{2}} = \tilde{\Lambda}^{-\frac{1}{2}} V^e \tilde{U} \tilde{\Lambda}^{\frac{1}{2}} \tilde{\Lambda}^{\frac{1}{2}} \tilde{V}^\dagger U^{e\dagger} \tilde{\Lambda}^{-\frac{1}{2}}$,

$\Rightarrow S_a = \tilde{\Lambda}^{-\frac{1}{2}} V^e \tilde{U} \tilde{\Lambda}^{\frac{1}{2}}$, $S_b = \tilde{\Lambda}^{\frac{1}{2}} \tilde{V}^\dagger U^{e\dagger} \tilde{\Lambda}^{-\frac{1}{2}}$, 中间的键重数即为 D_{max} 。

现在的问题是我们如何得到环境张量 M_{ijkl}^e 。实际上并没有什么太好的做法, 我们只能对环境张量再进行一次局部截断的 TRG。即对于其余部分, 利用上一节的局部截断的方式计算得到结果, 再待入此处并优化张量 M_{ijkl} , 然后利用此时优化后的张量再进行整体的 TRG 过程。由于进行了两次张量网络重整化群, 该方法称为 SRG。实际上这个思路和微扰论有点像, 即把一阶结果代入二阶展开得到二阶近似。



从结果上来看，利用二次重整化群，的确能够得到更精确的结果：

