SFB 1143

Tensor Networks (SS2021)

Lecture 19: Tensor renormalization group

Hong-Hao Tu (*ITP, TU Dresden*)

Email: hong-hao.tu@tu-dresden.de
Zoom: tuhonghao@gmail.com

July 15th, 2021

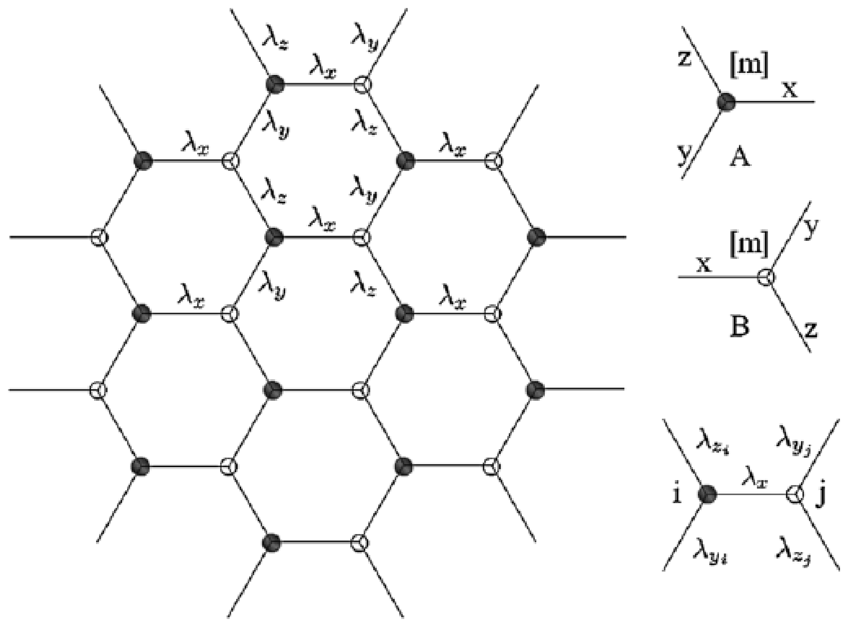
§ 4.2 PEPS: Optimization and contraction

- Infinite PEPS algorithm: simple update

2D analog of iTEBD:

$$\lim_{\tau\to\infty} e^{-\tau H}|\psi\rangle \to |\psi(D)\rangle$$

$$e^{-\delta\tau H} \approx e^{-\delta\tau H_1} e^{-\delta\tau H_2} \times e^{-\delta\tau H_3} + O(\delta\tau^2)$$

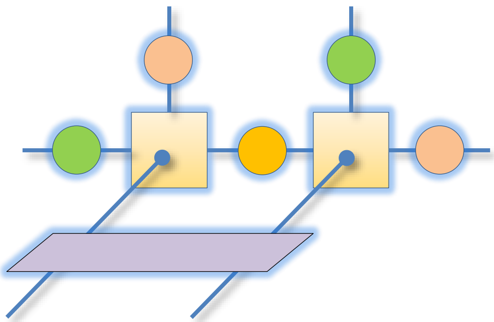


$\lambda_{x,y,z} > 0$

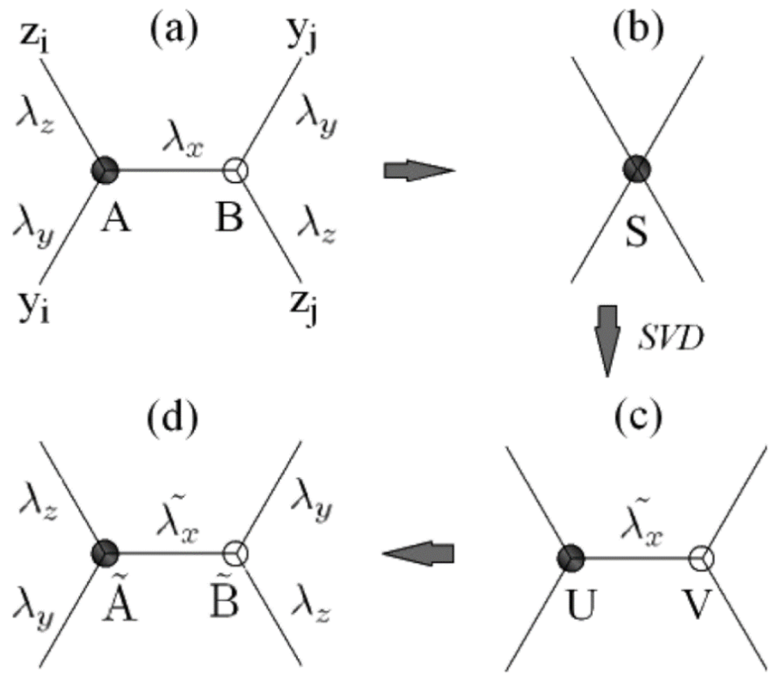
H. C. Jiang, Z. Y. Weng & T. Xiang, Phys. Rev. Lett. 101, 090603 (2008).

§ 4.2 PEPS: Optimization and contraction

- Infinite PEPS algorithm: simple update



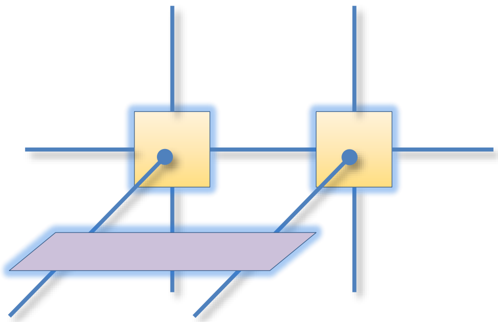
Local update!

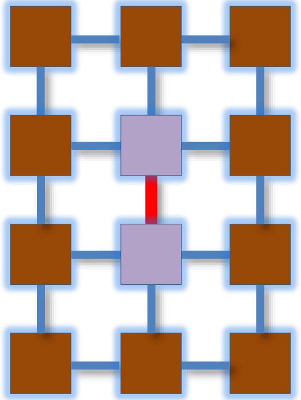


H. C. Jiang, Z. Y. Weng & T. Xiang, Phys. Rev. Lett. 101, 090603 (2008).

§ 4.2 PEPS: Optimization and contraction

- Infinite PEPS algorithm: full update





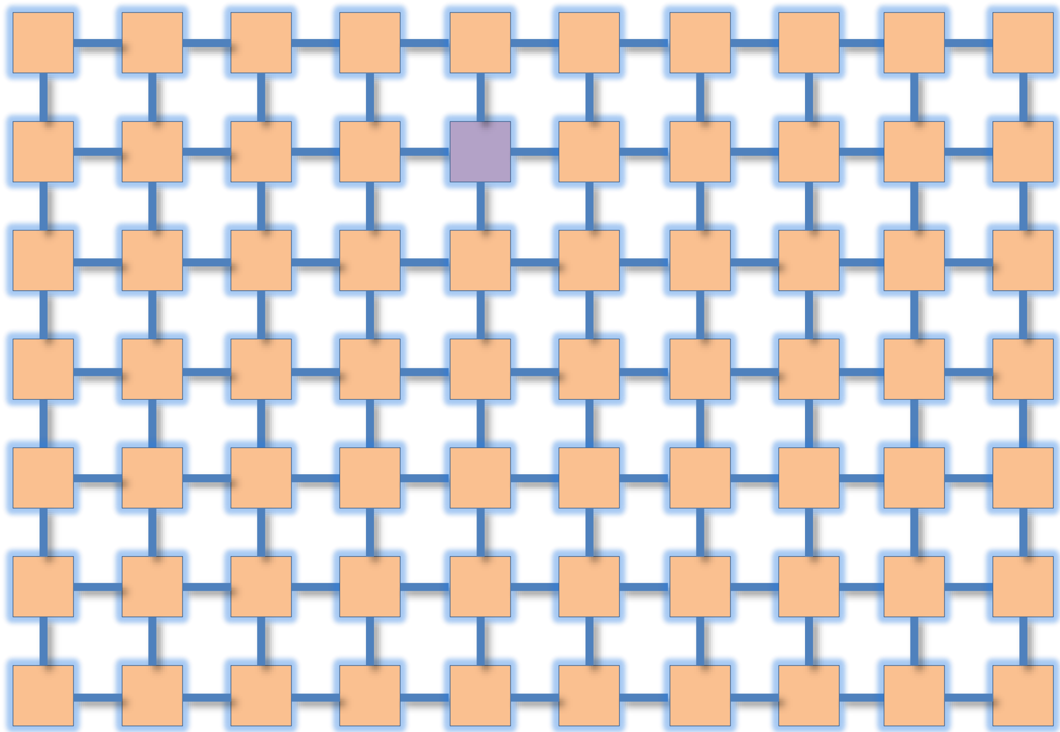
➢ Calculate the full environment (using MPS techniques)

➢ Truncate the bond dimension variationally

J. Jordan, R. Orús, G. Vidal, F. Verstraete & J. I. Cirac, Phys. Rev. Lett. 101, 250602 (2008).

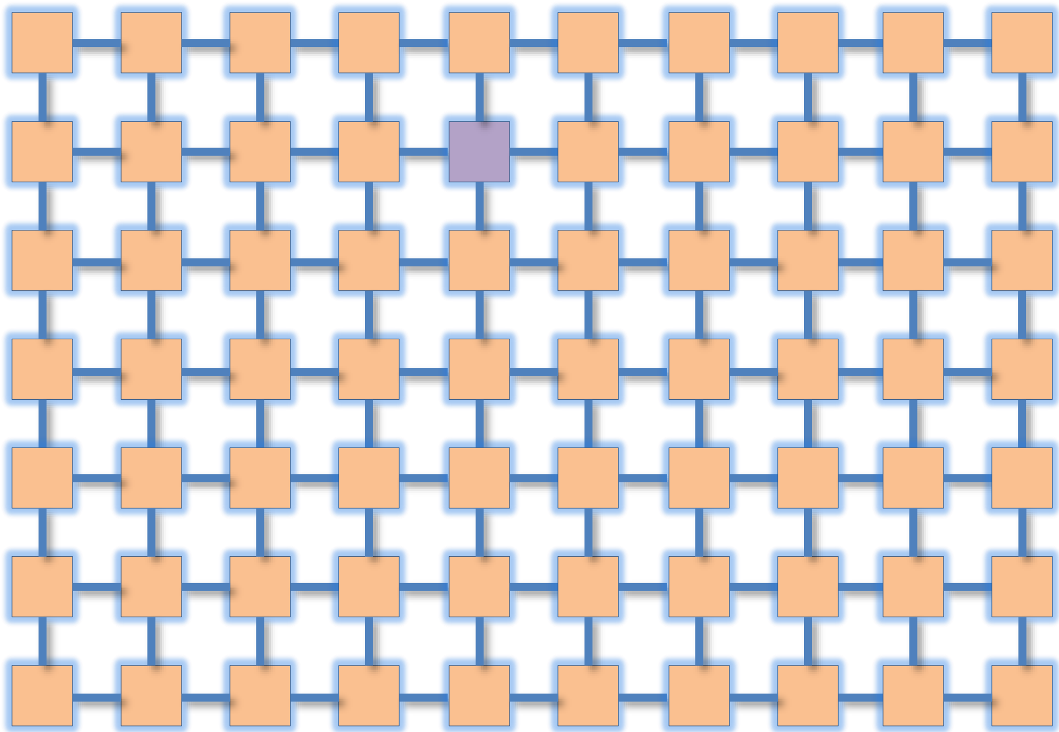
§ 4.2 PEPS: Optimization and contraction

$$\langle\psi|O_i|\psi\rangle =$$



So far we have been using MPS techniques to contract 2D tensor networks (such as the norm or expectation values). Any alternatives?

§ 4.2 PEPS: Optimization and contraction

$\langle\psi|O_i|\psi\rangle =$ 

Methods motivated by the real-space RG (c.f. Kadandoff's "block spin" approach) provide such an alternative!

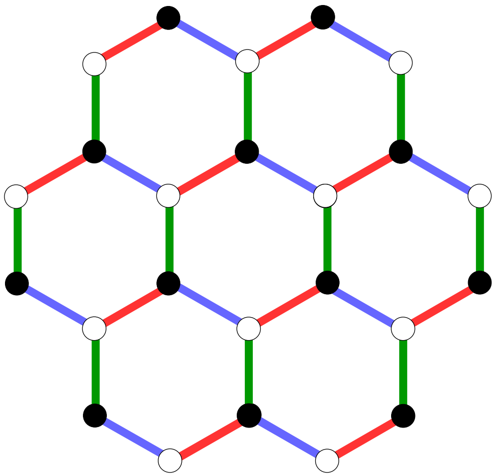
§ 4.3 Tensor renormalization group

- Partition functions of classical statistical models can be formulated with tensor networks.

Example: 2D classical Ising model

$$Z = \sum_{\{\sigma\}} e^{-\beta H(\{\sigma\})}$$

$$H(\{\sigma\}) = -\sum_{\langle ij \rangle} \sigma_i \sigma_j$$



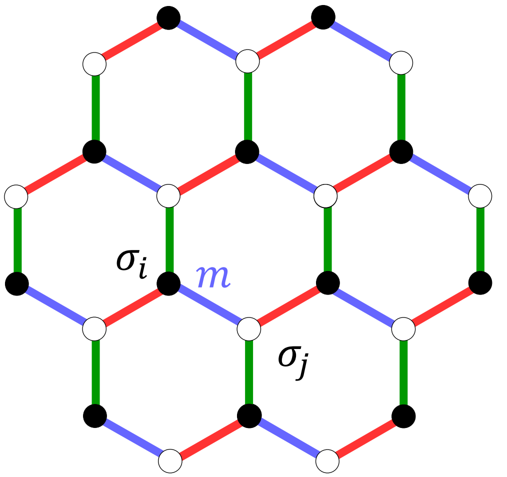
$\sigma_i = \pm 1$

M. Levin & C. P. Nave, Phys. Rev. Lett. 99, 120601 (2007).

§ 4.3 Tensor renormalization group

Convert the partition function into local tensor networks:

$$Z = \sum_{\{\sigma\}} e^{\beta(\sigma_1\sigma_2+\sigma_2\sigma_3+\cdots)}$$



$$e^{\beta\sigma_i\sigma_j} = \begin{pmatrix} e^{\beta} & e^{-\beta} \\ e^{-\beta} & e^{\beta} \end{pmatrix}_{\sigma_i\sigma_j}$$

$$= [U\Lambda U^T]_{\sigma_i\sigma_j}$$

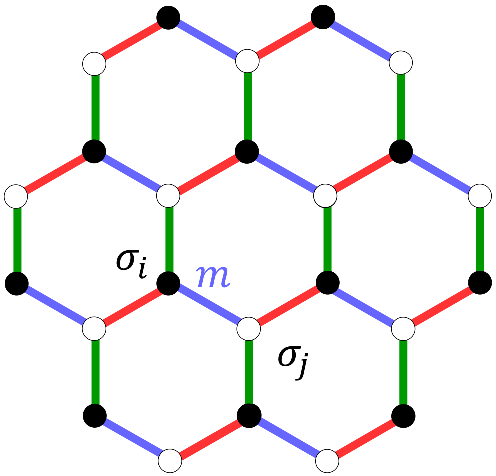
$$\Lambda = \begin{pmatrix} e^{\beta}+e^{-\beta} & 0 \\ 0 & e^{\beta}-e^{-\beta} \end{pmatrix} \qquad U = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

In general, you may use SVD.

§ 4.3 Tensor renormalization group

Convert the partition function into local tensor networks:

$$Z = \sum_{\{\sigma\}} e^{\beta(\sigma_1\sigma_2+\sigma_2\sigma_3+\cdots)}$$



$$e^{\beta\sigma_i\sigma_j} = \begin{pmatrix} e^{\beta} & e^{-\beta} \\ e^{-\beta} & e^{\beta} \end{pmatrix}_{\sigma_i\sigma_j}$$

$$= [U\Lambda U^T]_{\sigma_i\sigma_j}$$

$$= \left[\widetilde{U}\widetilde{U}^T\right]_{\sigma_i\sigma_j} \qquad \widetilde{U} = U\Lambda^{1/2}$$

$$= \sum_{m=1,2} \widetilde{U}_{\sigma_i m}\widetilde{U}_{\sigma_j m}$$

§ 4.3 Tensor renormalization group

Convert the partition function into local tensor networks:

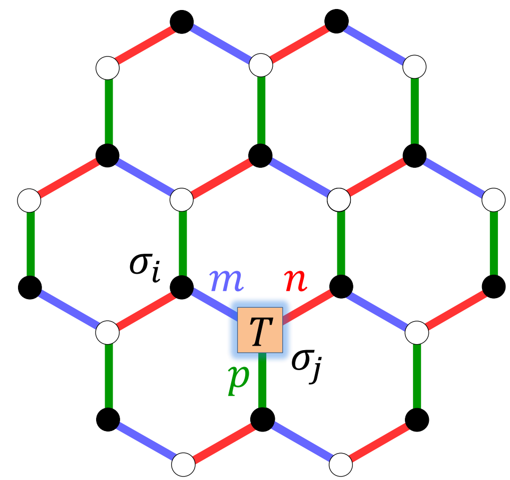
$$Z = \sum_{\{\sigma\}} e^{\beta(\sigma_1\sigma_2+\sigma_2\sigma_3+\cdots)}$$

$$e^{\beta\sigma_i\sigma_j} = \begin{pmatrix} e^{\beta} & e^{-\beta} \\ e^{-\beta} & e^{\beta} \end{pmatrix}_{\sigma_i\sigma_j}$$

$$= [U\Lambda U^T]_{\sigma_i\sigma_j}$$

$$= \left[\widetilde{U}\widetilde{U}^T\right]_{\sigma_i\sigma_j} \qquad \widetilde{U} = U\Lambda^{1/2}$$

$$= \sum_{m=1,2} \widetilde{U}_{\sigma_i m}\widetilde{U}_{\sigma_j m}$$



$$T_{mnp} = \sum_{\sigma_j} \widetilde{U}_{\sigma_j m}\widetilde{U}_{\sigma_j n}\widetilde{U}_{\sigma_j p}$$

§ 4.3 Tensor renormalization group

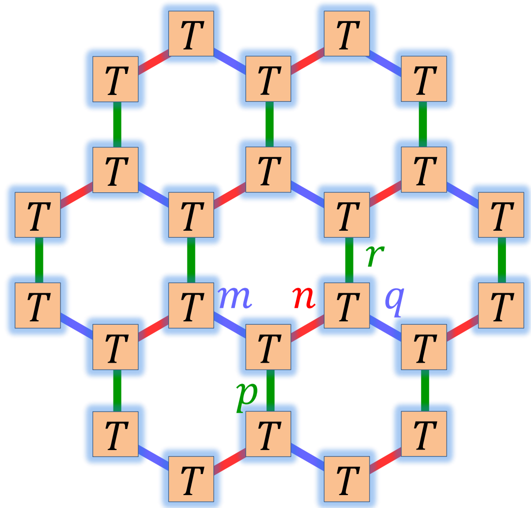
Convert the partition function into local tensor networks:

$$Z = \sum_{\{\sigma\}} e^{\beta(\sigma_1\sigma_2 + \sigma_2\sigma_3 + \cdots)}$$

$$e^{\beta\sigma_i\sigma_j} = \sum_{m=1,2} \widetilde{U}_{\sigma_i m} \widetilde{U}_{\sigma_j m}$$

$$Z = \sum_{\{\dots m,n,p,q,r\dots\}} (\cdots T_{mnp} T_{qnr} \cdots)$$

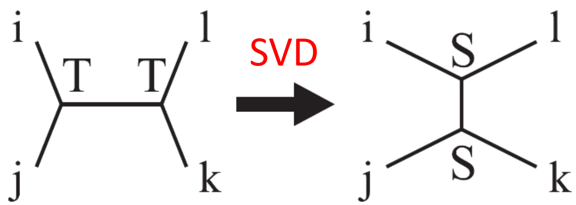
(also known as "vertex models" in statistical mechanics)

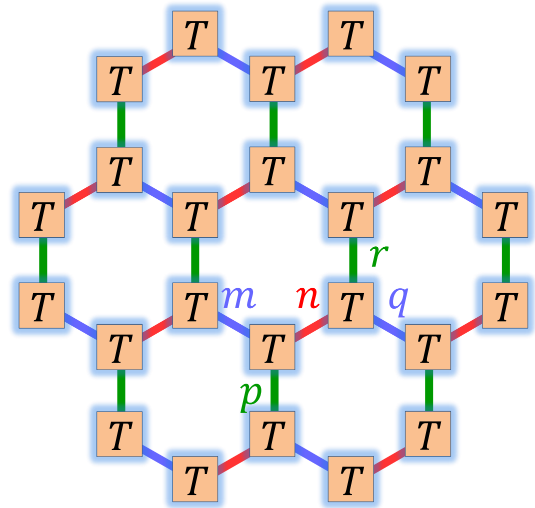


$$T_{mnp} = \sum_{\sigma_j} \widetilde{U}_{\sigma_j m} \widetilde{U}_{\sigma_j n} \widetilde{U}_{\sigma_j p}$$

§ 4.3 Tensor renormalization group

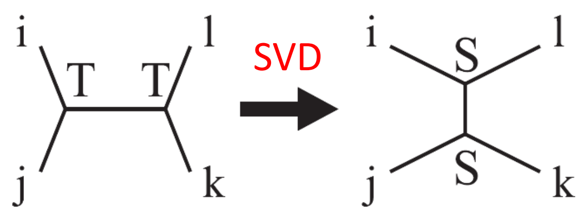
Calculate the partition function by contracting the tensor network:

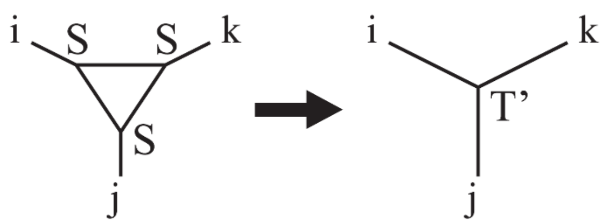


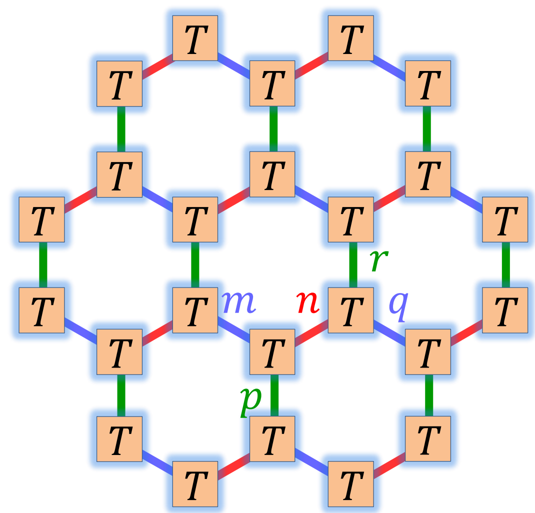


§ 4.3 Tensor renormalization group

Calculate the partition function by contracting the tensor network:

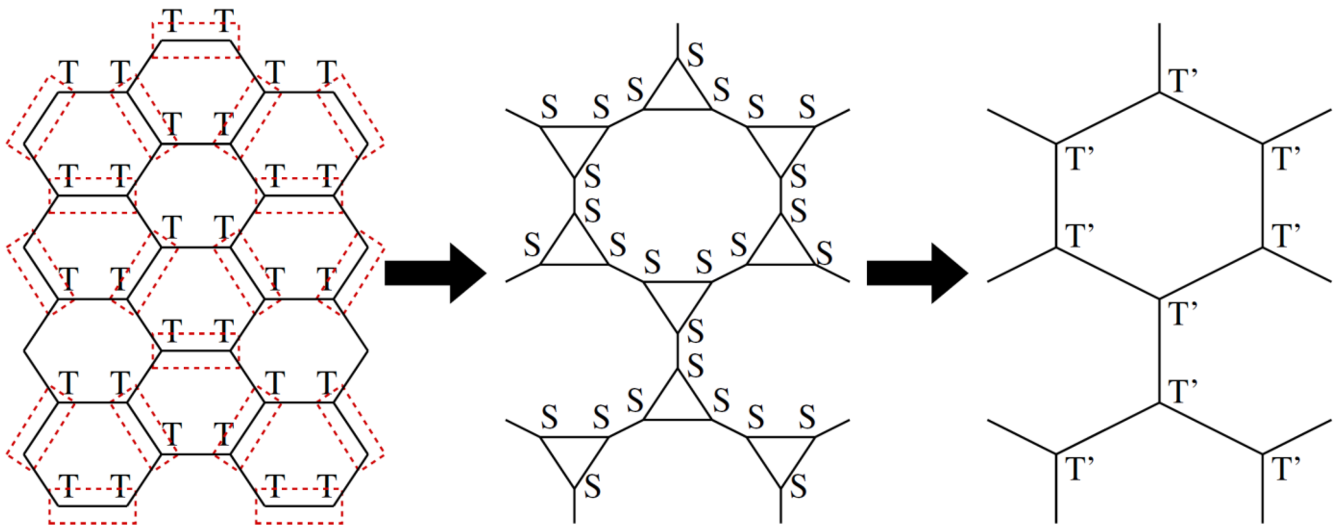






§ 4.3 Tensor renormalization group

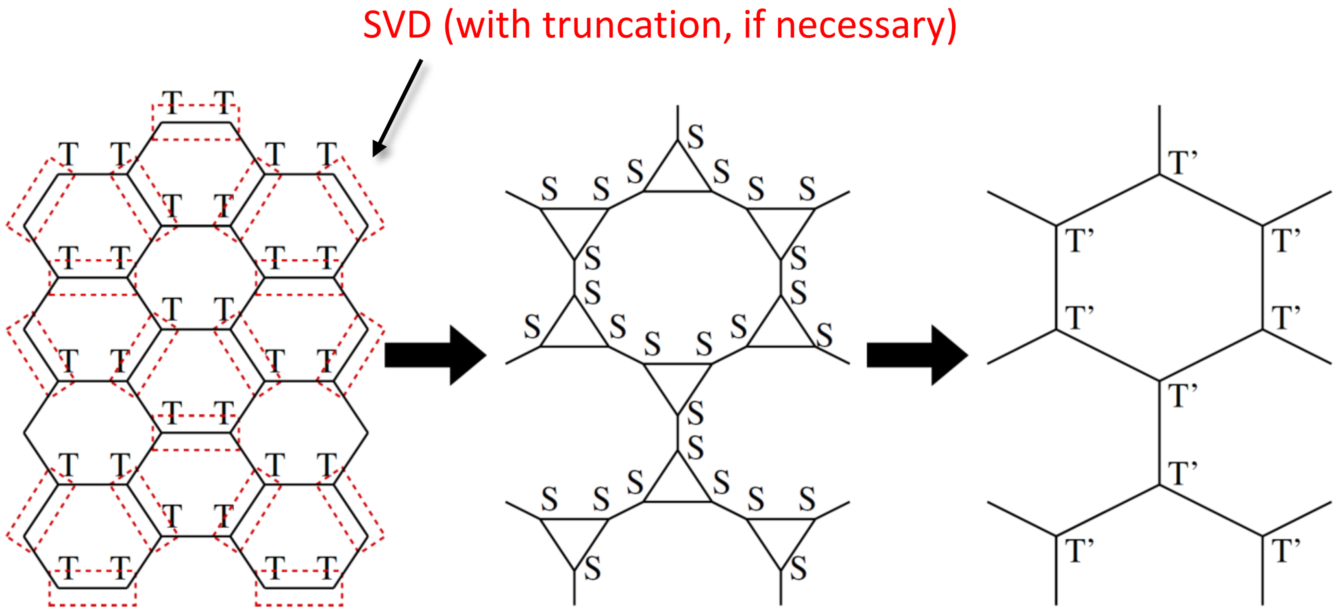
Calculate the partition function by contracting the tensor network:



One RG step: # of sites $N \rightarrow N/3$

§ 4.3 Tensor renormalization group

Calculate the partition function by contracting the tensor network:

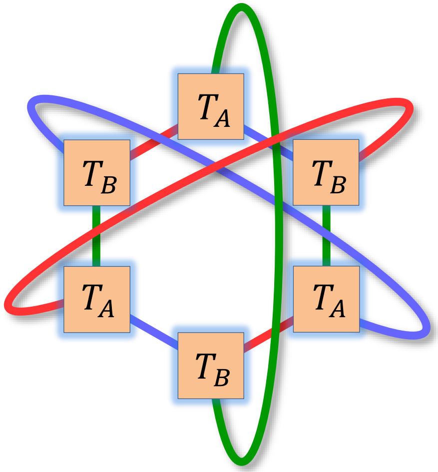


Repeat the RG steps to reduce the # of sites

§ 4.3 Tensor renormalization group

After M RG steps, only one (coarse-grained) hexagon remains:

# of sites: 6×3^M

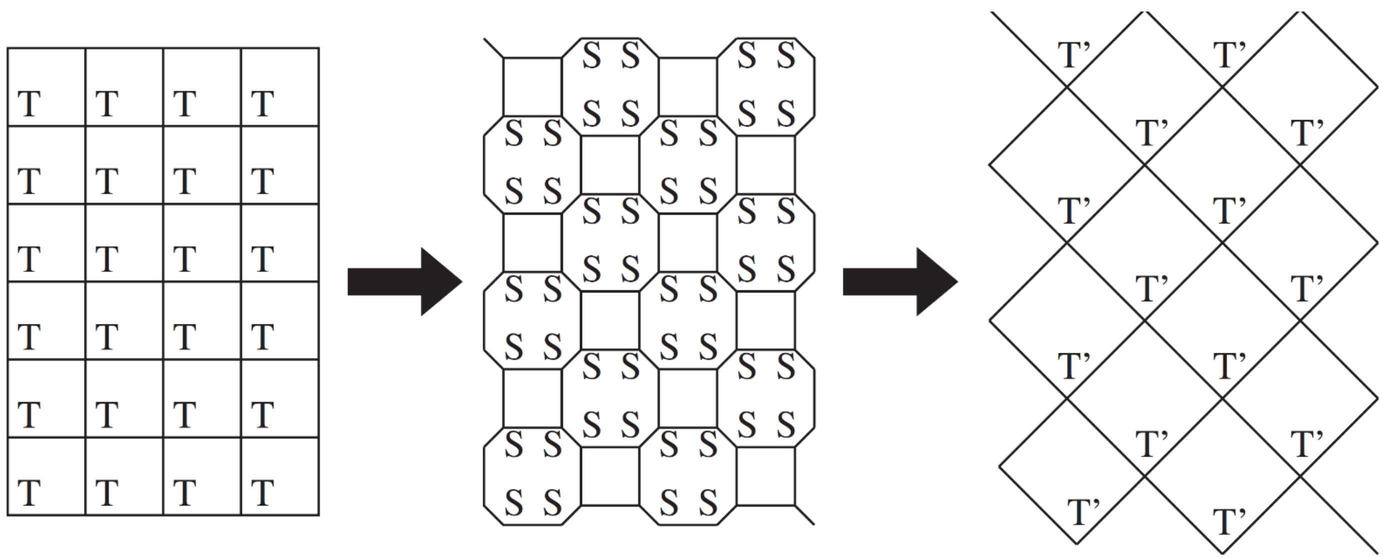


Contract the tensors to obtain Z and calculate free energy $F = -\frac{1}{N\beta} \ln Z$.

§ 4.3 Tensor renormalization group

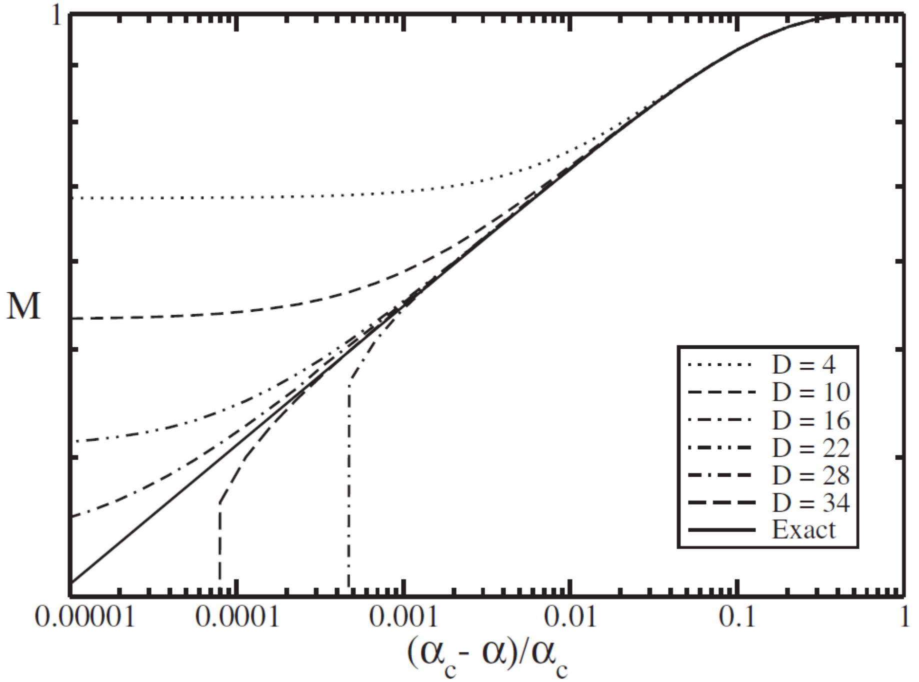
The TRG for other lattices can be similarly designed:

Square lattice:



§ 4.3 Tensor renormalization group

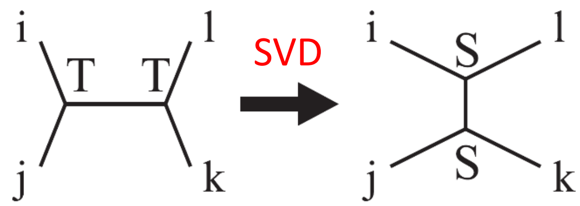
TRG Results for the magnetization of the triangular-lattice Ising model:

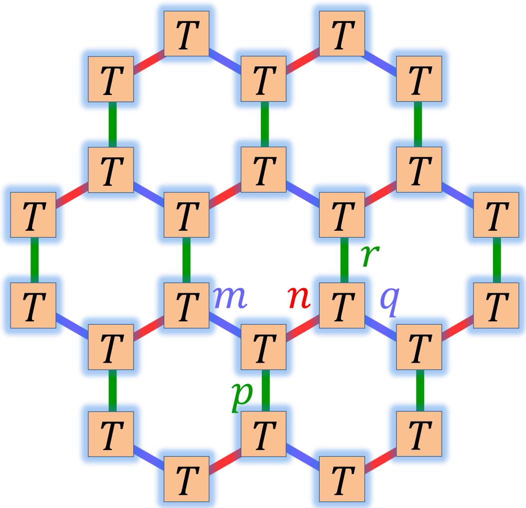


$\alpha = e^{-2\beta J}$

§ 4.3 Tensor renormalization group

Drawback: Truncation is done locally, i.e., environment is ignored!





Several refined methods available, such as SRG, HOTRG, TNR, … (still an active research topic!)