Corner Transfer Matrix Renormalization Group

an Algorithm for 2D Classical Lattice Models
a Literature Digestion of T. Nishino and K. Okunishi's article (1996).

Chengzhi Ye

ENS de Lyon

January 5, 2026

Outline

1. From 1D to 2D: The Contraction Issue

2. Going to 2D: A Fundamentally Harder Problem

3. CTMRG: Extending to 2D

4. Summary

Review: 1D Transfer Matrix (Audience's Pre-knowledge)

1D Ising Model:

$$H = -J\sum_{i} \sigma_i \sigma_{i+1}$$

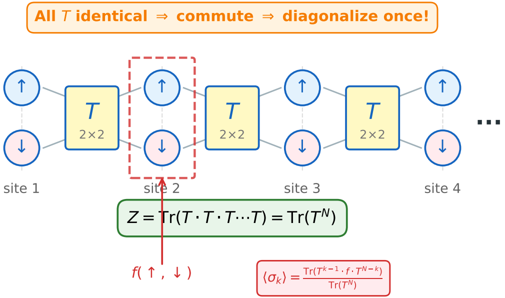
Partition function as contraction:

$$Z = \mathrm{Tr}(T \cdot T \cdot T \cdots T) = \mathrm{Tr}(T^N)$$

where $T \in \mathbb{R}^{2\times 2}$ encodes local interactions.

Local observable:

$$\langle \sigma_k \rangle = \frac{\mathrm{Tr}(T^k \cdot f(\uparrow,\downarrow) \cdot T^{N-k})}{\mathrm{Tr}(T^N)}$$



All T identical ⇒ commute ⇒ diagonalize once!

$Z = \mathrm{Tr}(T \cdot T \cdot T \cdots T) = \mathrm{Tr}(T^N)$

$\langle \sigma_k \rangle = \frac{\mathrm{Tr}(T^{k-1} \cdot f \cdot T^{N-k})}{\mathrm{Tr}(T^N)}$

Key Point

When Even 1D Classical Needs Renormalization Group

Recall: For uniform 1D Ising, $Z = \mathrm{Tr}(T^N)$ has no contraction issues.

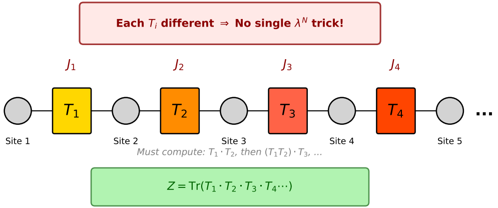
But what if...

- Couplings J_i are **site-dependent**?
- Random disorder: $J_i \sim \mathcal{N}(\bar{J}, \sigma)$?
- Open boundary conditions (no translation symmetry)?

Then: Cannot diagonalize T once!

$$Z = \mathrm{Tr}(T_1 \cdot T_2 \cdot T_3 \cdots T_N)$$

Each T_i is different $\Rightarrow$ no simple λ^N formula.



Each T_i different $\Rightarrow$ No single λ^N trick!

Must compute: $T_1 \cdot T_2$, then $(T_1 T_2) \cdot T_3$, ...

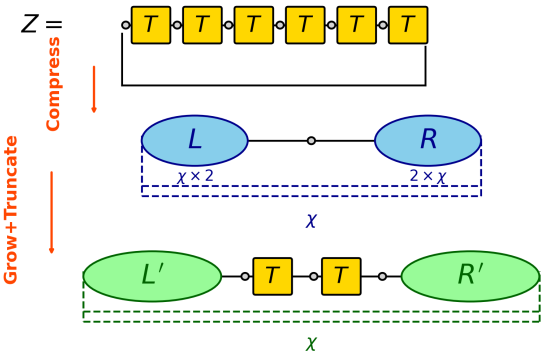
$Z = \mathrm{Tr}(T_1 \cdot T_2 \cdot T_3 \cdot T_4 \cdots)$

The Problem

For N sites: need $O(N)$ matrix multiplications. Not exponential, but **no closed-form solution**.

Solution: Transfer Matrix Renormalization Group (TMRG)

Insight: Not all configurations contribute equally to $Z \Rightarrow$ Keep only the **most relevant** ones!



χ = bond dimension (most relevant configurations in compressed basis)

TMRG Algorithm:

1. **Grow:** $L' = L \cdot T\ ,\ \ R' = T \cdot R$ (add sites)
2. **Truncate:** SVD $\rightarrow$ keep χ largest (coarse-grain)

RG Fixed Point

Iterate until L^*, R^* converge $\Rightarrow$ Thermodynamic limit!

Kind-of like infinite-DMRG by adding new pair of sites

2D: The Transfer Matrix Becomes a **Network**

1D: Ordered product of matrices

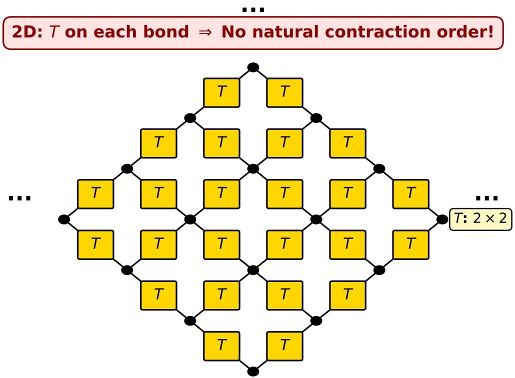
$$Z = \mathrm{Tr}(T_1 \cdot T_2 \cdots T_N)$$

Contract left-to-right: $O(N)$.

2D: T still 2×2 (from now on $d \times d$ in order to distinguish physical dimension $D = 2$) on each **bond**, but bonds form a **2D network**!

$\Rightarrow$ No natural ordering.

2D: T on each bond ⇒ No natural contraction order!



The Fundamental Problem

No natural contraction order! 2D tensor network cannot be reduced to a simple trace.

One Approach: Row-to-Row Transfer (DMRG-style)

Idea: Group one row of L spins → treat as a “super-spin” with d^L states.

[Figure: Row → super-spin, $T_{row} \in \mathbb{R}^{d^L \times d^L}$]

Then $Z = \mathrm{Tr}(T_{\mathrm{row}}^L)$ looks like 1D!

Apply DMRG/TMRG ideas:

- Row config space → MPS
- Row-to-row transfer → MPO
- Truncate via SVD

Pros & Cons

+ Systematic, well-understood

– Breaks 2D symmetry

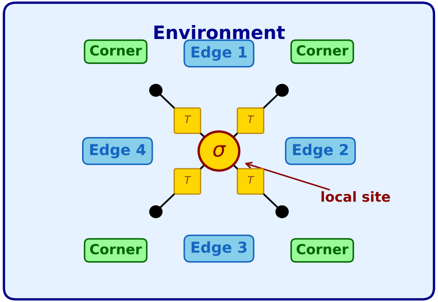
– Hard to generalize to other lattices

(Details in Appendix)

A More Natural Approach: Use Environment just like in 1D TMRG

Question: If we only want **local observables** $\langle \sigma_{i,j} \rangle$,
do we really need to contract the *entire* infinite lattice?

Only need local $\langle\sigma\rangle$? ⇒ Compress the environment!



$$\langle\sigma\rangle = \frac{\mathrm{Tr}(\mathrm{Environment}\cdot\sigma)}{\mathrm{Tr}(\mathrm{Environment})}$$

Key Insight:
Decompose the infinite 2D environment into **geometrically natural** pieces:

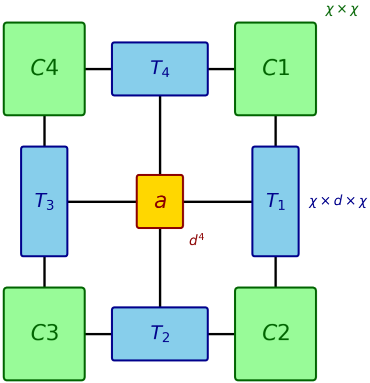
- **4 Corners** (quarter-planes)
- **4 Edges** (half-infinite strips)

Baxter's Insight (1968)

The **Corner Transfer Matrix** encodes a quarter of the infinite plane!

2D Analog: Four Corners + Four Edges

CTMRG: Decompose 2D environment into 4 Corners + 4 Edges



Corners C**:**

- Quadrant $(\chi \times \chi)$

Edges T**:**

- Half-strip $(\chi \times d \times \chi)$

Local tensor a**:**

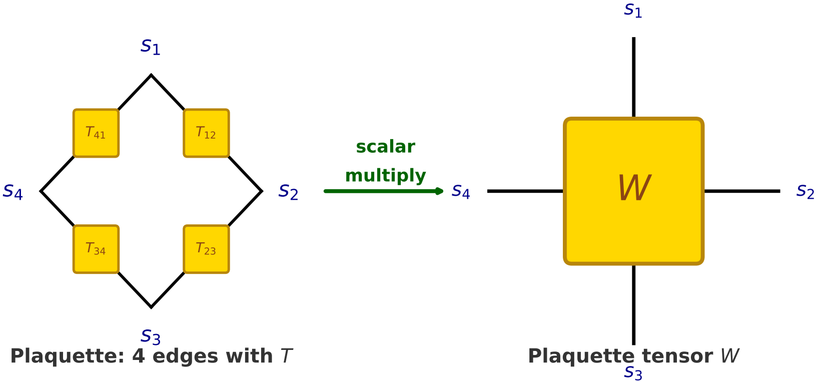
- Rank-4 (d^4)

Key: What is a?

a is a **rank-4 tensor** — for computing Z: $a = W$ *plaquette tensor*; for $\langle \sigma_1 \rangle$: $a = s_1 W_{s_1 s_2 s_3 s_4}$, etc.

The Plaquette Tensor: From Bonds to Faces

$$\boxed{W_{s_1,s_2,s_3,s_4} = T_{s_1,s_2} \cdot T_{s_2,s_3} \cdot T_{s_3,s_4} \cdot T_{s_4,s_1}}$$



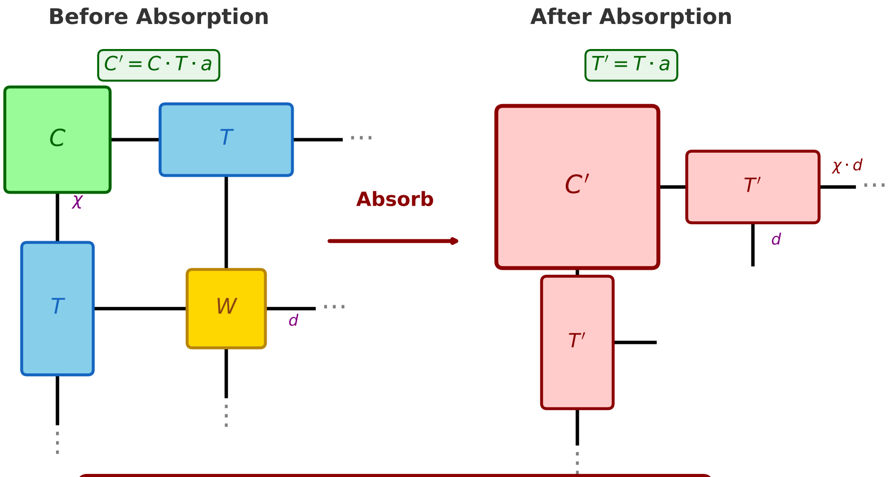
Plaquette: 4 edges with T

Plaquette tensor W

rank-4: (s_1, s_2, s_3, s_4)

Key: Simple scalar multiplication of 4 bond weights

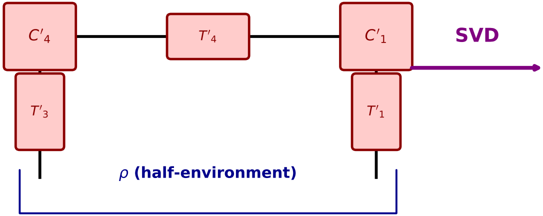
CTMRG Iteration: Step 1 — Grow (Absorb)



Bond dimension grows: $\chi \to \chi \cdot d$ — Need truncation!

CTMRG Iteration: Step 2 — Truncate

Truncation: Use environment to find optimal projector *P*

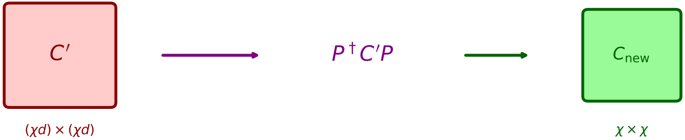


$\rho = U\Sigma V^\dagger$

Keep χ largest:

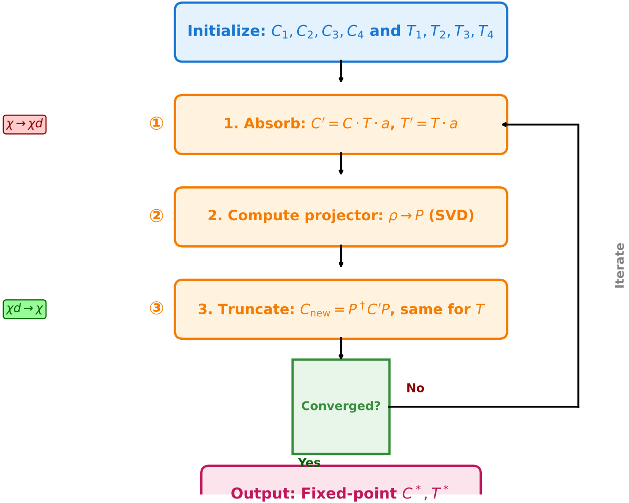
$\boxed{P = U_{:\chi}}$

Apply Projector *P*:



Similarly: $T_{\text{new}} = P^\dagger T' P$

CTMRG: Complete Algorithm



Initialize: C_1, C_2, C_3, C_4 **and** T_1, T_2, T_3, T_4

$\chi \to \chi d$ ① **1. Absorb:** $C' = C \cdot T \cdot a$, $T' = T \cdot a$

② **2. Compute projector:** $\rho \to P$ **(SVD)**

$\chi d \to \chi$ ③ **3. Truncate:** $C_{\text{new}} = P^{\dagger} C' P$, **same for** T

Converged?

No

Iterate

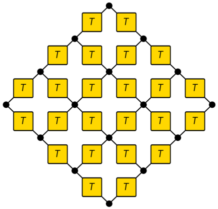
Yes

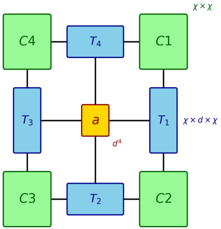
Output: Fixed-point C^*, T^*

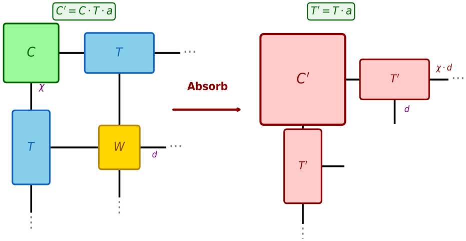
Summary: The Core of CTMRG

CTMRG in Three Steps

1. **Decompose:** Infinite square lattice → 4 corners + 4 edges (customed for other lattice);
2. **Grow and Truncate (RG):** add row+column then absorb, SVD to keep χ most relevant;
3. $Z = W_{s_1 s_2 s_3 s_4} \mathrm{Tr}(C_1\, T_1^{s_1}\, C_2\, T_2^{s_2}\, C_3\, T_3^{s_3}\, C_4\, T_4^{s_4})$; $\langle \mathcal{O} \rangle = \frac{\mathcal{O}_{s_1 s_2 s_3 s_4}}{Z} \mathrm{Tr}(C_1\, T_1^{s_1}\, C_2\, T_2^{s_2}\, C_3\, T_3^{s_3}\, C_4\, T_4^{s_4})$







Thank You All for Your Attention!

And thanks to Prof. Tommaso Roscilde, Dr. Fabio Mezzacapo, Filippo Caleca and Saverio Bocini for your teaching and assistance!

also a small thanks to Claude Opus that assists me generating *all* the schemas and teaching me `beamer`.

Appendix: TMRG Setup in Detail

Goal: Compute $Z = \mathrm{Tr}(T_1 \cdot T_2 \cdots T_N)$ for large N.

[Figure: Left environment L, local site T, right environment R]

Left Environment L:

$$L_\sigma = \sum T_1 \cdot T_2 \cdots T_{k-1}$$

Right Environment R:

$$R_\sigma = \sum T_{k+1} \cdots T_N$$

Appendix: TMRG Step 1 — Grow (Absorption)

Absorption: Add one more site to the environment.

[Figure: $L' = L \cdot T_k$, growing the left environment]

Mathematically:

$$L'_{\sigma_k} = \sum_{\sigma_{k-1}} L_{\sigma_{k-1}} \cdot (T_k)_{\sigma_{k-1},\sigma_k}$$

Appendix: TMRG Step 2 — Truncate (SVD)

Idea: Compress L' back to dimension χ using SVD.

Form the "density matrix":

$$\rho = L' \cdot R'^T$$

SVD:

$$\rho = U\Sigma V^\dagger$$

Truncate:

$$P = U_{:,1:\chi}$$

New environment:

$$L_{\text{new}} = P^\dagger L'$$

[Figure: SVD spectrum, keep χ largest]

Eckart-Young Theorem

SVD gives the **optimal** rank-χ approximation

Appendix: TMRG Convergence and Observables

Iterate: Grow → Truncate → Grow → Truncate → ⋯

Convergence criterion:

Fixed point: L^*, R^* such that

$$L^* \xrightarrow{\text{grow+truncate}} L^*$$

Free energy:

$$f = -k_B T \lim_{N\to\infty} \frac{1}{N} \ln Z$$

$$= -k_B T \ln \sigma_1^*$$

Local observables:

$$\langle \sigma_k \rangle = \frac{L^* \cdot \sigma_k \cdot R^*}{L^* \cdot R^*}$$

Correlation functions:

$$\langle \sigma_i \sigma_j \rangle = \frac{L^* \cdot \sigma_i \cdot T^{|i-j|} \cdot \sigma_j \cdot R^*}{Z}$$

Physical Meaning

L^*, R^* encode the **thermodynamic limit**.

Appendix: 2D Row-to-Row — Row as Super-Spin

Idea: Treat one row of L spins as a single "super-spin".

[Figure: Row $\rightarrow$ super-spin with 2^L states]

Row configuration:

$$\vec{\sigma} = (\sigma_1, \sigma_2, \ldots, \sigma_L)$$

Total states: 2^L.

Row-to-row transfer:

$$(T_{\text{row}})_{\vec{\sigma},\vec{\sigma}'} = \prod_{i=1}^{L} e^{\beta J \sigma_i \sigma_i'} \prod_{i=1}^{L-1} e^{\beta J \sigma_i \sigma_{i+1}}$$

$T_{\text{row}} \in \mathbb{R}^{2^L \times 2^L}$.

Key Point

Now, $Z = \text{Tr}(T_{\text{row}}^M)$ looks like 1D problem with 2^L dimensional transfer matrix!

Appendix: Row Transfer Matrix = MPO

Key insight: T_{row} has a **tensor network** structure!

Decompose T_{row} into local tensors:

$$T_{\vec{\sigma},\vec{\sigma}'} = \sum_{\alpha_1,\ldots} W^{\sigma_1\sigma_1'}_{\alpha_0\alpha_1} W^{\sigma_2\sigma_2'}_{\alpha_1\alpha_2} \cdots$$

[Figure: MPO structure of T_{row}]

Local tensor W**:**

$$W^{\sigma\sigma'}_{\alpha\beta} = \text{local Boltzmann weight}$$

- σ, σ': spins in rows n, $n+1$
- α, β: auxiliary (horizontal bonds)

No Quantum!

"MPO" is just a factorization of the classical transfer matrix.

Appendix: Boundary = MPS

MPS: Efficient representation of row configuration space.

Full vector:

$$|R\rangle = \sum_{\vec{\sigma}} R_{\vec{\sigma}} |\vec{\sigma}\rangle$$

has 2^L components.

MPS compression:

$$R_{\vec{\sigma}} = A^{\sigma_1} A^{\sigma_2} \cdots A^{\sigma_L}$$

Only $L \cdot \chi^2 \cdot 2$ parameters!

[Figure: MPS tensor network]

Physical Meaning

MPS compresses 2^L row configs into χ effective states with **limited entanglement**.

Appendix: Row-to-Row DMRG Algorithm

Combine: MPS (boundary) + MPO (transfer) + SVD (truncation).

1. Initialize MPS $|L\rangle$, $|R\rangle$ for boundaries
2. **Grow:** Apply MPO T_{row} to boundaries

$$|L'\rangle = T_{\text{row}}|L\rangle$$

3. **Truncate:** SVD to compress bond dimension back to χ
4. Iterate until convergence to fixed point $|L^*\rangle$, $|R^*\rangle$

Pros

- Systematic, well-understood
- Controlled approximation

Cons

- Breaks 2D rotational symmetry
- Hard to generalize to other lattices

Appendix: No Monte Carlo Sign Problem

Classical models: All Boltzmann weights are **positive**!

$$W = e^{-\beta H} > 0 \quad \text{always}$$

[Figure: Classical (positive) vs Quantum (sign problem)]

Advantage of Tensor Network Methods

Appendix: Why is 3D Difficult?

In 2D: Environment tensors are **1D objects** (edges).

In 3D: Environment would be **2D surfaces** — back to exponential!

[Figure: 2D boundary problem in 3D systems]

Appendix: Extensions and Generalizations

Beyond square lattice Ising:

Different lattices:

- Honeycomb
- Triangular
- Kagome

Different models:

- Potts model
- Clock model
- Vertex models

Quantum systems (via iPEPS):

- 2D Heisenberg model
- Frustrated magnets
- Topological phases

Improvements:

- Directional CTMRG
- Full-update vs simple-update
- Gradient optimization

Comparison: iDMRG vs CTMRG

Aspect	**iDMRG (1D)**	**CTMRG (2D)**
Dimension	1D chain	2D square lattice
Environment	Left + Right	4 Corners + 4 Edges
Grow step	Add site pair	Add row + column
Truncation	SVD on center bond	SVD on corner boundaries
Bond dimension	χ (MPS)	χ (environment)
Fixed point	L^*, R^*	C_i^*, T_i^*
Computational cost	$O(\chi^3)$	$O(\chi^6)$ or $O(\chi^5)$

Appendix: Computational Complexity

CTMRG scaling:

Operation	**Cost**	**Bottleneck?**
Corner absorption	$O(\chi^4 d^2)$	
Edge absorption	$O(\chi^3 d^2)$	
Build density matrix	$O(\chi^4)$	
SVD for projector	$O(\chi^3 d^3)$	✓
Apply truncation	$O(\chi^3 d)$	
Total per iteration	$O(\chi^3 d^3)$ to $O(\chi^6)$	

Comparison

- iDMRG (1D): $O(\chi^3)$ per sweep
- CTMRG (2D): $O(\chi^5)$ – $O(\chi^6)$ per iteration
- More expensive, but still **polynomial** in χ!

Appendix: Key References

Original works:

- R. J. Baxter, *J. Math. Phys.* **9**, 650 (1968) — Corner transfer matrices
- R. J. Baxter, *J. Stat. Phys.* **19**, 461 (1978) — CTM method

Modern CTMRG:

- T. Nishino & K. Okunishi, *J. Phys. Soc. Jpn.* **65**, 891 (1996)
- T. Nishino & K. Okunishi, *J. Phys. Soc. Jpn.* **66**, 3040 (1997)

CTMRG for iPEPS:

- R. Orús & G. Vidal, *Phys. Rev. B* **80**, 094403 (2009)
- P. Corboz et al., *Phys. Rev. B* **84**, 041108(R) (2011)

Reviews:

- R. Orús, *Ann. Phys.* **349**, 117 (2014) — Tensor networks review

Appendix: Original CTMRG Results — 2D Ising Model

From T. Nishino & K. Okunishi, *J. Phys. Soc. Jpn.* 65, 891 (1996)

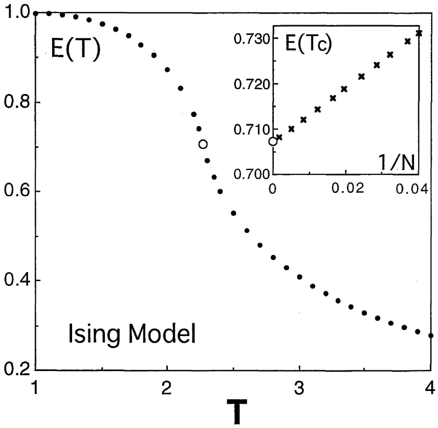


Fig. 5. Local energy $E(T)$ of the Ising model. We determine $E(T_c)$ by observing the convergence with respect to N, which is the number of iterations of the CTMRG procedure.

Appendix: CTMRG on Other 2D Lattices — Ruby Lattice

Extension to non-square lattices

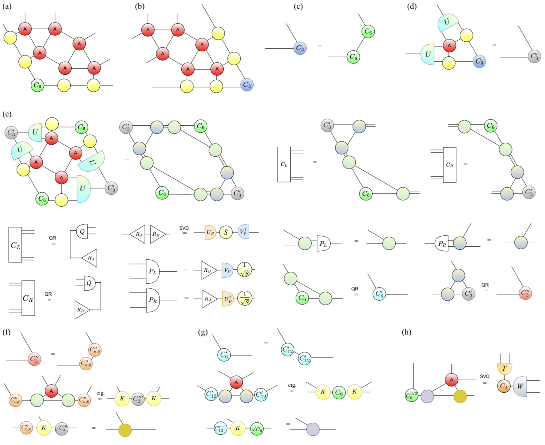


FIG. 19. Ruby lattice: (a) Definition of the corner matrix C_6 inside the hexagon as an intersection point between two bMPS with the angle $\pi/3$. (b) The definition of the corner matrix C_3 inside the hexagon as an intersection point between two bMPS with the angle $2\pi/3$. (c) The consistency condition between the corner matrices C_3 and C_6. (d) The first update step of the