Higher-order tensor renormalization group with the corner transfer matrix

Satoshi Morita (ISSP, UTokyo)

TNSAA 2019-2020 @ National Cheng-Chi University, Taipei, Taiwan

1. Higher-order tensor renormalization group with the corner transfer matrix

2. TeNeS: Tensor Network Solver

Parallelized solver for 2D quantum systems

Satoshi Morita (ISSP, UTokyo)

Outline

1. HOTRG + CTM
   - Real-space renormalization based on tensor networks
   - Review of the higher-order second renormalization group (HOSRG)
   - Environment tensor and corner transfer matrix
   - Benchmark results on 2D Ising model

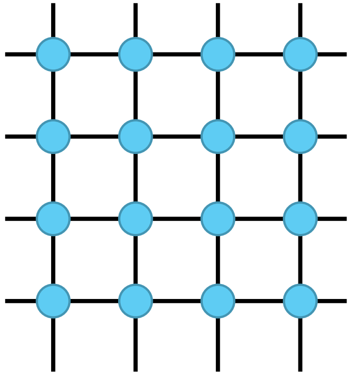
2. TeNeS (Tensor Network Solver)
   - Parallelized solver for 2D quantum lattice system
   - Based on a TePS (PEPS) wave function and the CTM method
   - Simple input files with TOML format

Tensor Networks in Physics

- Lagrangian mechanics
  - Partition function (Path integral)

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

$O(d^N)$ terms

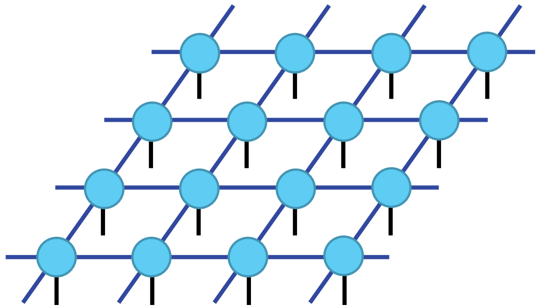


Representation by tensor decomp.

- Hamiltonian mechanics
  - Wave func. of many-body systems

$$|\psi\rangle = \sum_{i_1 \cdots i_N} C_{i_1 \cdots i_N} |i_1 i_2 \cdots i_N\rangle$$

$O(d^N)$ coefficients



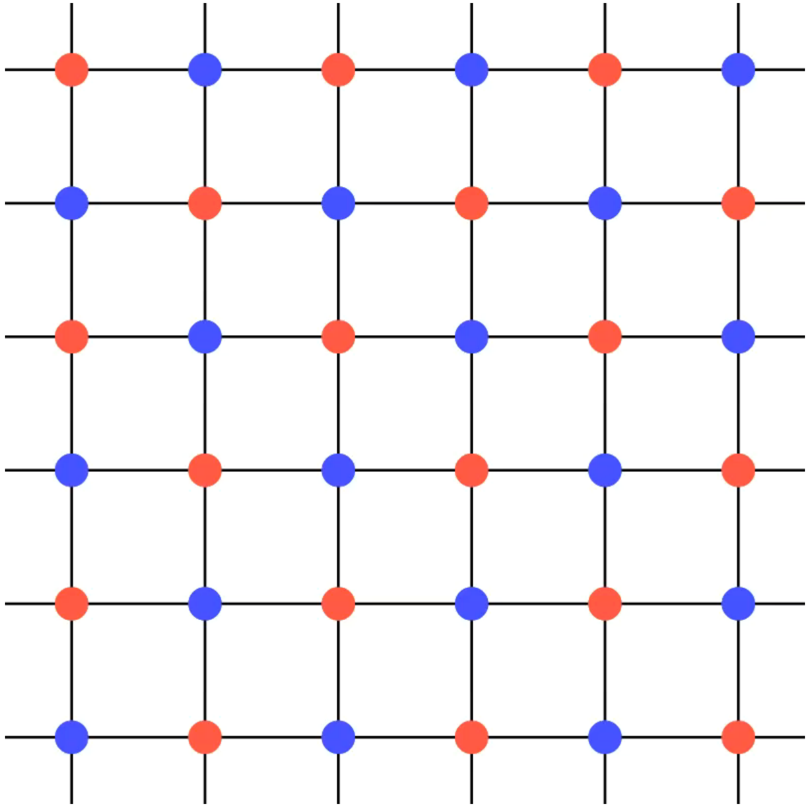
Approx. by tensor decomp.

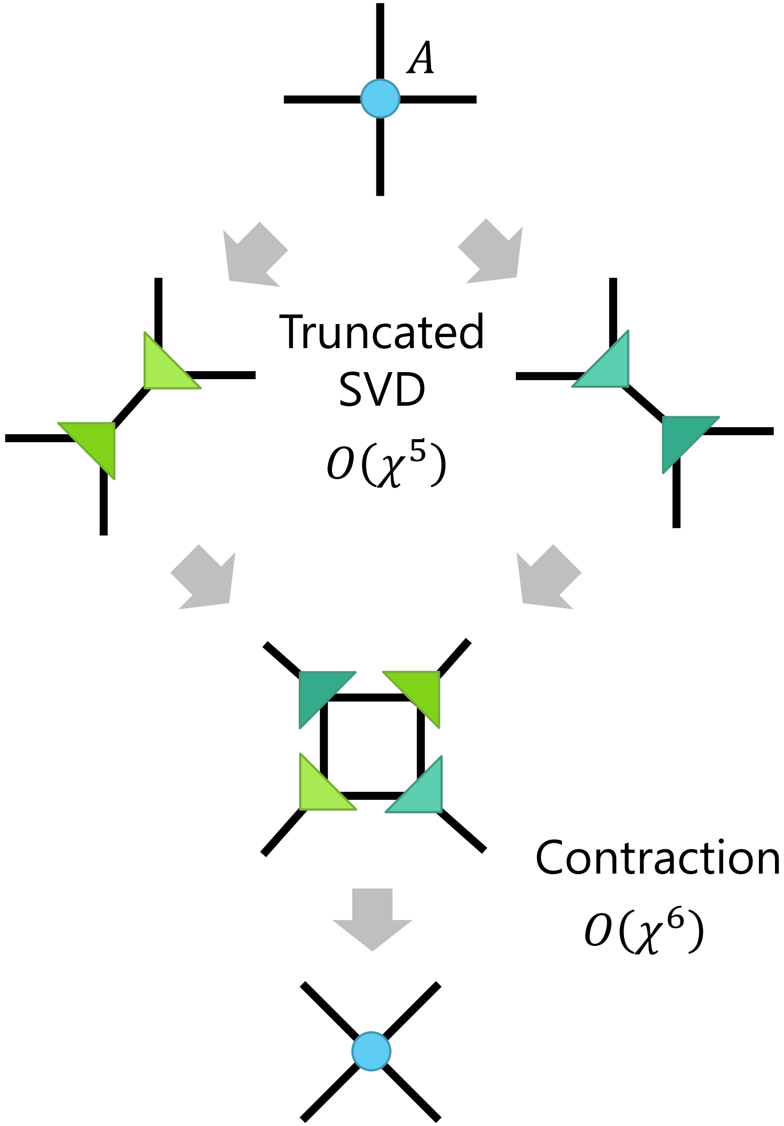
Tensor network representations reduce exponential computational cost to polynomial order.

Real-space renormalization

○ TRG (Tensor Renormalization Group)

Levin, Nave, Phys. Rev. Lett. **99**, 120601 (2007)



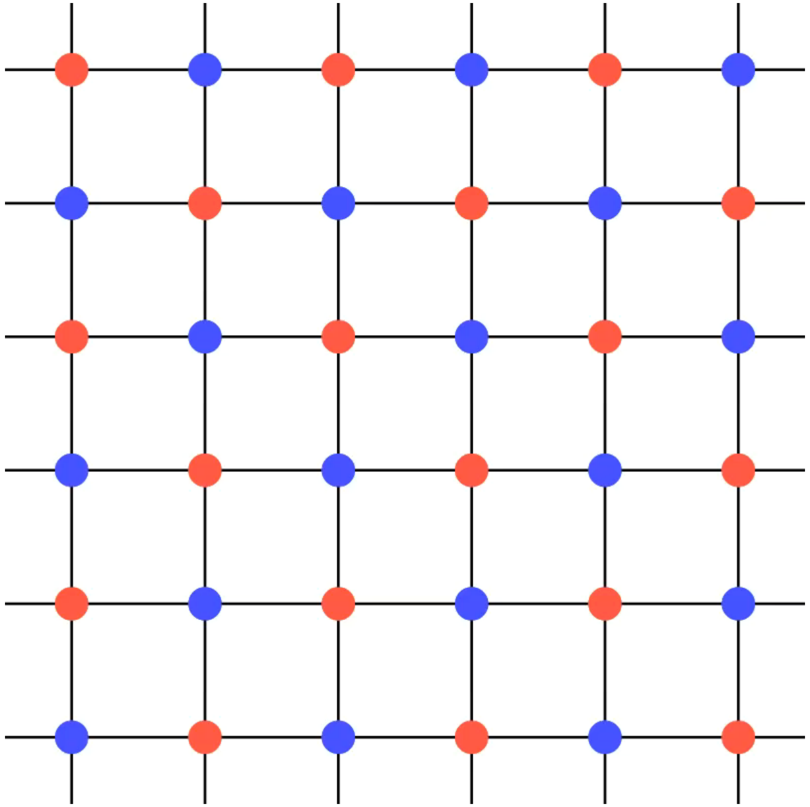


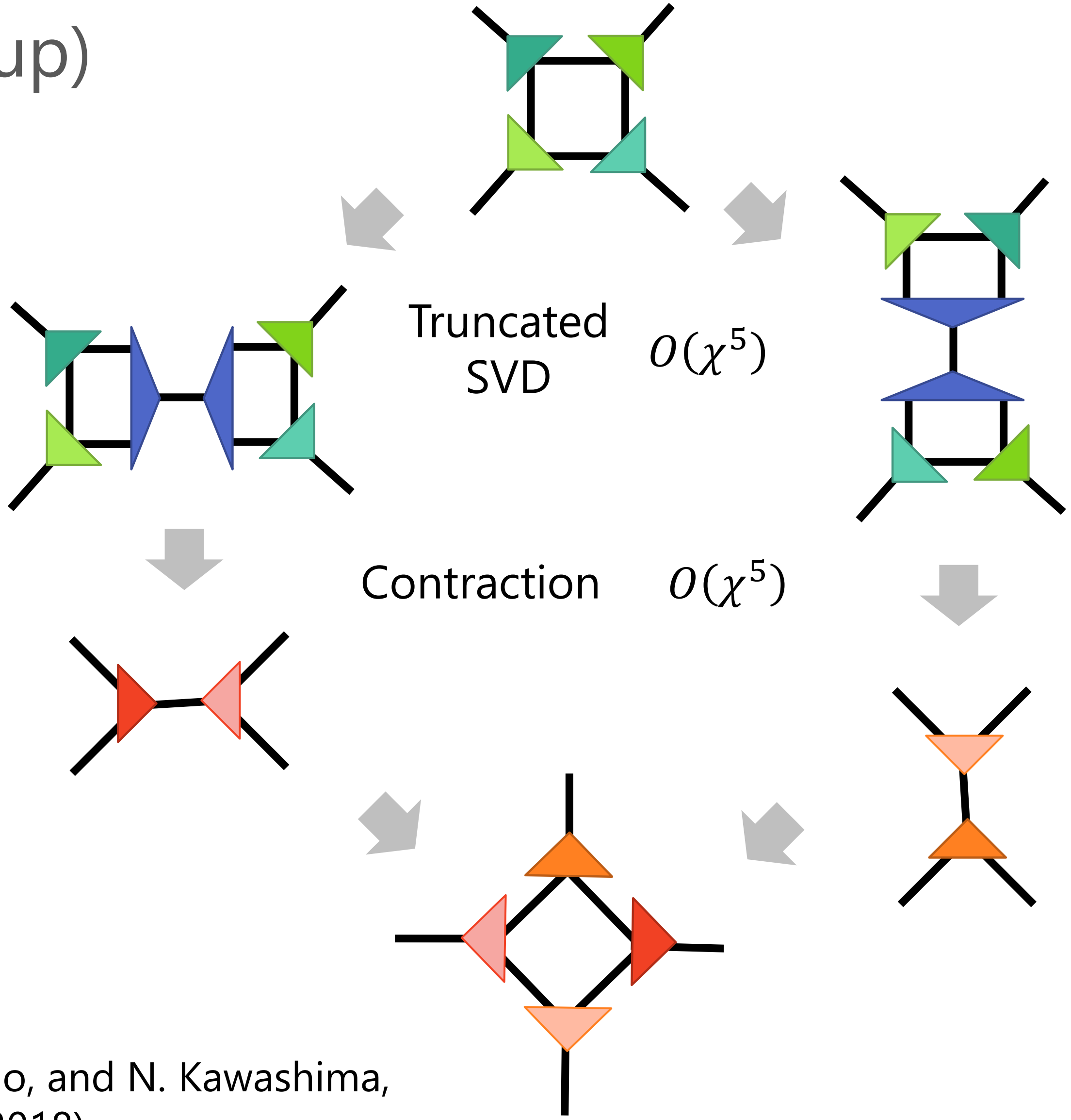
You can download the above movie from
https://smorita.github.io/TN_animation/

Real-space renormalization

- TRG (Tensor Renormalization Group)

Levin, Nave, Phys. Rev. Lett. **99**, 120601 (2007)



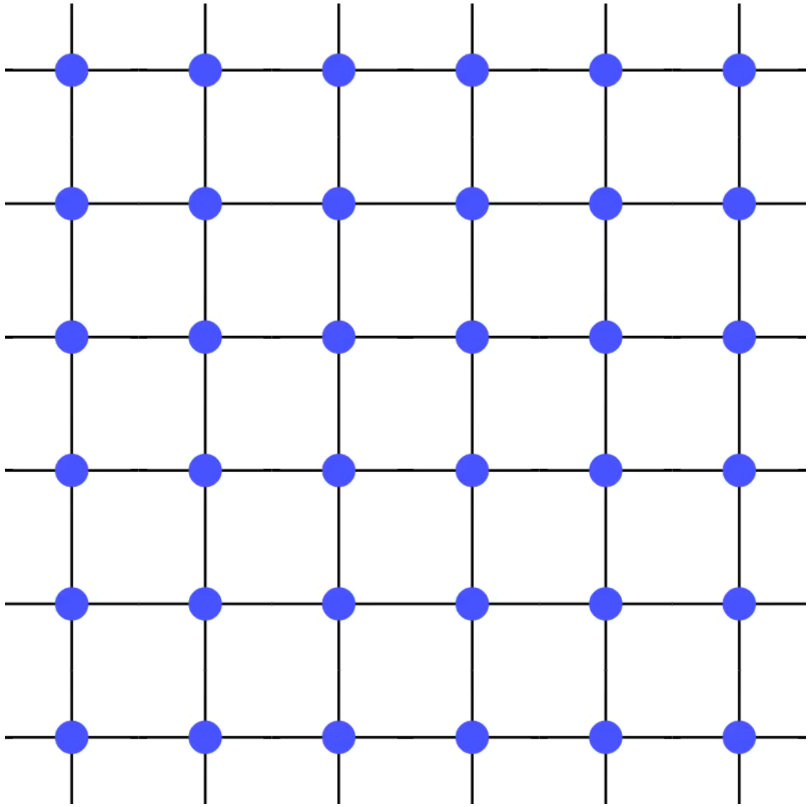


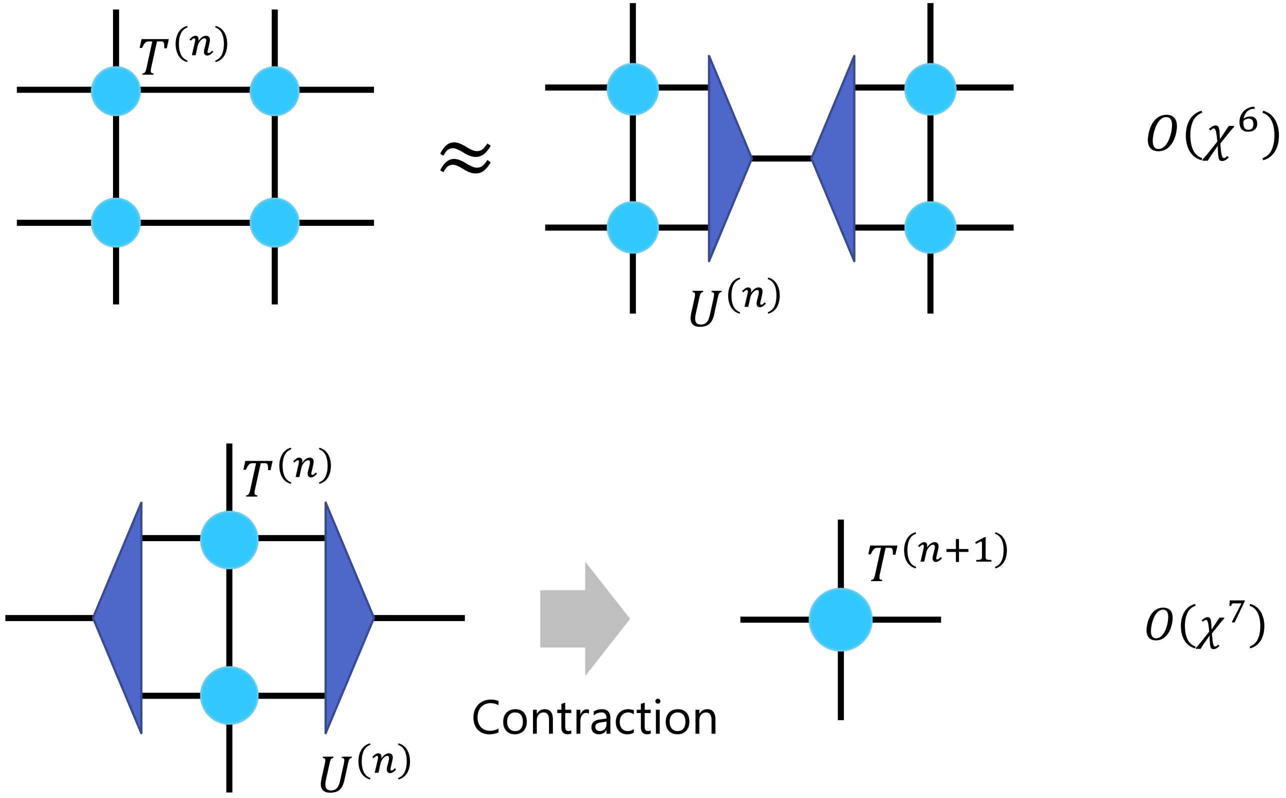
SM, R. Igarashi, H.-H. Zhao, and N. Kawashima,
Phys. Rev. E 97, 033310 (2018)

Real-space renormalization

○ HOTRG (Higher-order Tensor Renormalization Group)

Xie, et al., PRB **86**, 045139 (2012)



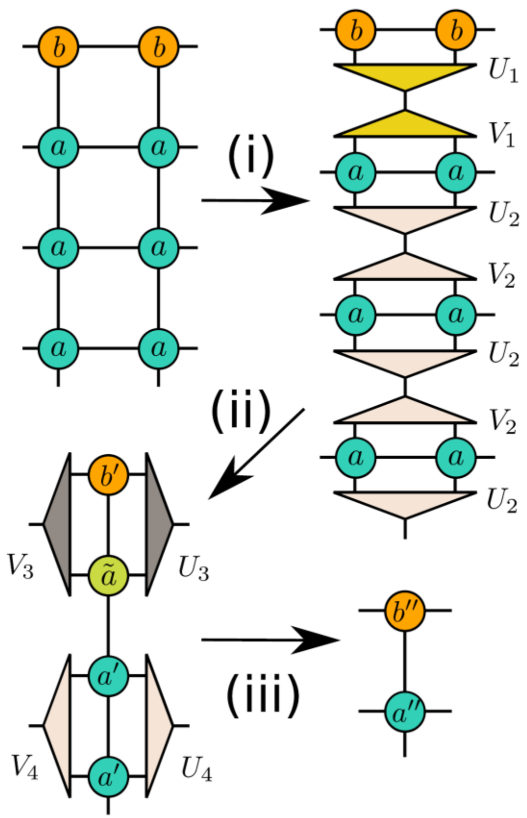


You can download the above movie from
https://smorita.github.io/TN_animation/

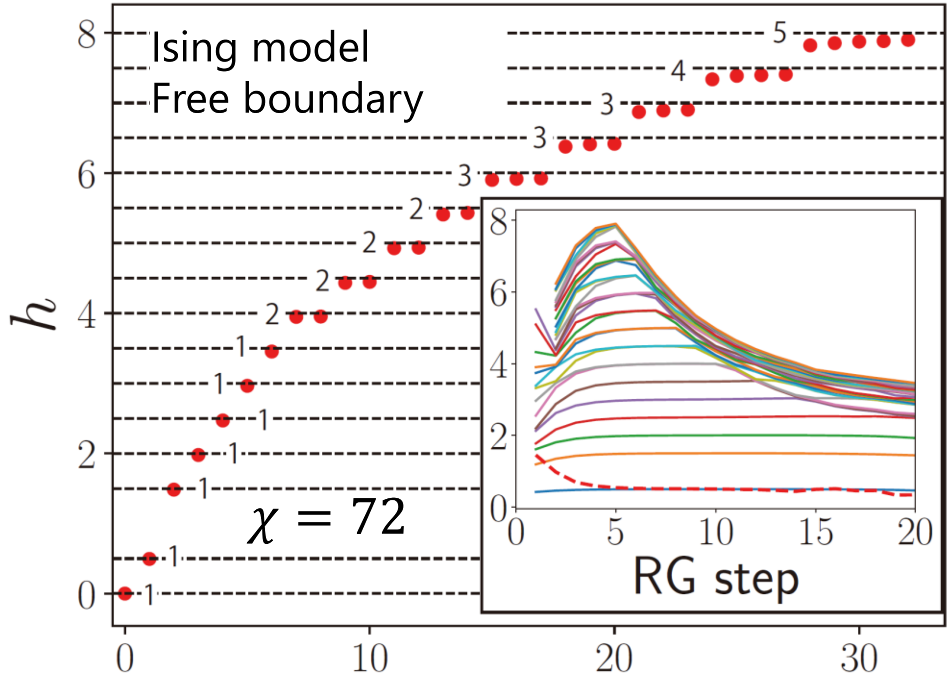
Boundary Tensor Renormalization Group (BTRG)

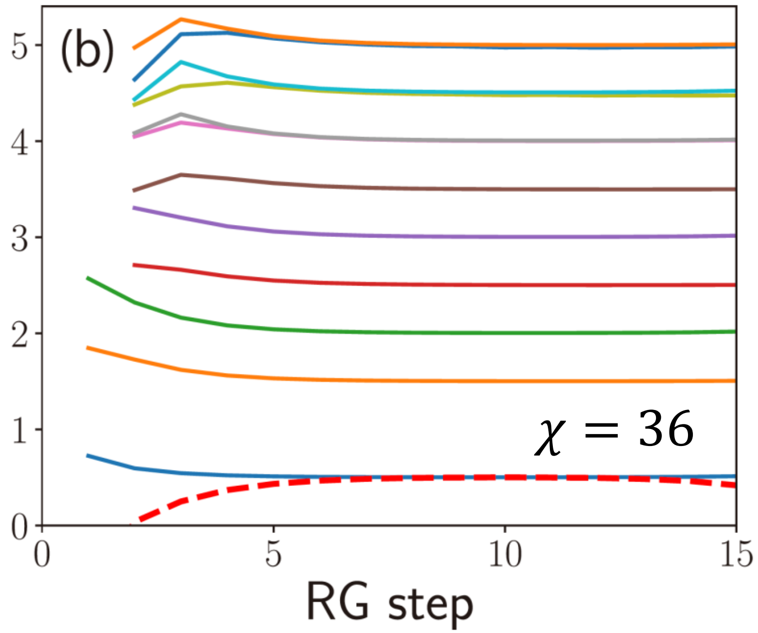
○ Renormalization of boundary tensors

Iino's poster



Scaling dimensions from boundary CFT



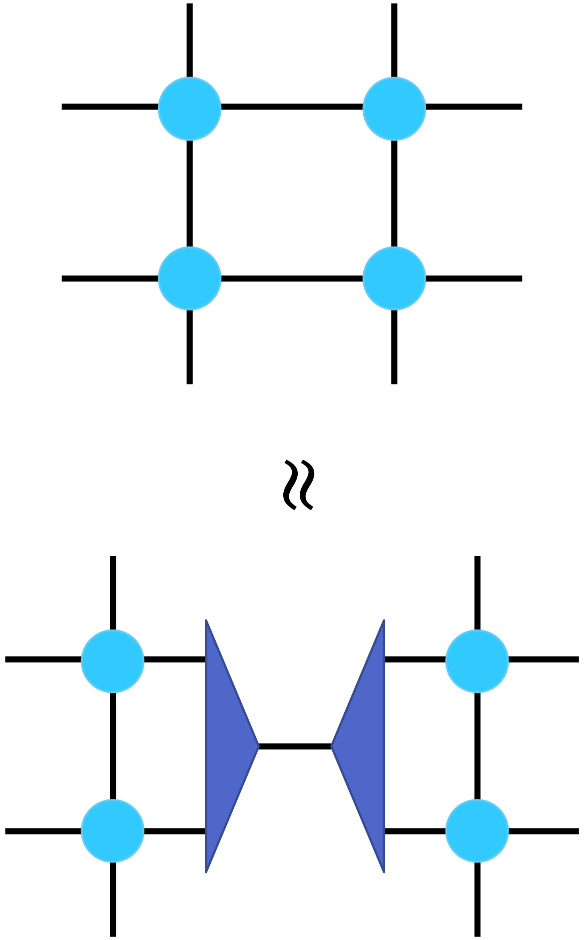


TNR-like algorithm (BTNR)
converges to the true fixed point!

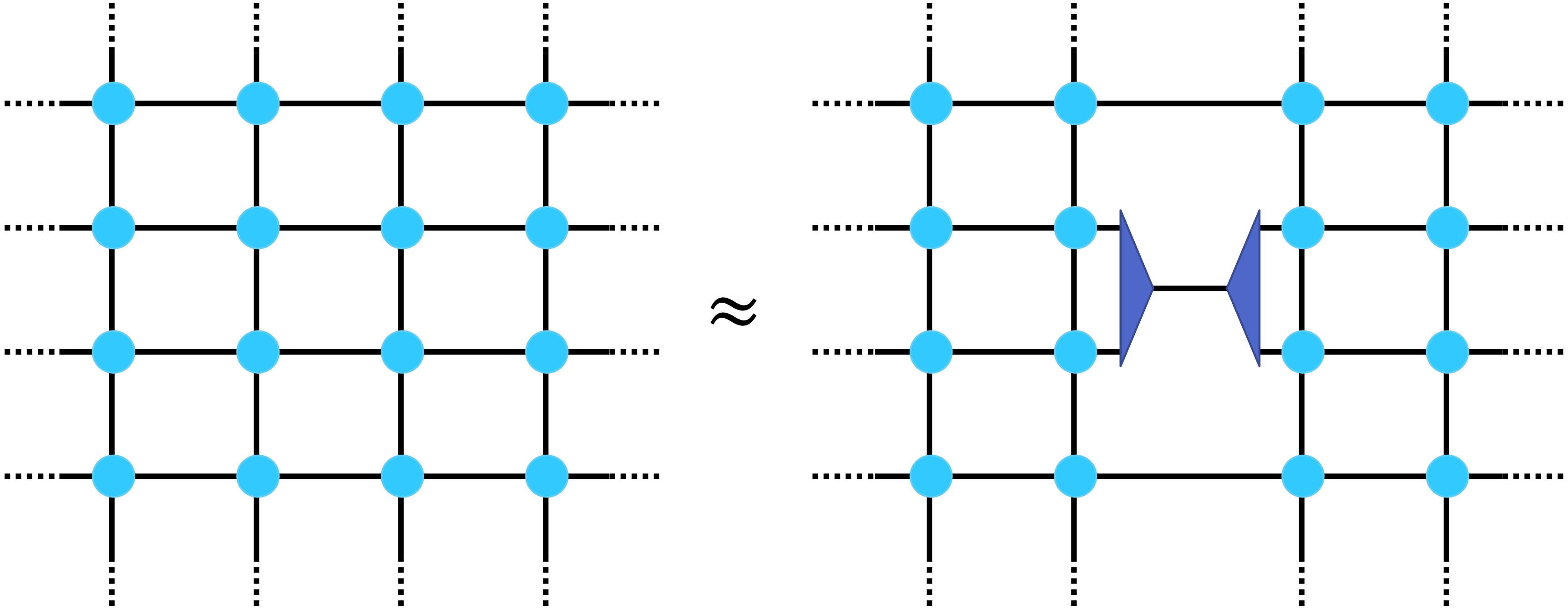
S. Iino, SM. N. Kawashima, Phys. Rev. B **100**, 035449 (2019)
S. Iino, SM. N. Kawashima, arXiv:1911.09907 (2019)

Local vs. Global optimizations

○ Local approx.



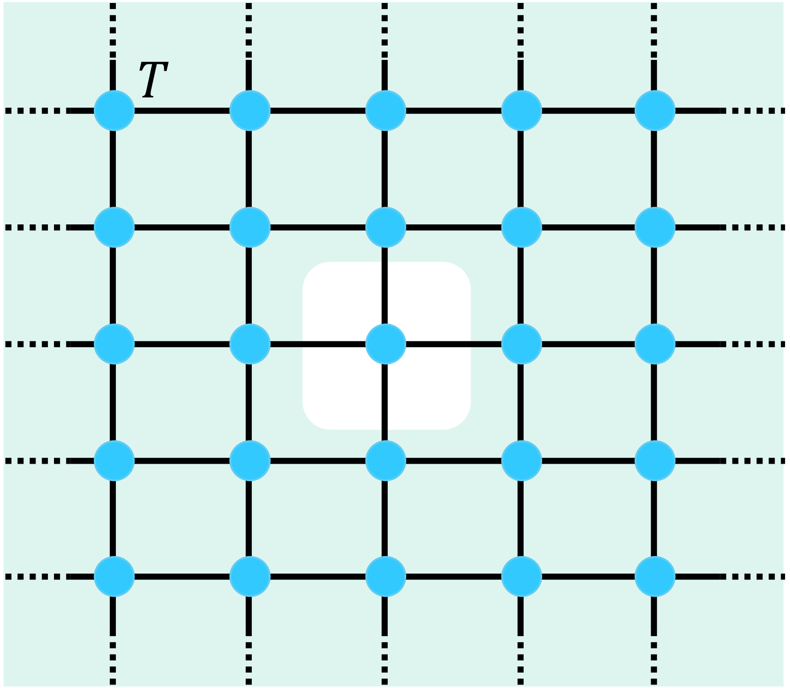
○ Global approx.



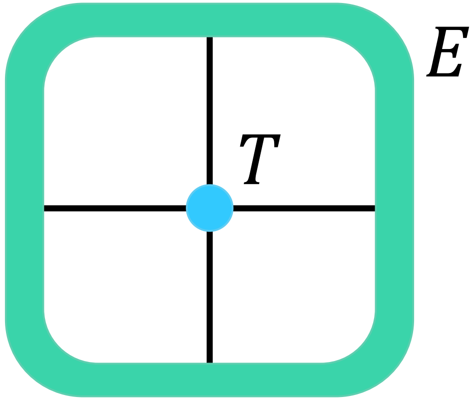
"Second Renormalization Group"

Z.Y. Xie, et al., Phys. Rev. Lett. **103**, 160601 (2009)

Environment tensor



$\approx$

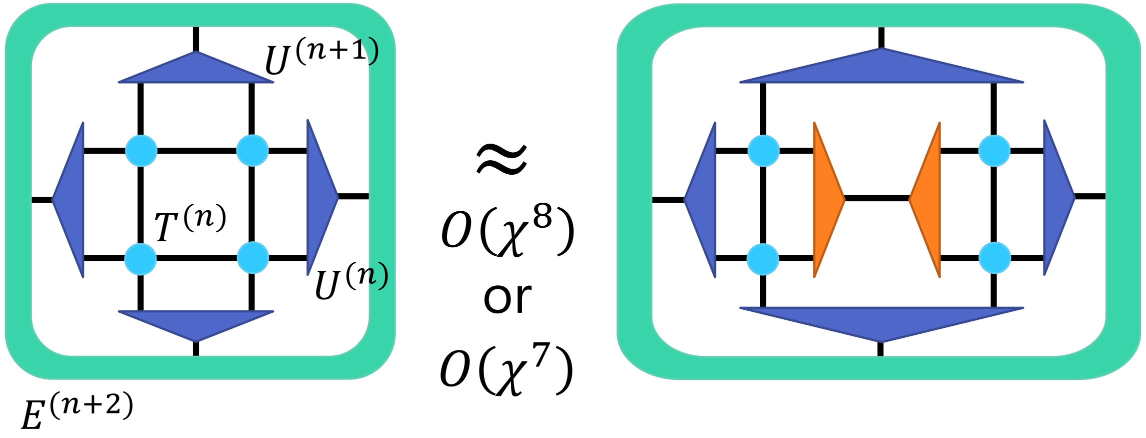


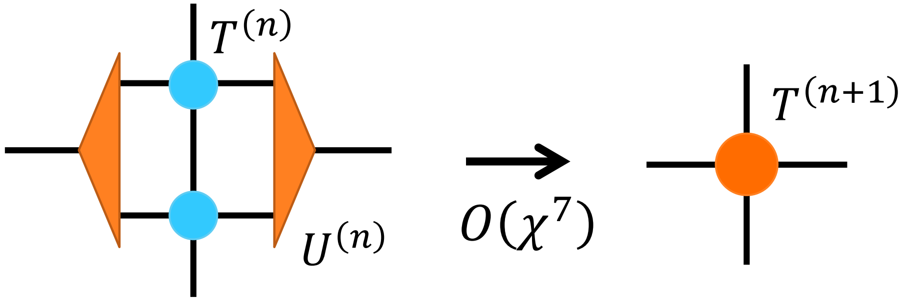
$$Z = \mathrm{Tr}\, T^{\otimes N} \approx \sum_{ijkl} T_{ijkl} E_{ijkl}$$

Xie, et al., PRB **86**, 045139 (2012)

HOSRG: Higher-Order Second Renormalization Group

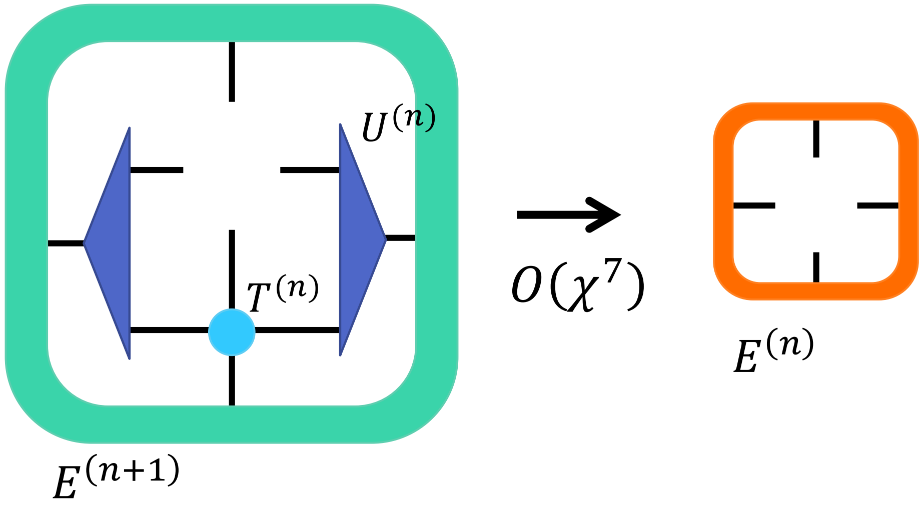
○ Forward iteration





Find the new isometry $U^{(n)}$ from $E^{(n+2)}, U^{(n+1)}, U^{(n)}, T^{(n)}$.
Update $T^{(n+1)}$ from $T^{(n)}$ and $U^{(n)}$ as HOTRG.

○ Backward iteration

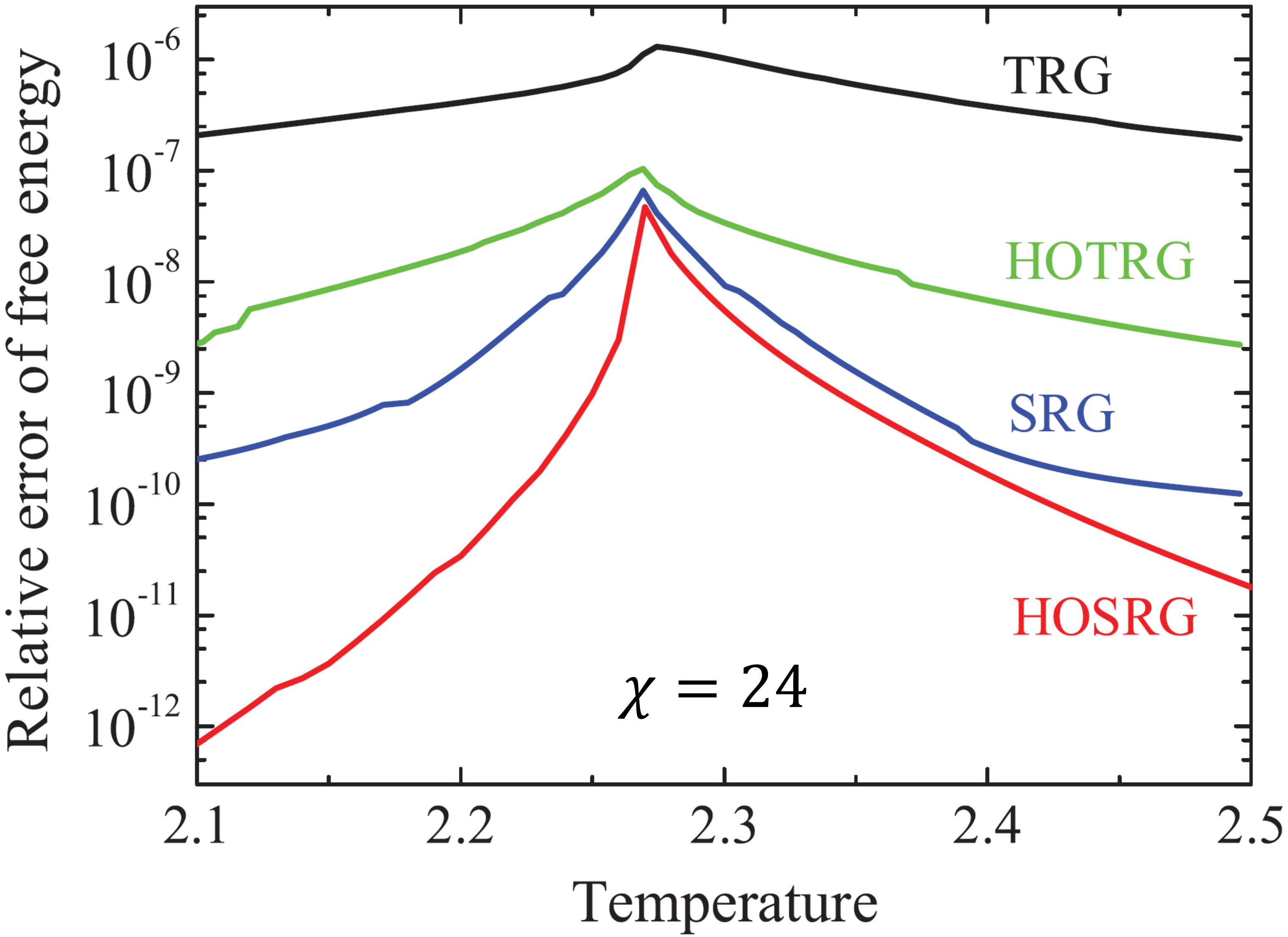


Update the environment $E^{(n)}$
from $E^{(n+1)}, U^{(n)}, T^{(n)}$

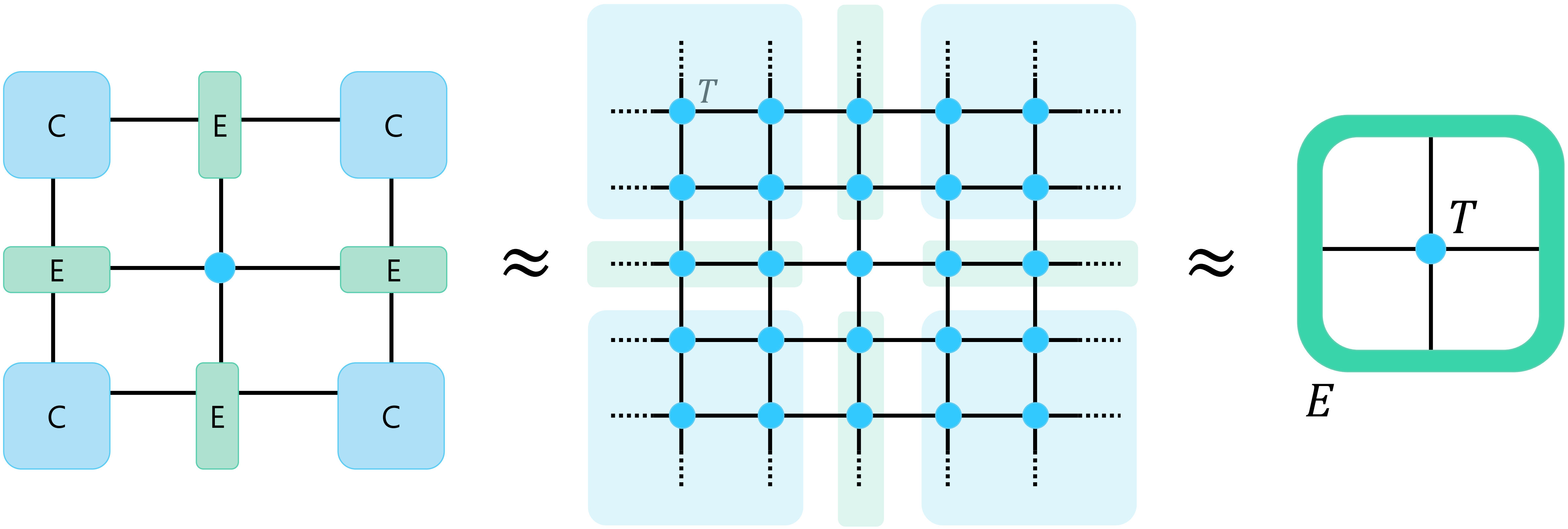
Repeat them until convergence

Benchmark on the 2D Ising model

Xie, et al., PRB **86**, 045139 (2012)

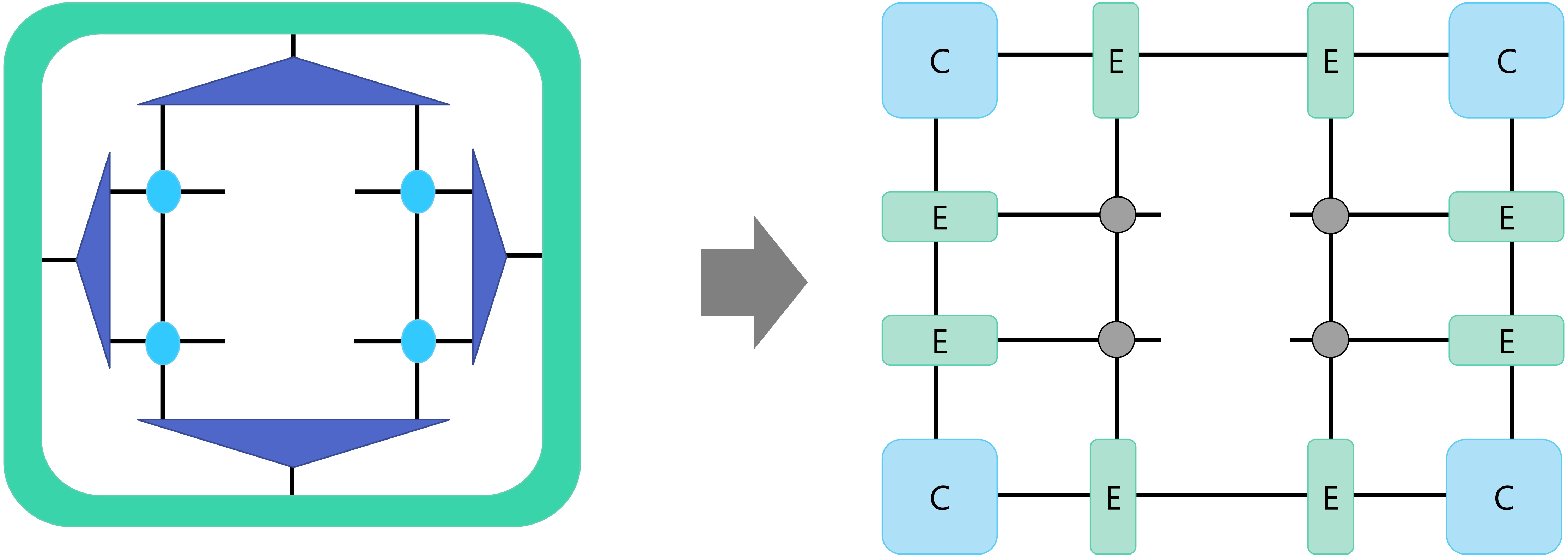


Corner Transfer Matrix (CTM)



CTM: R. J. Baxter, J. Math. Phys. **9**, 650 (1968) 650
CTMRG: T. Nishino, K. Okunishi, J. Phys. Soc. Japan **65**, 891 (1996)

Idea of HOTRG + CTM



- Represent the environment tensor by the corner transfer matrices and the edge tensors.
- The isometry $U^{(n)}$ is calculated by eigenvalue decomposition of the bond density matrix.
- Cost of contraction: $O(\chi^8) \rightarrow O(\chi^6)$

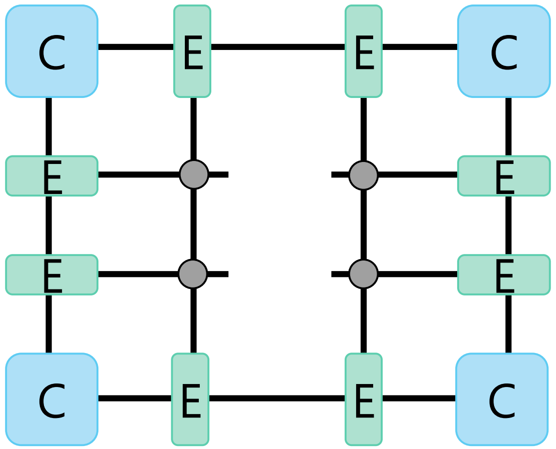
Algorithm of HOTRG+CTM

1. Update $C^{(n)}$ and $E^{(n)}$ using CTMRG
2. Calculate $\rho^{(n)}$
3. Calculate $U^{(n)}$ from $\rho^{(n)}$

$$\rho^{(n)} = U^{(n)}\Lambda^{(n)}U^{(n)\dagger}$$

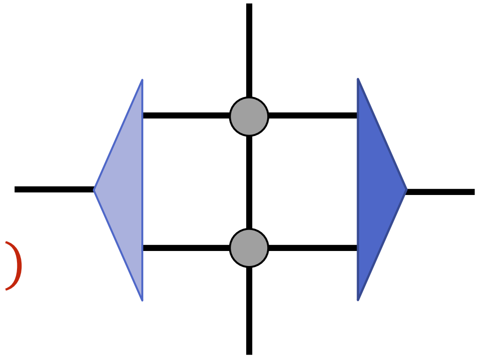
4. Calculate $T^{(n+1)}$ from $T^{(n)}$, $U^{(n)}$
5. Calculate $E^{(n+1)}$ from $E^{(n)}$,$U^{(n)}$
6. Set $C^{(n+1)} = C^{(n)}$
7. Swap x and y axes

$\rho^{(n)} \equiv$

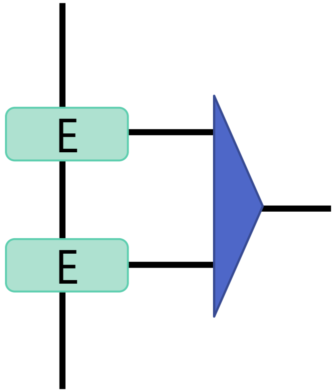


$T^{(n+1)} =$

$O(\chi^7)$



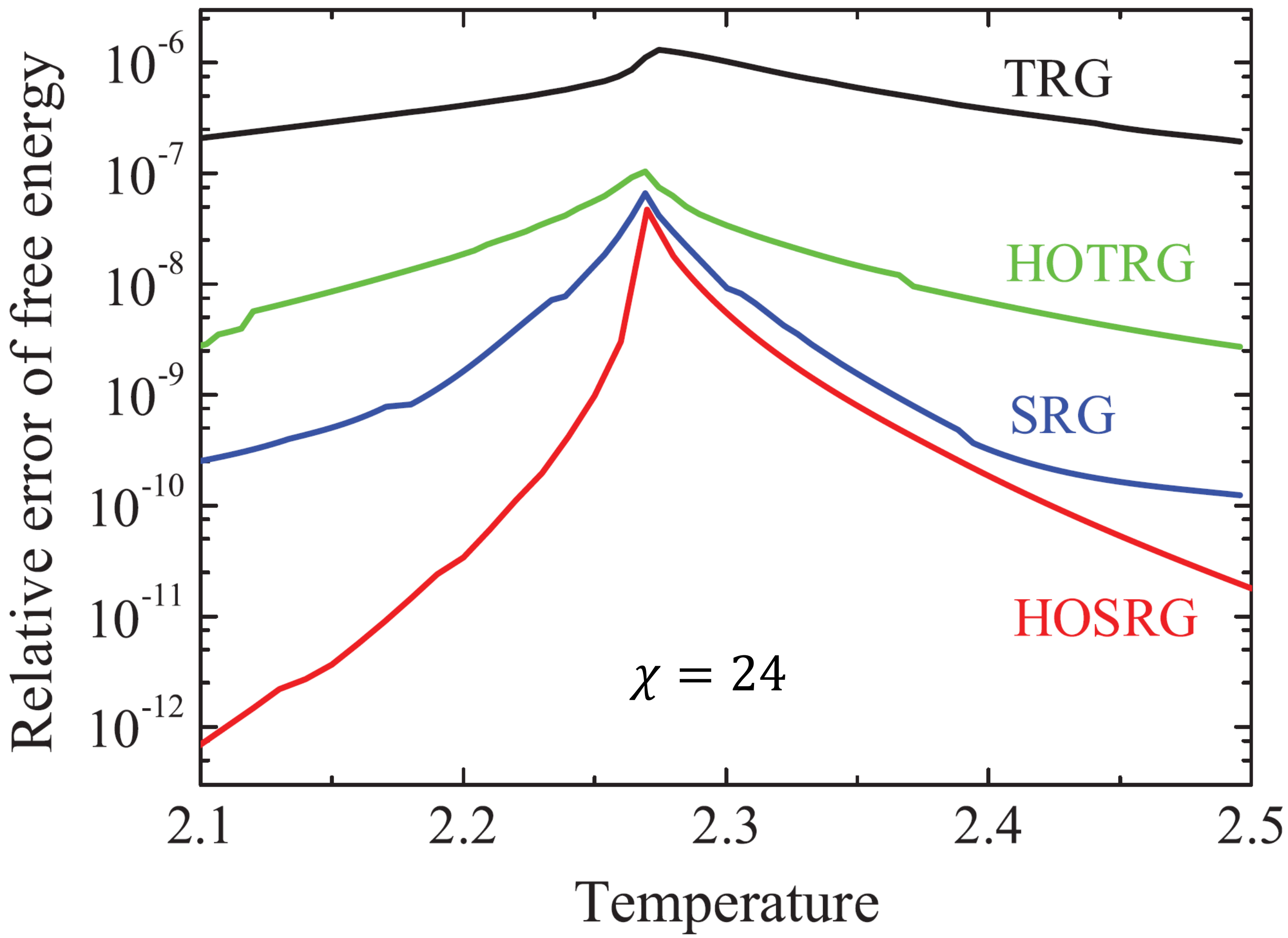
$E^{(n+1)} =$

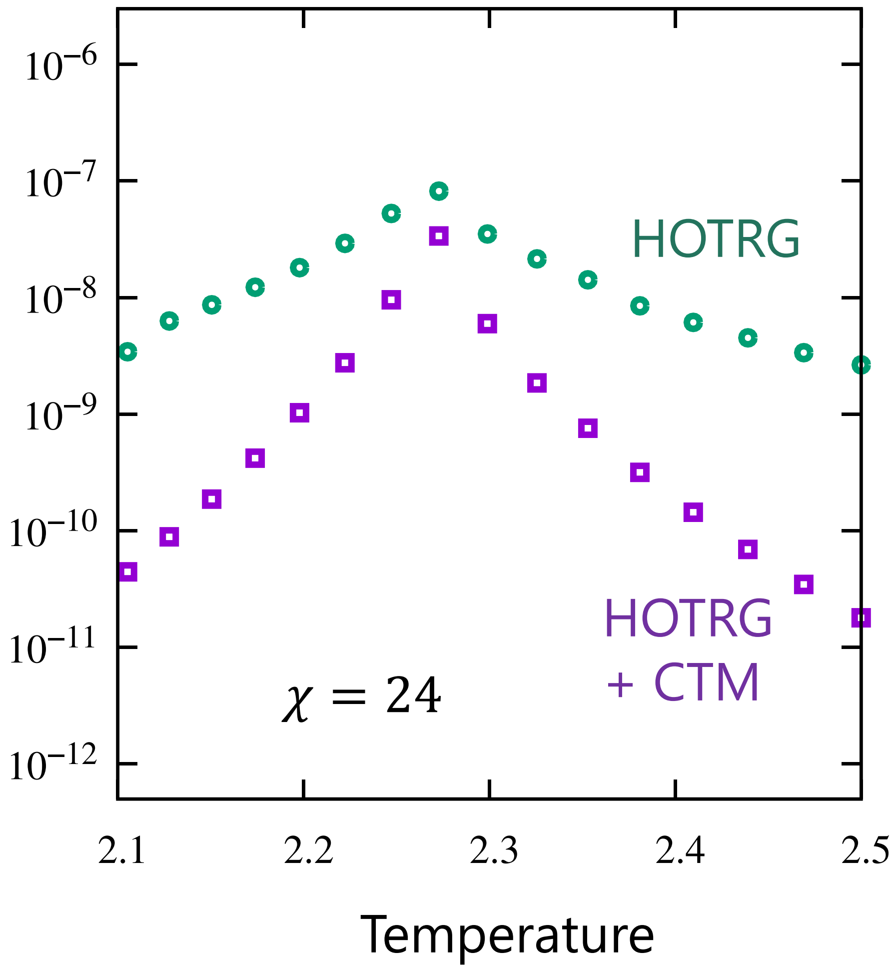


No backward iteration

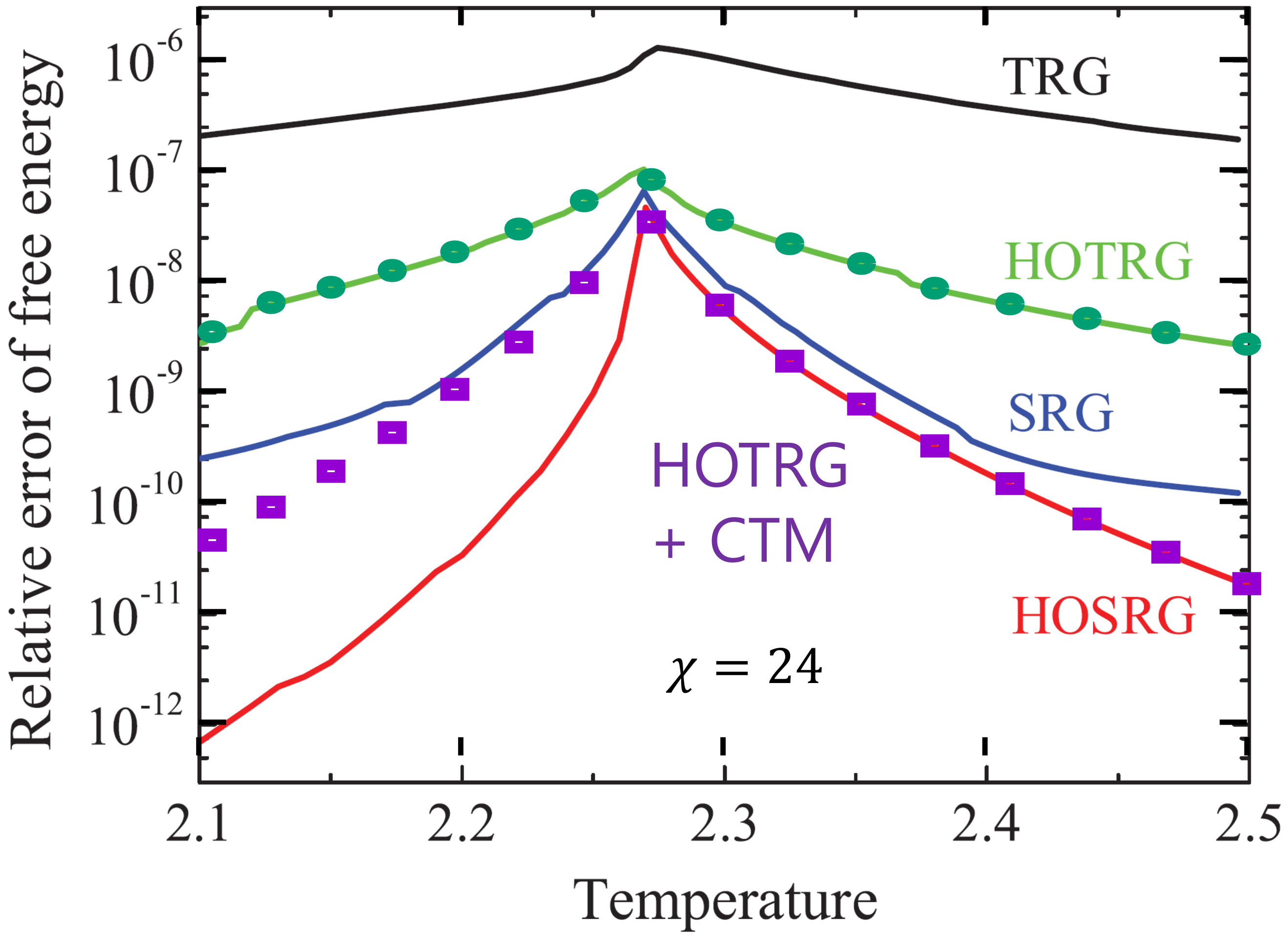
Benchmark on the 2D Ising model

Xie, et al., PRB **86**, 045139 (2012)



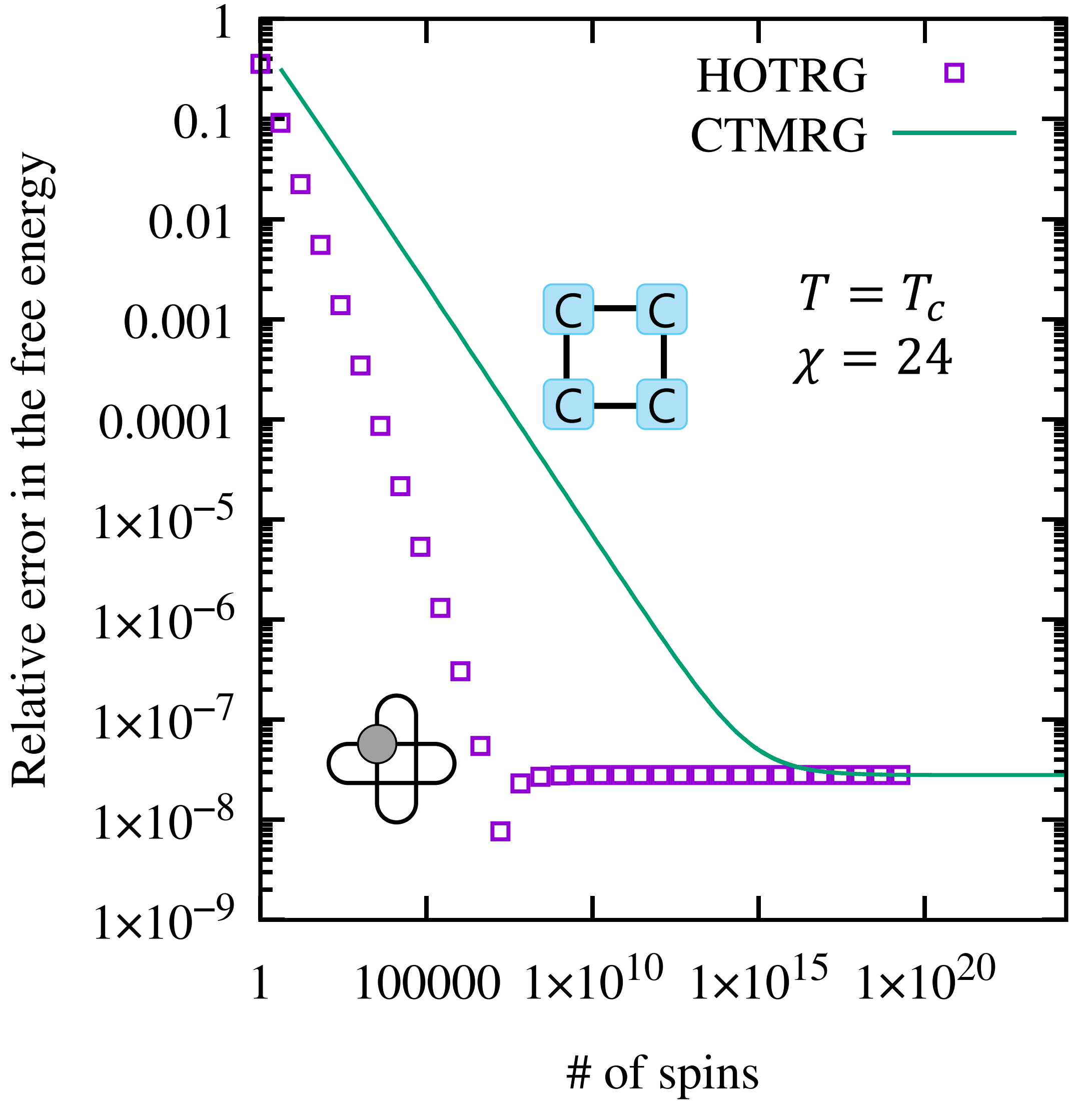


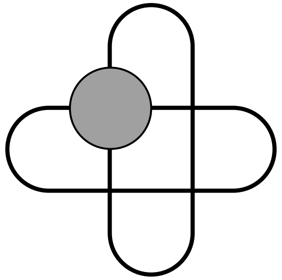
Benchmark on the 2D Ising model



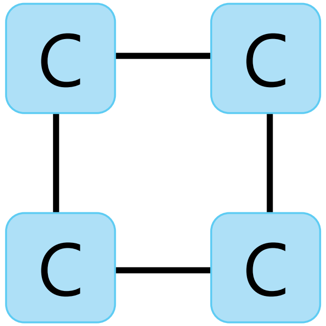
CTM does not converge to the all-up state in the ordered phase, since we use Z2 symmetric tensor.

Convergence of the free energy



$Z_{\mathrm{HOTRG}} =$ 

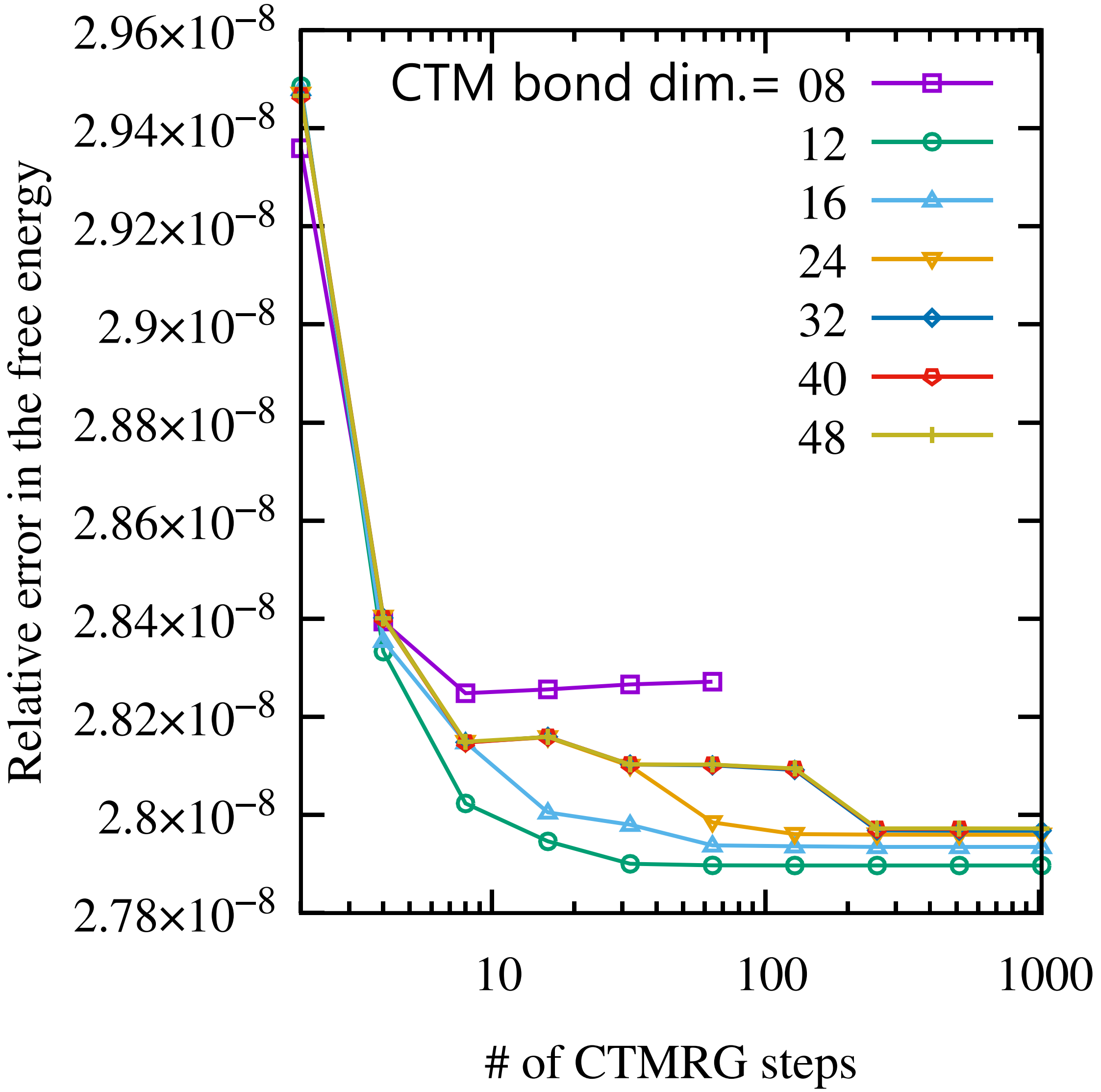
Periodic boundary condition

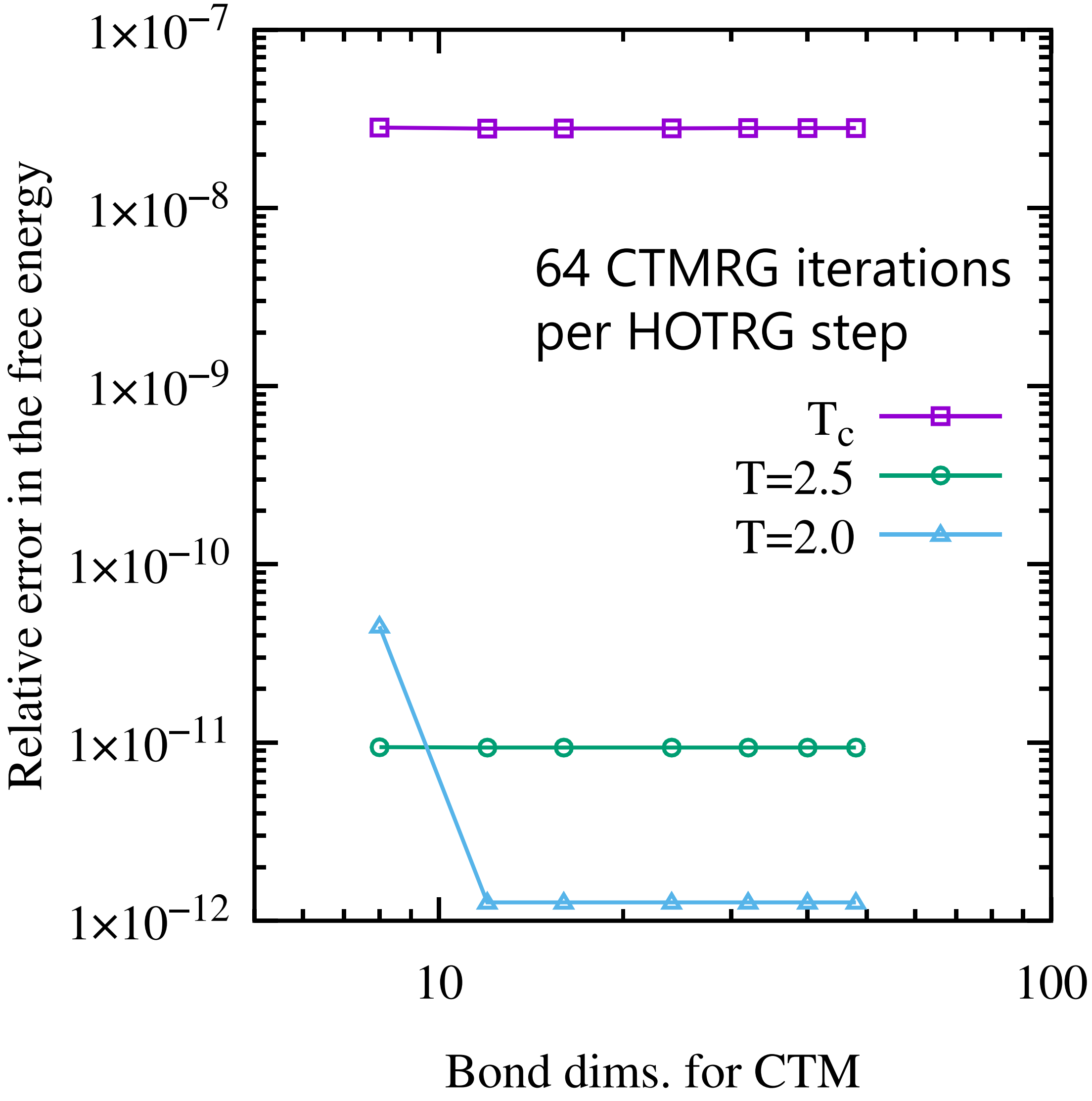
$Z_{\mathrm{CTM}} =$ 

Free boundary condition

Dependence on CTMRG parameters

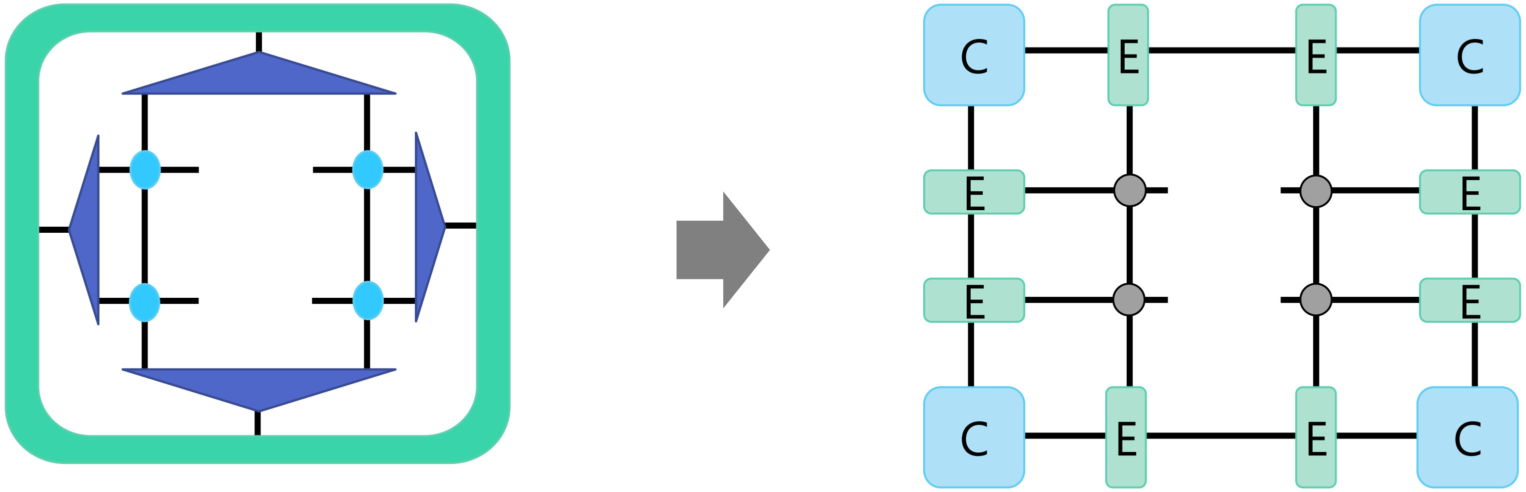
($\chi = 24$ for HOTRG, $L = 2^{24}$)





Short summary of 1st part

- ○ Improvement of HOSRG by using CTM
  - ➢ Replace the environment tensor in HOSRG with CTMs and edge tensors
  - ➢ Computational cost scales as the same as HOTRG
  - ➢ Small iterations of CTMRG is enough to obtain the same results as HOSRG
    - Backward iteration is not necessary



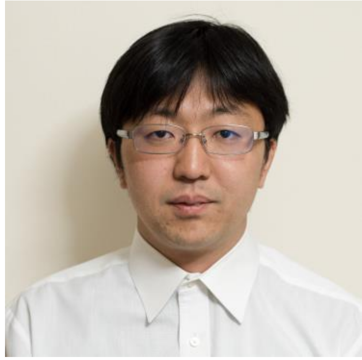
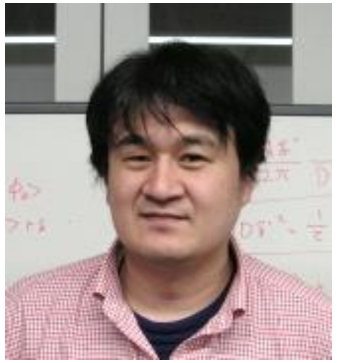
TeNeS: Tensor Network Solver

Massively parallel tensor network for 2D quantum lattice systems
based on a TPS (PEPS) wave function and the CTM method

https://github.com/issp-center-dev/TeNeS

github TeNeS

Developers

					
T. Okubo (UTokyo)	S. Morita (ISSP)	Y. Motoyama (ISSP)	K. Yoshimi (ISSP)	T. Kato (ISSP)	N. Kawashima (ISSP)

- Support
  - Post-K projects
    - CBSM2(Frontiers of Basic Science: Challenging the Limits)
    - CDMSI (Creation of New Functional Devices and High-Performance Materials to Support Next-Generation Industries)
  - PASUMS, ISSP
    - "Project for advancement of software usability in materials science"

CBSM²
Challenge of Basic Science
– Exploring Extremes through
Multi-Physics and
Multi-Scale Simulations

CDMSI

PASUMS
Project for advancement of
software usability in materials science

Softwares for Tensor Networks

- Script language
  - Python + Numpy, Scipy, etc.
  - Julia
  - MATLAB

https://www.tensors.net/
By G. Evenbly

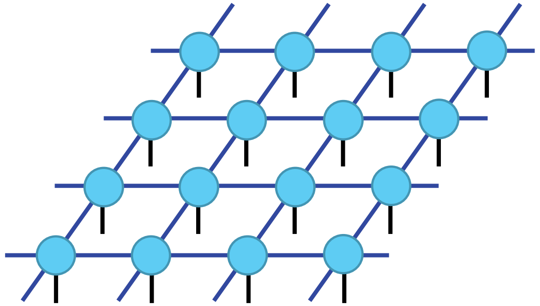
- Applications
  - Uni10
  - iTensor
  - Tensor Network Theory
  - TeNPy

These application does not support parallel calculations on distributed memory.

Parallelization of TN methods

- Huge computational cost and memory usage

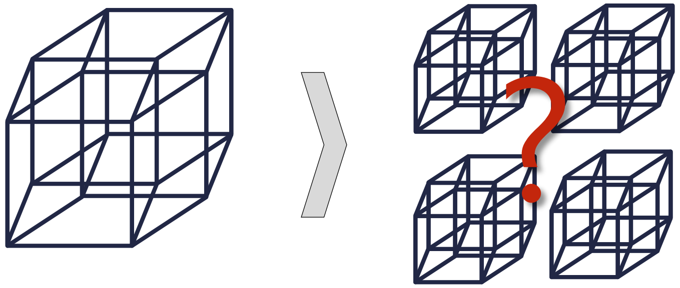
2D PEPS: CPU D^{10}
Memory D^{8}



[Memory]
D=10 : 80 MB
D=20 : 200 GB
D=30 : 5 TB
D=40 : 50 TB

ISSP Supercom.
128 GB / node

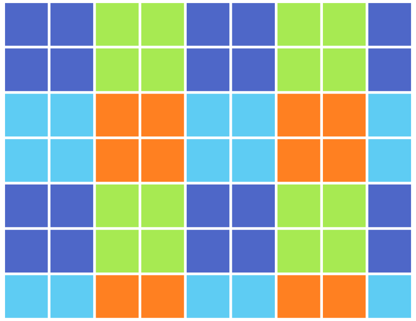
- Problems in parallel library of TN methods
  - How do we distribute tensor elements?
  - How do we design interfaces?
  - What operations do we need?



"mptensor" : Parallel Library for TN methods

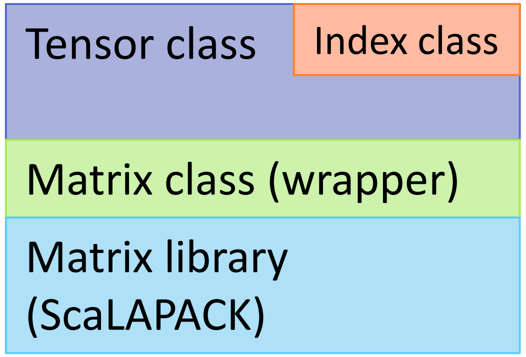
https://github.com/smorita/mptensor

- Tensors on distributed memory
  - Store local elements in the form of distributed matrix
    - Regard a tensor as a matrix. $T_{ijkl} \rightarrow T_{(ij)(kl)}$
    - Use ScaLAPACK for parallel linear algebra libraries
    - Block-cyclic distribution

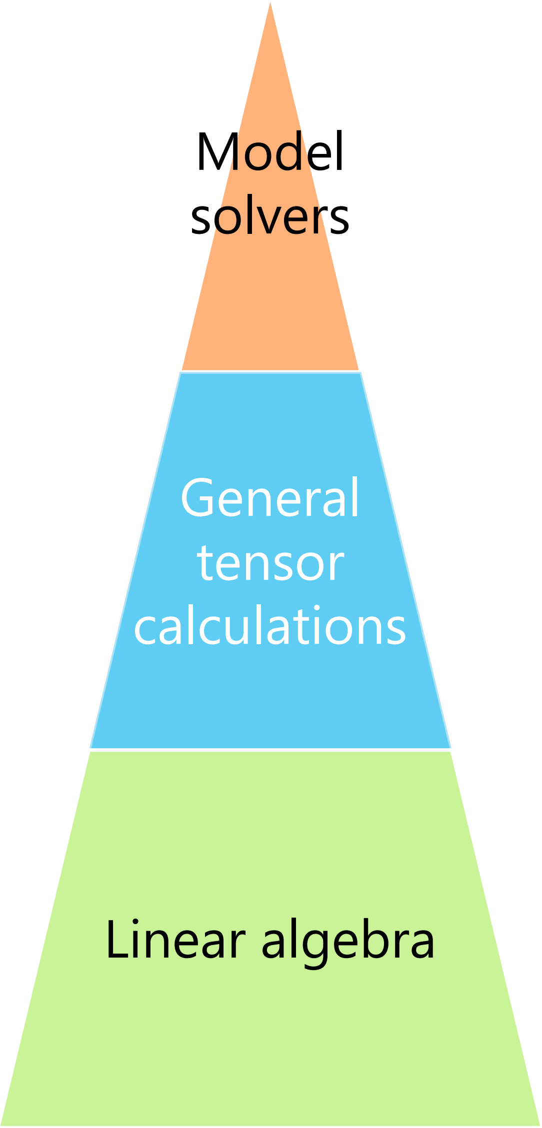


  - Programming language
    - C++98 (some supercomputers do not support C++11, C++14)
    - Hybrid parallelization: MPI + OpenMP
  - Numpy-like interface
    - Easily convert from Python test code

```
       A = transpose(A, Axes(1,3,2,0));
Numpy: A = np.transpose(A, [1,3,2,0])
```



Hierarchy of computation library for TN



Algorithms of TN methods

Ex) PEPS, MERA, TRG, TNR

TeNeS

Operations commonly used in TN methods

Ex) Tensor contraction, Tensor decomposition

mptensor

Matrix operations

Ex) Matrix-matrix multiplication, SVD, QR

Libraries: BLAS, LAPACK, ScaLAPACK, Eigen

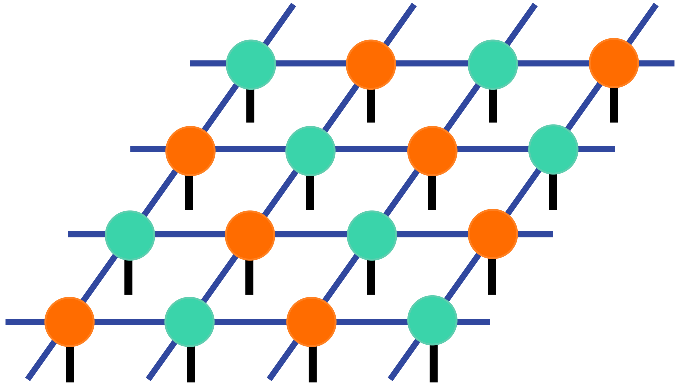
TeNeS: Tensor Network Solver

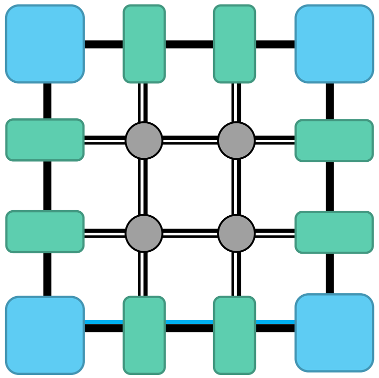
https://github.com/issp-center-dev/TeNeS

- An open-source program package for calculation of many-body quantum states base on the tensor network method
  - 2D quantum spin systems
  - Parallelized based on "mptensor"
  - Use TOML for input-file format

TOML: Tom's Obvious, Minimal Language
https://github.com/toml-lang/toml

- Method
  - TPS (PEPS) + CTM
    - Simple update
    - Full update





TeNeS v0.1 was released yesterday!

Install of TeNeS

Prerequisites

- C++11 compiler
- CMake (>=2.8.14)
- MPI and ScaLAPACK
- Python & toml module

These libraries are automatically downloaded.

- mptensor
- cpptoml
- sanitizers-cmake

Install

- Download from github

https://github.com/issp-center-dev/TeNeS

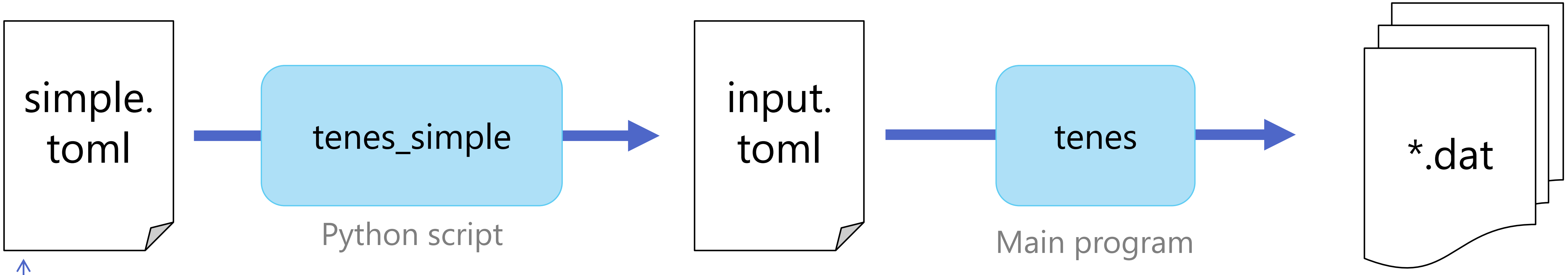
- Build using CMake

```
$ mkdir build
$ cd build
$ cmake ../
$ make
```

License

- GNU GPL v3

Usage of v0.1



- Parameter
- Lattice
  - square or honeycomb
  - unit-cell size
- Model
  - S=1/2 Spin systems
- Correlation
  - $C(r) = \langle A(0)B(r)\rangle$

$$\mathcal{H} = \sum_{\langle ij\rangle}\left[\sum_{\alpha}^{x,y,z} J_{ij}^{\alpha} S_i^{\alpha} S_j^{\alpha} + B\left(\vec{S}_i \cdot \vec{S}_j\right)^2\right] - \sum_i \left[h S_i^z + \Gamma S_i^x - D\left(S_i^z\right)^2\right]$$

parameter.dat
energy.dat
site_obs.dat
neighbor_obs.dat
correlation.dat
time.dat

Example of an input file for "tenes_simple"

- Transverse field Ising model

```
[parameter.tensor]
D = 2
CHI = 10

[parameter.simple_update]
num_step = 1000
tau = 0.01

[parameter.full_update]
num_step = 0
tau = 0.01

[parameter.ctm]
iteration_max = 10
```

```
[lattice]
type = "square lattice"
L_sub = [ 2, 2,]

[model]
type = "spin"
Jz = -1.0
Jx = 0.0
Jy = 0.0
G = 1.0
```

Only 20 lines!

Output to stdout

```
Energy = -0.757303161476
Local operator 0 = 0.297854801816
Local operator 1 = 0.386031967038
```

$\langle S_z \rangle$ (Local operator 0)

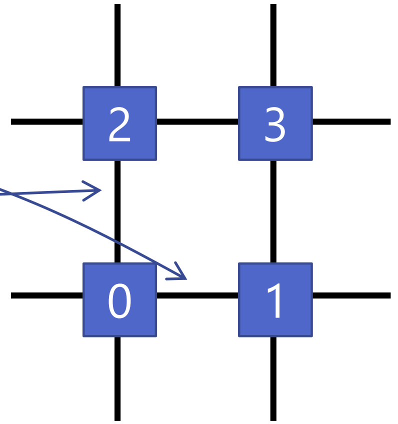
$\langle S_x \rangle$ (Local operator 1)

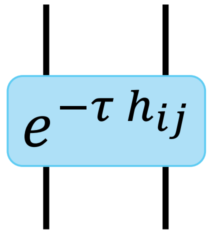
Input file for main program "tenes"

- [parameter]
  - tensor
  - simple_update
  - full_update
  - ctm
  - random
- [lattice]
- [evolution]
- [observable]
  - Sz, Sx, etc.
- [correlation]

```
[evolution]
simple_update = """
0 1 h 0
3 2 h 0
2 3 h 0
1 0 h 0
0 2 v 0
3 1 v 0
2 0 v 0
1 3 v 0
"""

matrix = [
"""
0.9975031223974601 0.0 0.0 0.0
0.0 1.0025156589209967 -0.005012536523536887 0.0
0.0 -0.005012536523536888 1.0025156589209967 0.0
0.0 0.0 0.0 0.9975031223974601
"""
]
```





In "tenes_simple", imaginary-time evolution ops. are automatically calculated.

Summary of 2nd part

https://github.com/issp-center-dev/TeNeS

- Development of TeNeS
  - Ver. 0.1 was released yesterday!
  - Lattice solver for quantum many-body systems
  - PEPS + CTM, simple- & full-update
  - Parallelized by "mptensor" (MPI+OpenMP)
  - Simple input files with TOML format
- Future plan
  - Other models: spin-S systems, bosonic systems
  - Other lattice: Kagome, triangular lattices
  - Long-range interactions
  - Variational optimization

github TeNeS

Your pull requests and comments are welcome!