AD-BC-VUMPS

Contents

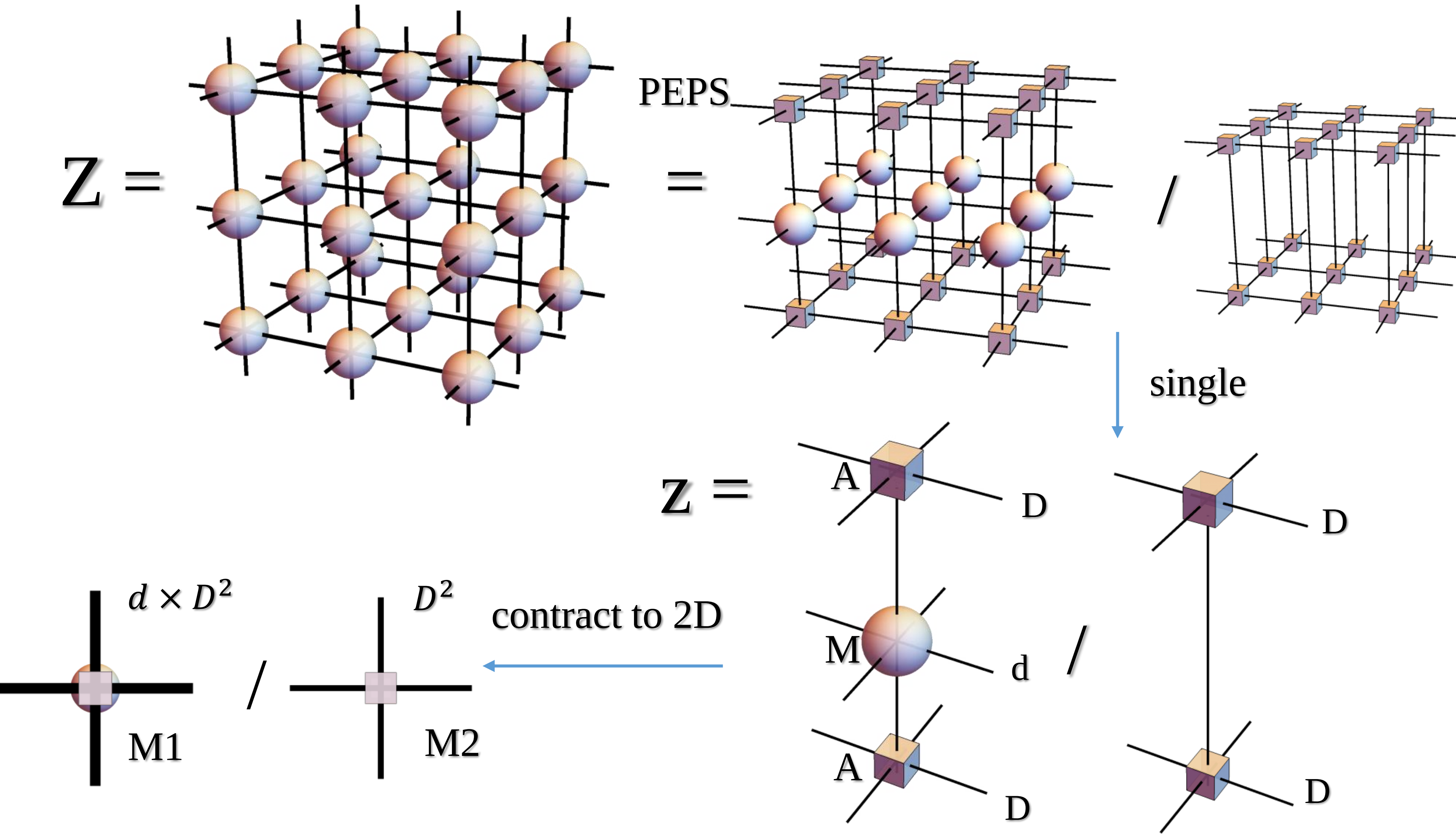
Why AD? → AD-VUMPS

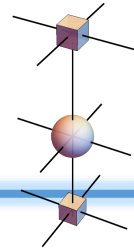
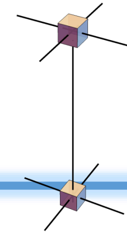
↓ big cell

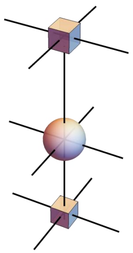
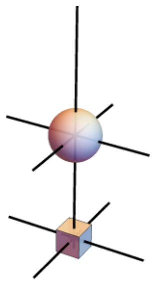
AD-BC-VUMPS

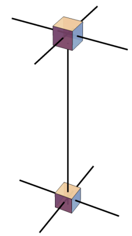
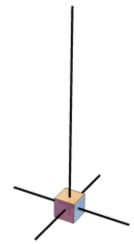
Why AD?

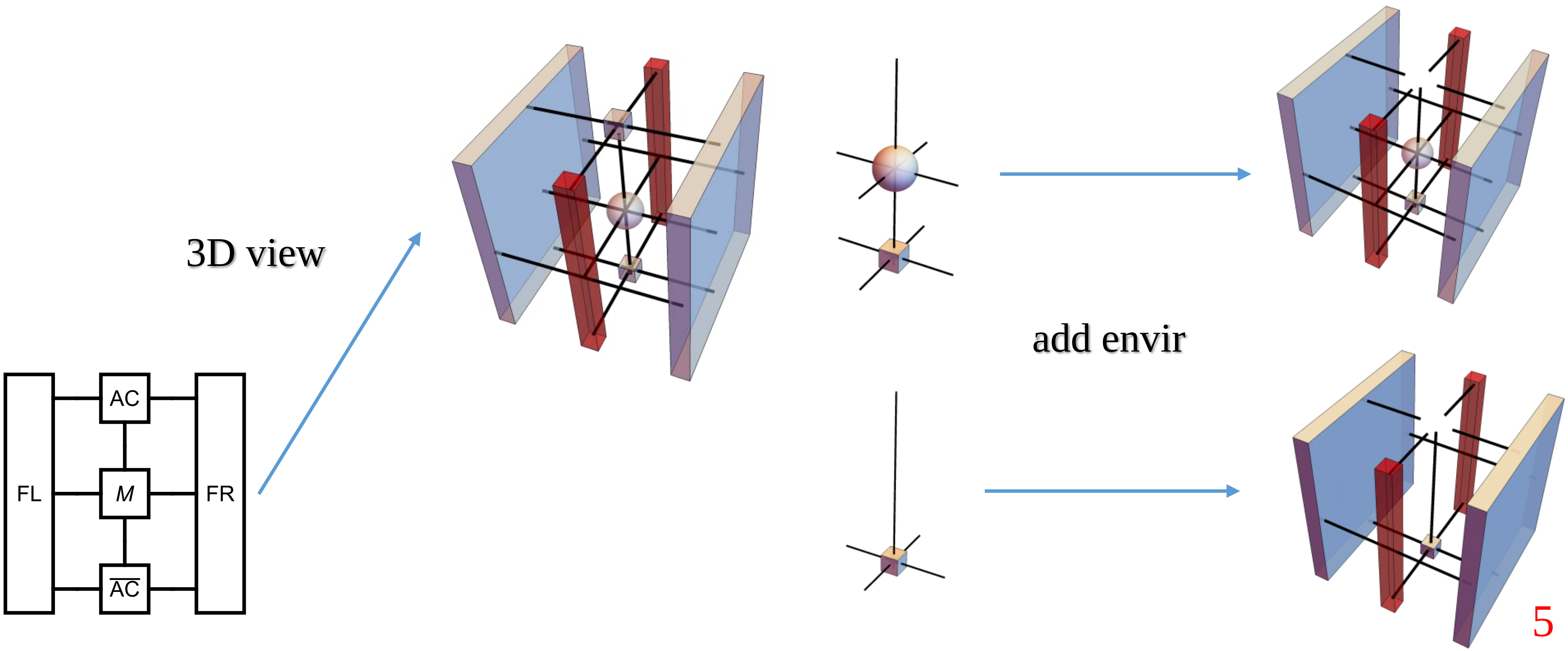
3D ising contract



Gradient to  and 

$\frac{\partial}{\partial A}($  $) = 2$ 

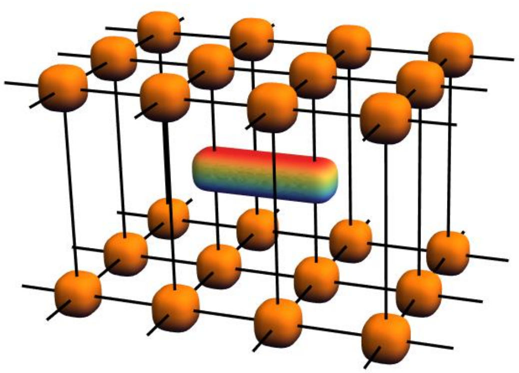
$\frac{\partial}{\partial A}($  $) = 2$ 

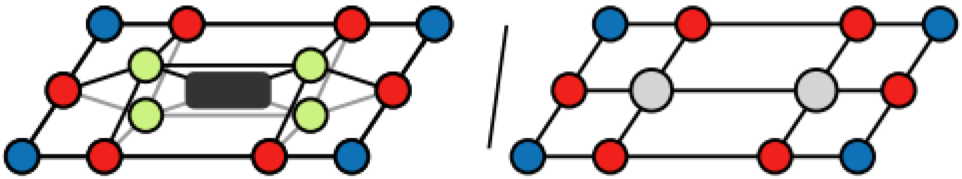


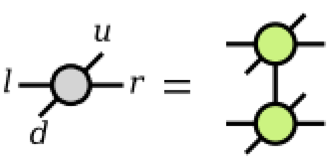
Quantum case

- 2D Energy

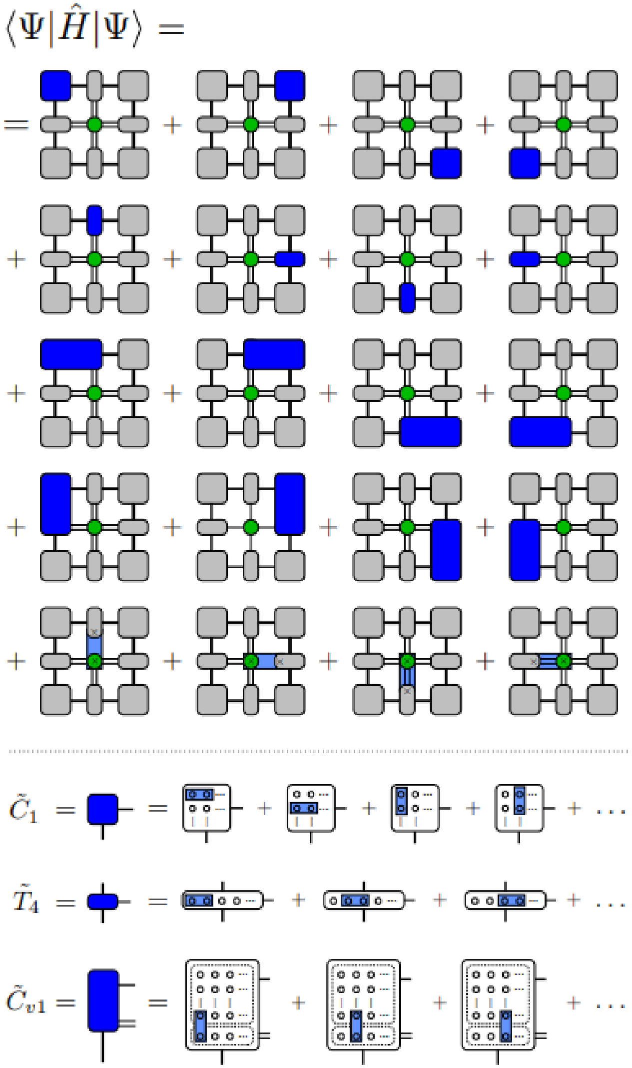
$$\min_A E(A) = \min_A \frac{\langle \Psi(A) | \hat{H} | \Psi(A) \rangle}{\langle \Psi(A) | \Psi(A) \rangle}$$

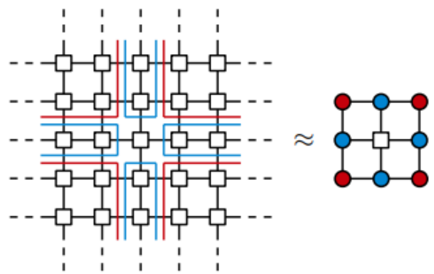


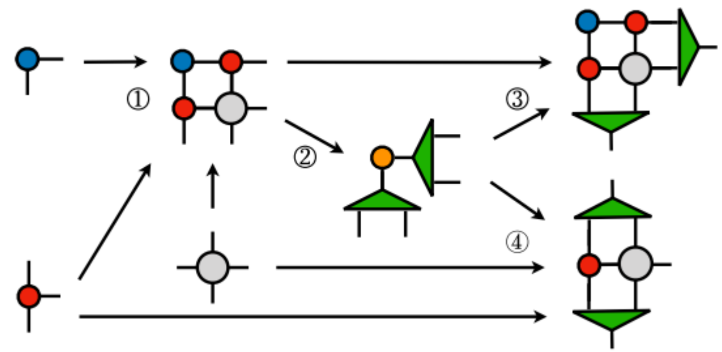
$\mathcal{L} =$ 

$T_{uldr} =$ 

CTMRG







- Experimental optimization method

(1) Compute $E(1)$ (corresponding to the previous energy with the old tensor $A' = A$) and $E(0.5)$ (corresponding to the energy with $A' = \tilde{A}$).

(2) If $E(0.5) < E(1)$, take $A' = \tilde{A}$ as the solution and exit.

(3) Define an initial step size Δ_0 (e.g., $\Delta_0 = 0.1$) and a tiny step size h (e.g., $h = 10^{-4}$).

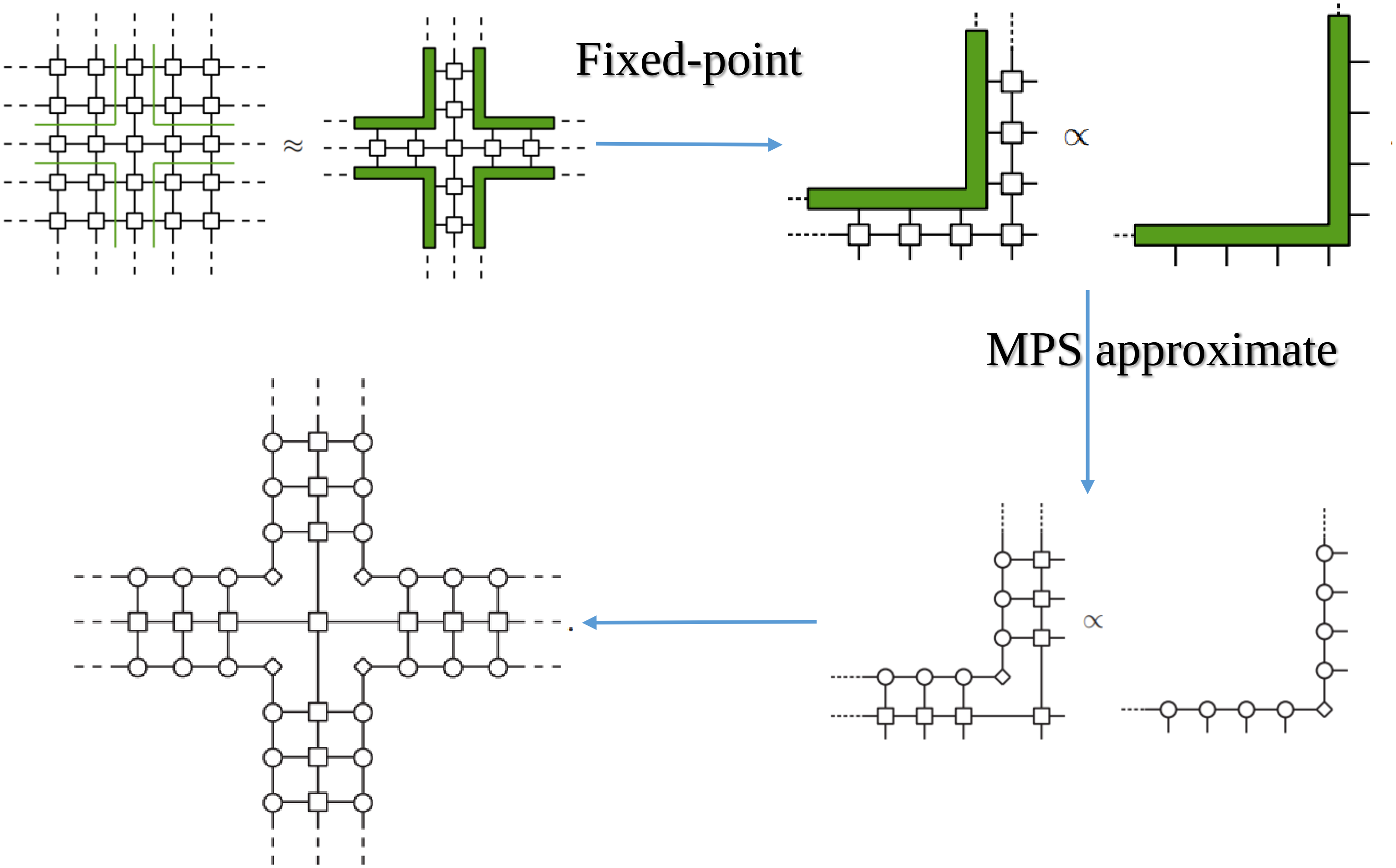
(4) If $E(1+h) < E(1)$, set $\Delta = \Delta_0$, else $\Delta = -\Delta_0$.

(5) For *iter* $= 1$ to *maxiter*

(a) If $E(1+\Delta) < E(1)$, accept solution [44] with $\lambda = 1 + \Delta$ and exit.

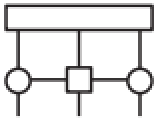
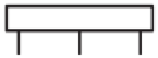
(b) Else $\Delta = \Delta/2$.

Channel environments

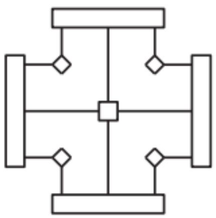


PRB 94, 155123 (2016)

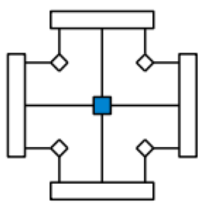
Gradient

 $= \lambda \times$ 

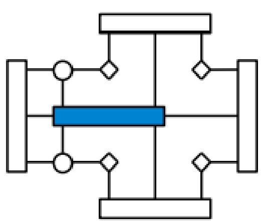
overlap

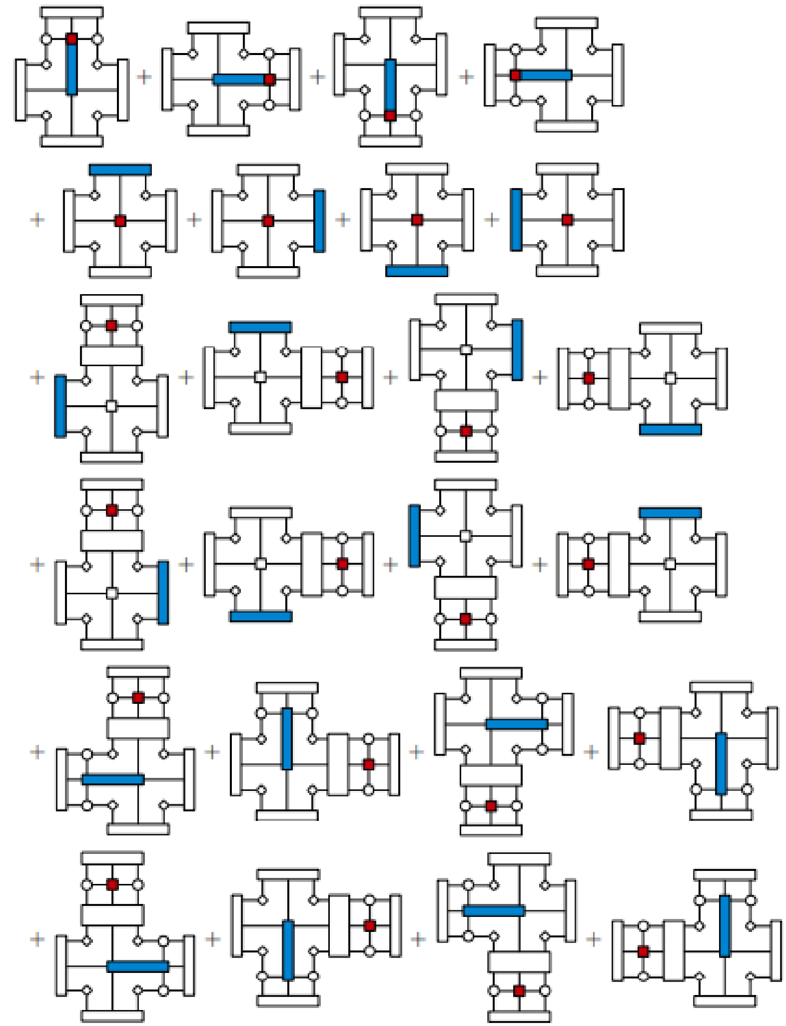
$\langle \Psi(A) | \Psi(A) \rangle =$ 

observable

$\langle \Psi(A) | O | \Psi(A) \rangle =$ 

two-site

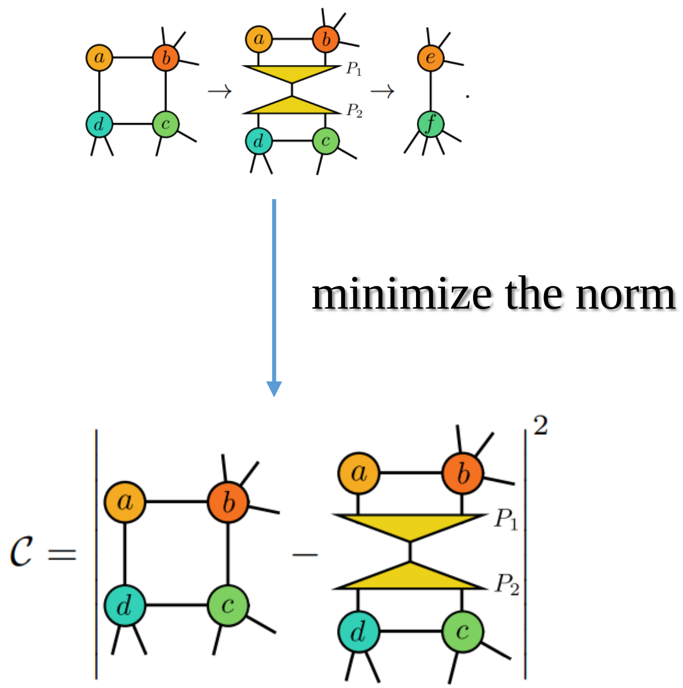
$\langle \Psi(A) | O | \Psi(A) \rangle =$ 

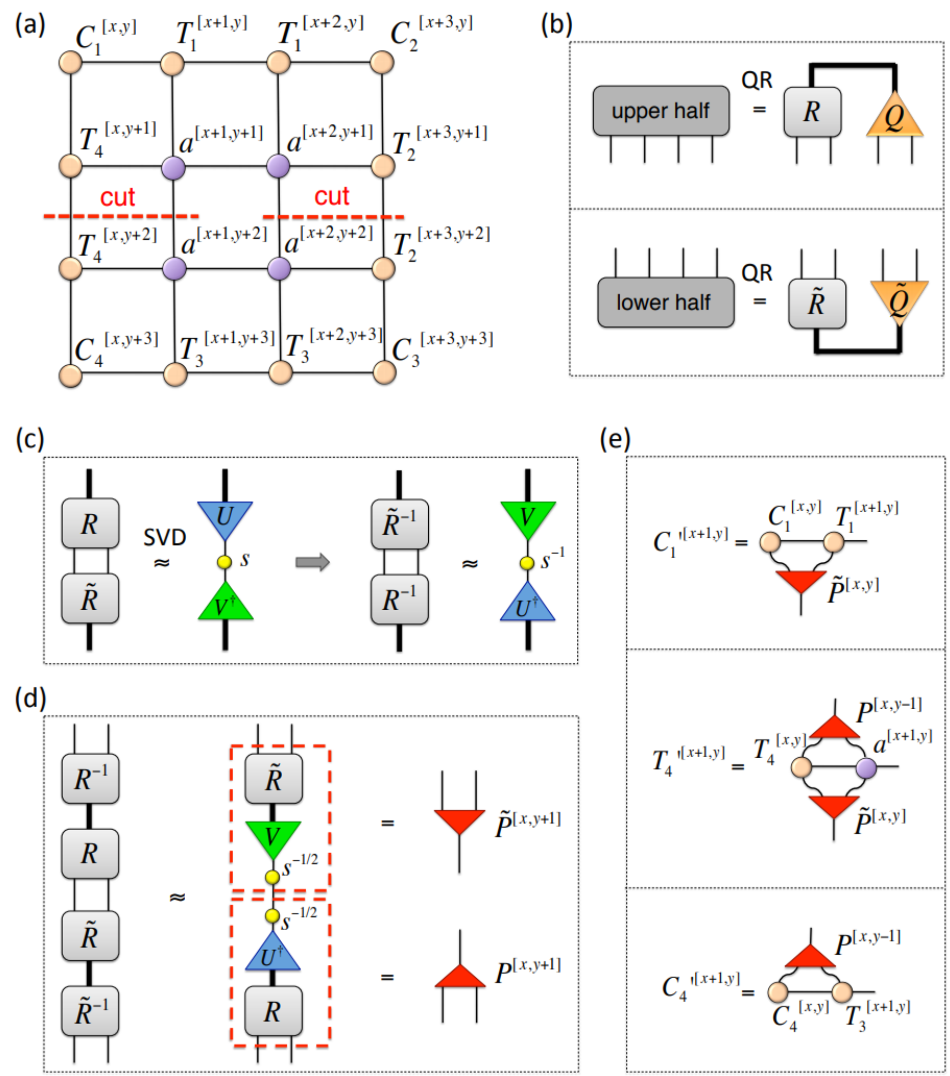
grad = 

PRB 94, 155123 (2016)

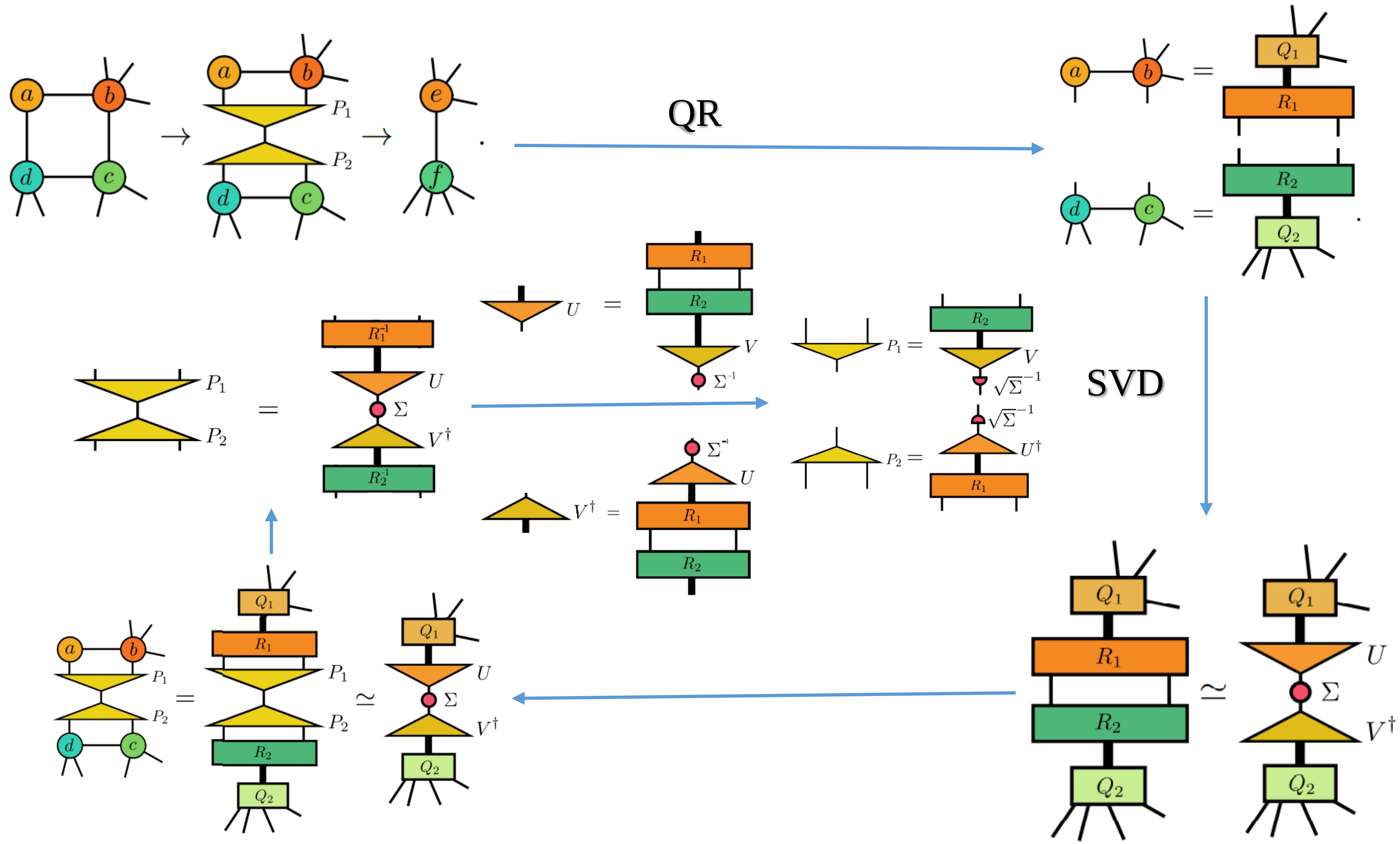
Big Cell CTMRG

- General case





Get P_1 and P_2

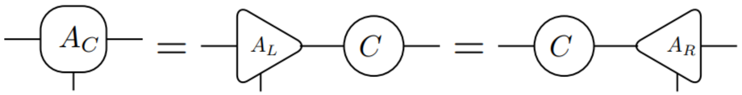


https://arxiv.org/abs/1905.02351

AD-VUMPS

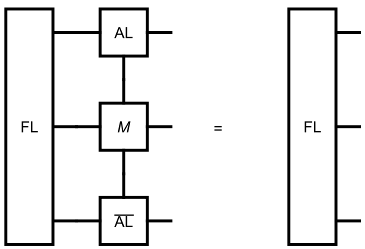
VUMPS-algorithm

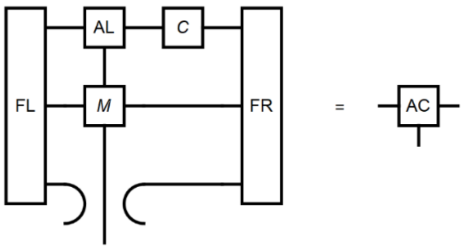
# Initial orthonormal form MPS
M = classical2Disingmpo(β; J = 1.0, h = 0.)
AL,C = leftorth (A)
AR = rightorth (AL,C)



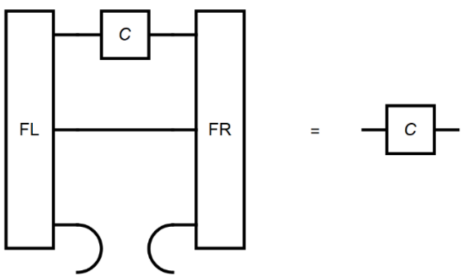
while error(AC,FL,FR,M) > tol_error

# Get environment
FL = leftenv(AL, M)
FR = rightenv(AR, M)

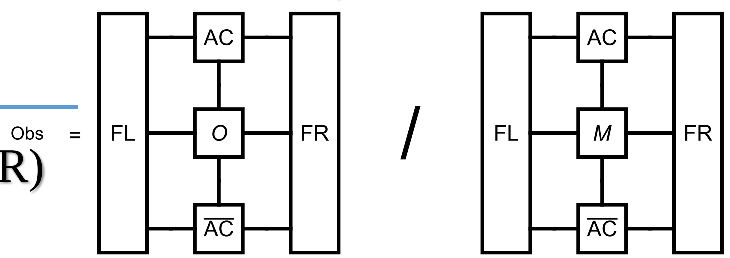




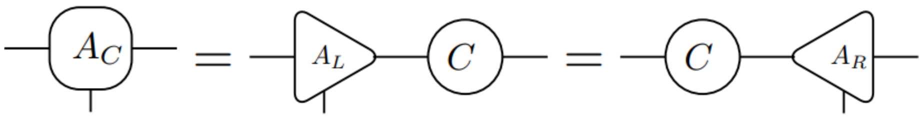
Get orthonormal form
AC, C = vumpsstep(AL, C, AR, M, FL, FR)



# calculate observable
Obs = obser(O, AC, M, FL, FR)



QR to avoid inverse



$$A_C = Q_{A_C} \cdot R_{A_C}$$
$$C = Q_C \cdot R_C$$

$$Q^T = Q^{-1}$$

↓

$$\begin{aligned} A_L &= A_C \cdot C^{-1} \\ &= Q_{A_C} \cdot R_{A_C} \cdot R_C^{-1} \cdot Q_C^{-1} \\ &= Q_{A_C} \cdot R_{A_C} \cdot R_C^{-1} \cdot Q_C^{T} \\ &\approx Q_{A_C} \cdot Q_C^{T} \end{aligned}$$

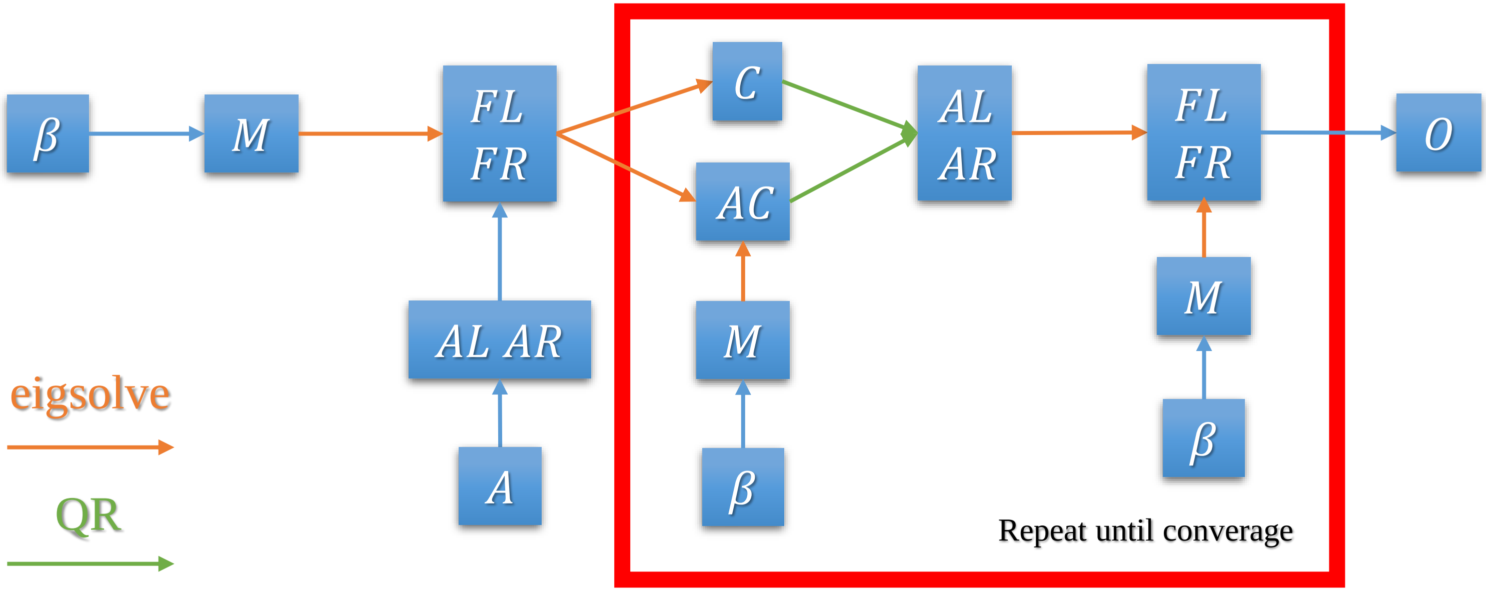
Error = R_{A_C} - R_C

↓

$$A_L^T \cdot A_L = Q_C \cdot Q_{A_C}^T \cdot Q_{A_C} \cdot Q_C^T = 1$$

inverse and orthogonality at the same time

Computation Graphs



Adjoint of eigsolve

- $l^T A = \lambda l^T, \quad Ar = \lambda r, \quad l^T r = 1,$

$$(A - \lambda I)\xi_l = (1 - r l^T)\bar{l}, \quad l^T \xi_l = 0$$
$$(A^T - \lambda I)\xi_r = (1 - l r^T)\bar{r}, \quad r^T \xi_r = 0$$

$$\longrightarrow \bar{A} = \bar{\lambda} l r^T - l \xi_l^T - \xi_r r^T$$

gauge invariant

$$l^T \bar{l} = r^T \bar{r} = 0$$

$$(A - \lambda I)\xi_l = \bar{l}, \quad l^T \xi_l = 0$$
$$(A^T - \lambda I)\xi_r = \bar{r}, \quad r^T \xi_r = 0$$

Only l

$$\boxed{(A - \lambda I)\xi_l = \bar{l}, \quad l^T \xi_l = 0}$$
$$(A^T - \lambda I)\xi_r = 0, \quad r^T \xi_r = 0$$

$$\boxed{\bar{A} = -l \xi_l^T}$$

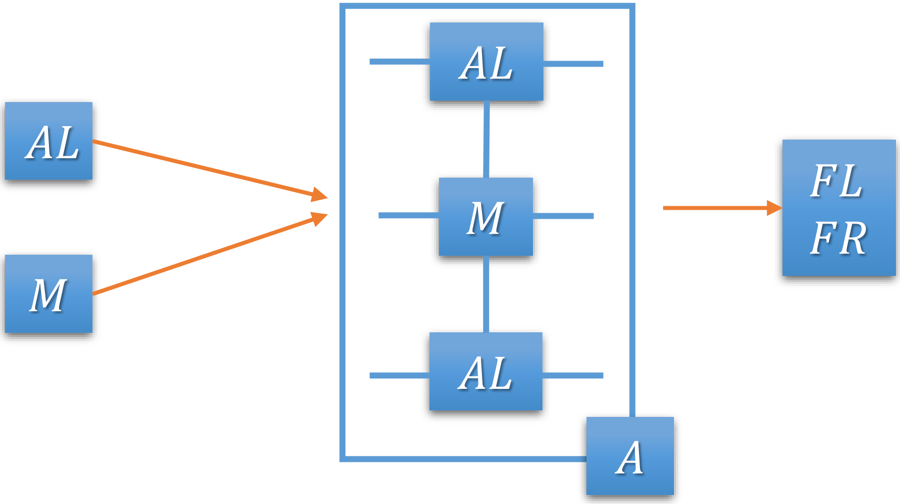
Only r

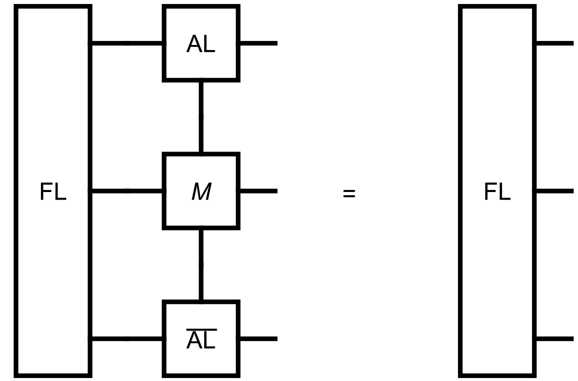
$$(A - \lambda I)\xi_l = 0, \quad l^T \xi_l = 0$$
$$\boxed{(A^T - \lambda I)\xi_r = \bar{r}, \quad r^T \xi_r = 0}$$

$$\boxed{\bar{A} = -\xi_r r^T}$$

PRB 101, 245139 (2020)

Adjoint of VUMPS envirment





$$(A - \lambda I)\xi_l = \bar{l}, \quad l^T \xi_l = 0$$

$$\bar{A} = -l\xi_l^T$$

$$\overline{AL} = \bar{A} \cdot \frac{\partial A}{\partial AL}$$

$$\bar{M} = \bar{A} \cdot \frac{\partial A}{\partial M}$$

Adjoint of QR

- $A = QR$

$$\bar{A} = [\bar{Q} + Q\texttt{copyltu}(M)]R^{-T}$$

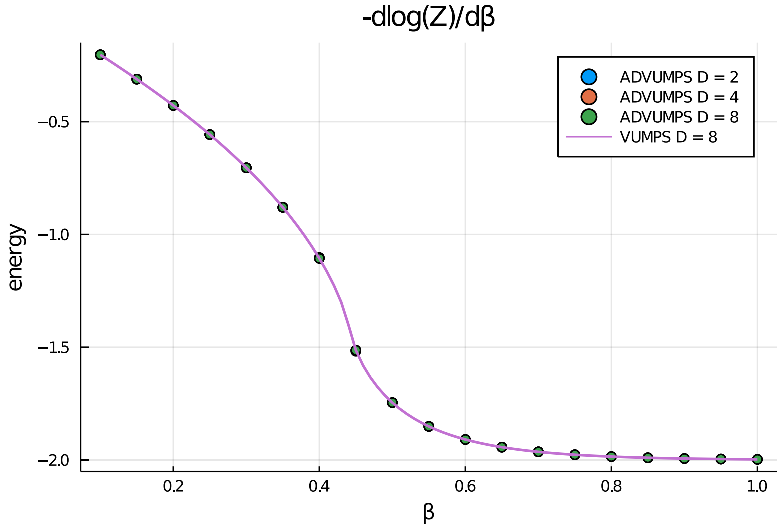
$$M = R\bar{R}^T - \bar{Q}^T Q$$

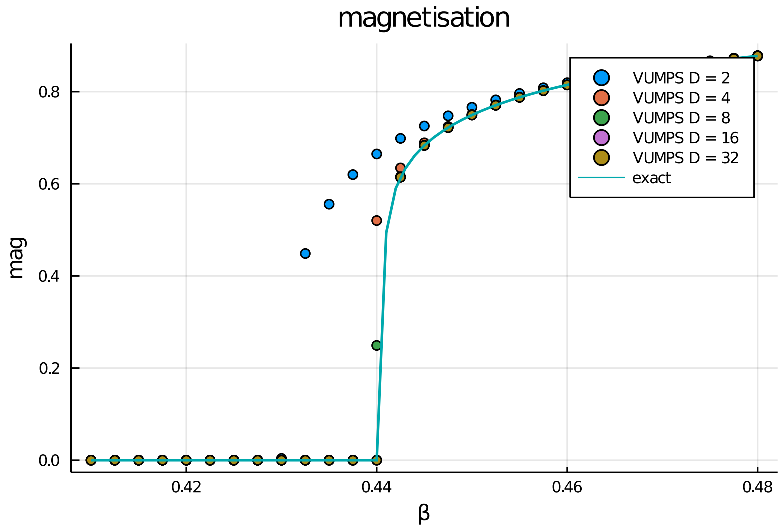
$$[\texttt{copyltu}(M)]_{ij} = M_{\max(i,j),\min(i,j)}$$

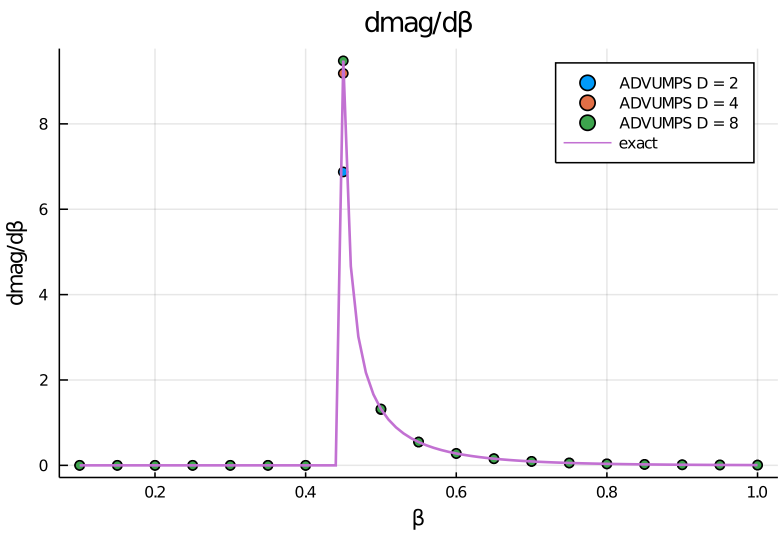
- Trick: add $\delta = 10^{-6}$ to R's diagonal element for the stability of inverse

PRX 9, 031041 (2019)

2D Classical Ising Model





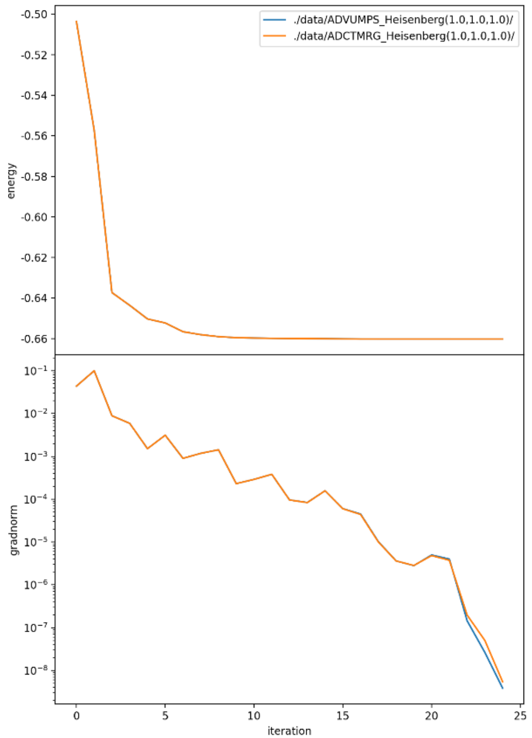


https://github.com/XingyuZhang2018/ADVUMPS.jl

Finding the Ground State of infinite 2D Heisenberg model

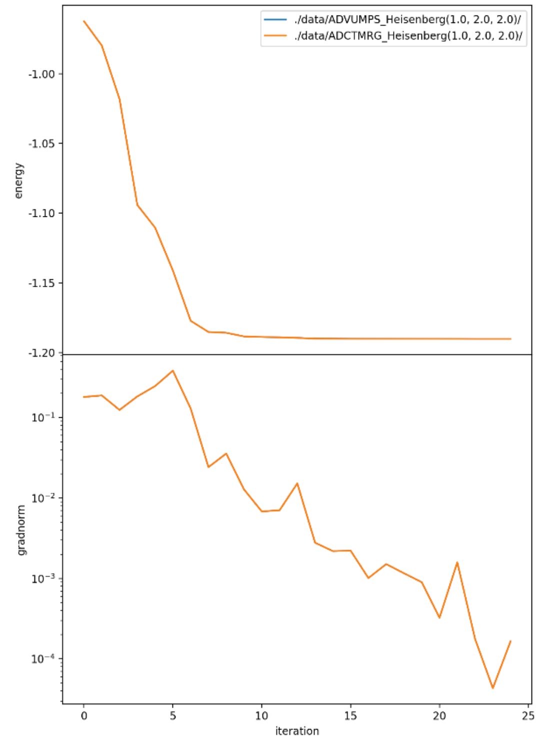
ADVUMPS vs ADCTMRG $D = 2\ \chi = 20\ \delta = 10^{-12}$

33.76s vs 15.62s



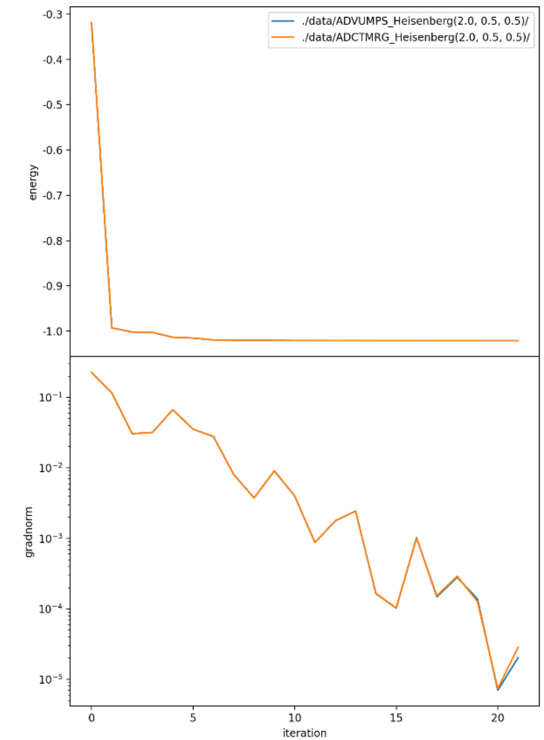
Difference ~1e-14

29.11s vs 15.58s



Difference ~1e-9

16.61s vs 4.94s

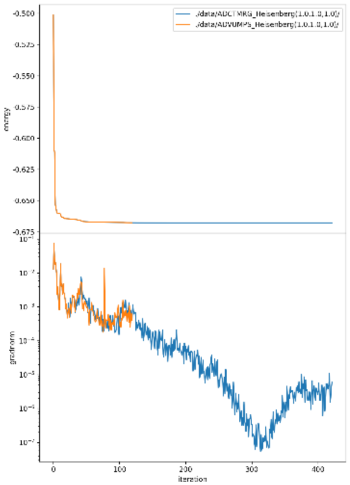


Difference ~1e-9

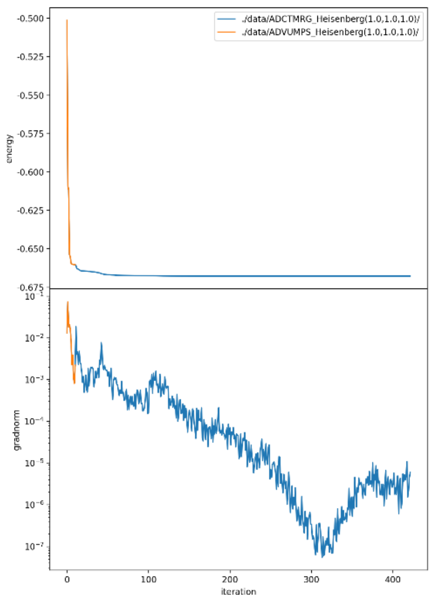
		Time			Allocations		
Tot / % measured:		8.66s / 100%			7.54GiB / 100%		
Section	ncalls	time	%tot	avg	alloc	%tot	avg
backward	61	6.22s	72.1%	102ms	6.11GiB	81.0%	103MiB
forward	61	2.41s	27.9%	39.5ms	1.43GiB	19.0%	24.0MiB

		Time			Allocations		
Tot / % measured:		30.2s / 100%			26.2GiB / 100%		
Section	ncalls	time	%tot	avg	alloc	%tot	avg
backward	62	22.8s	75.8%	368ms	20.1GiB	76.6%	332MiB
forward	62	7.30s	24.2%	118ms	6.13GiB	23.4%	101MiB

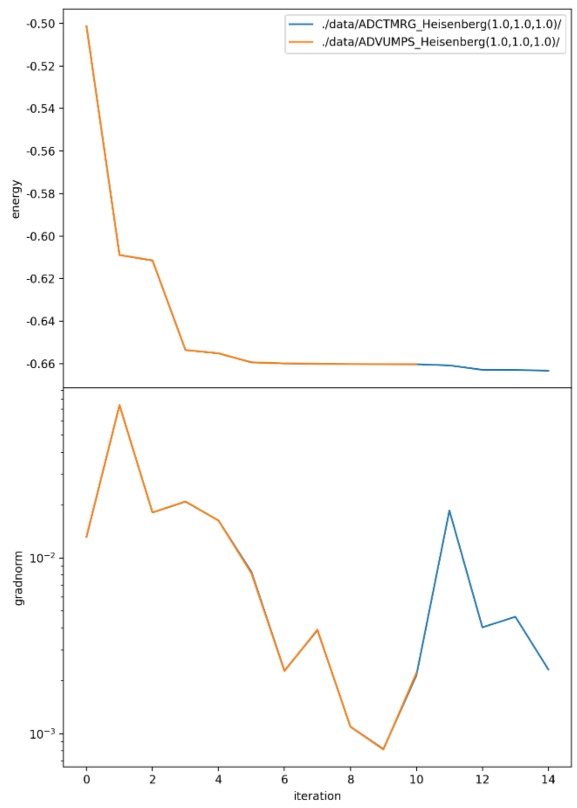
Problem of D=3



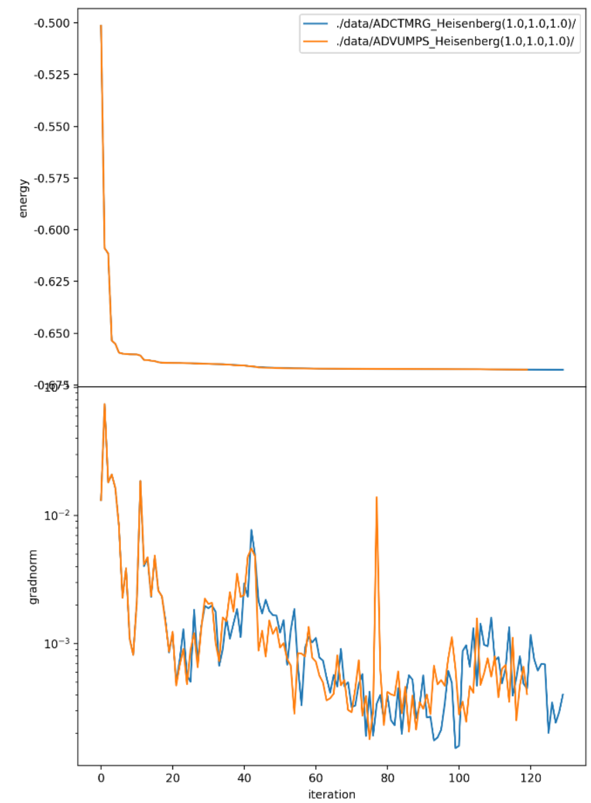
- QR isn't stable for R^{-1}



tol=1e-10



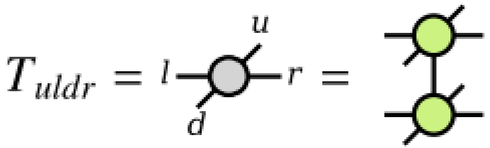
tol=1e-20



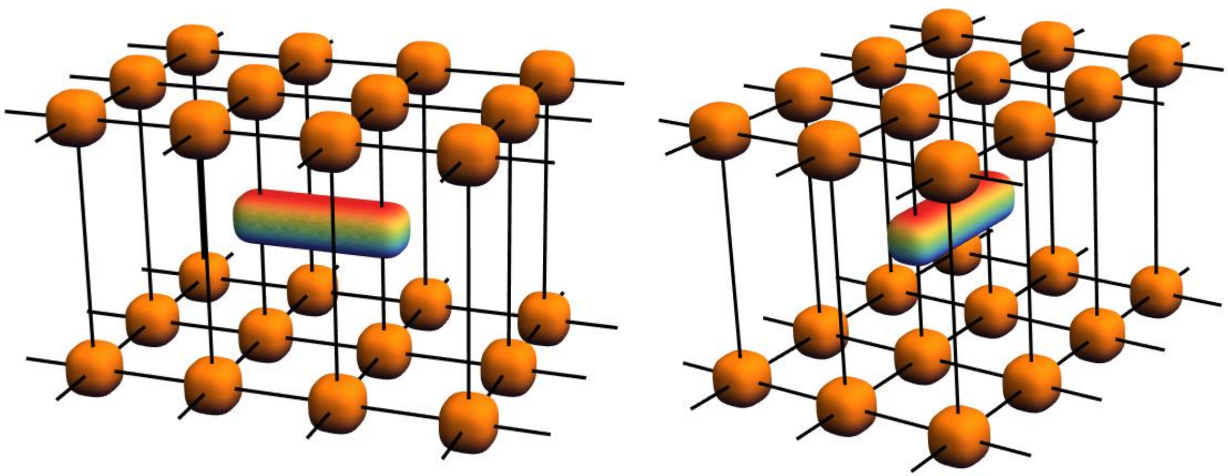
Problem of iPEPS symmetry

- When only require up and down symmetry, Heisenberg(1,1,1)

$$E_0 = -0.85 < -0.66$$



- Vertical and horizontal bond energy should be optimized at the same time



AD-BC-VUMPS

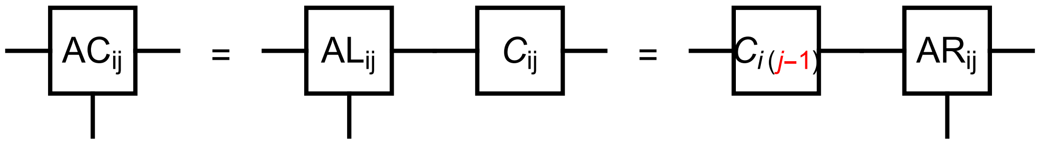
BCVUMPS-algorithm

Initial orthonormal form MPS

M_{ij} = mpo(params)

AL_{ij}, C_{ij} = leftorth (A_{ij})

AR_{ij} = rightorth (AL_{ij}, C_{ij})

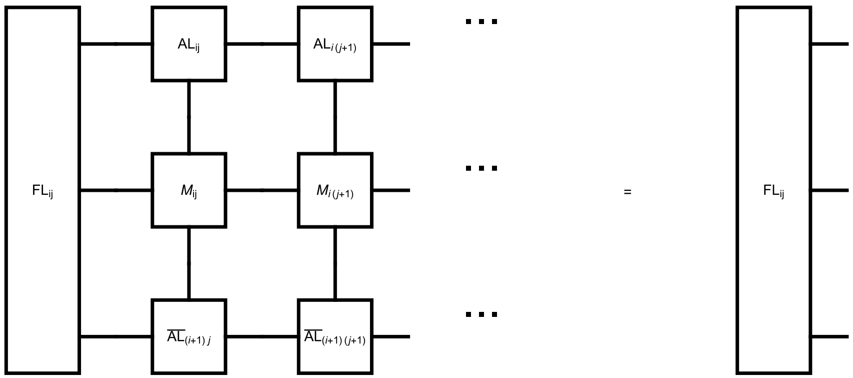


while error(AC,FL,FR,M) > tol_error

Get environment

AL[i, j] = leftenv(AL[i , :], M [i , :])

AR[i, j]= rightenv(AR[i , :], M [i , :])

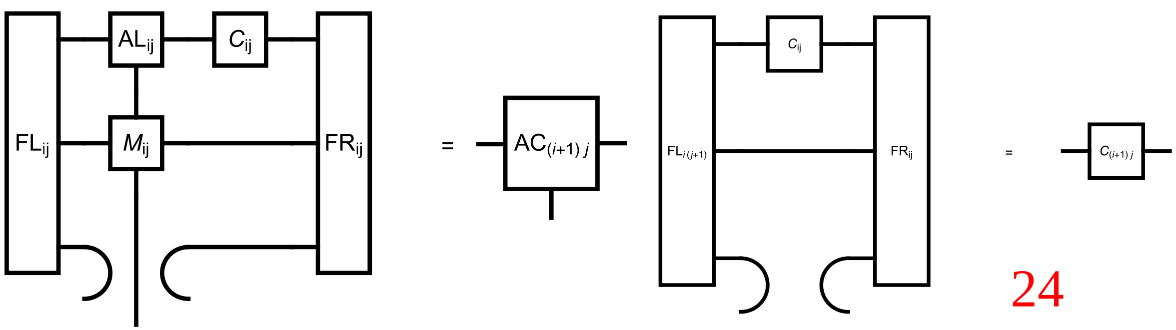


Get orthonormal form

AC[i, j], C[i, j] = vumpsstep(AL[:, j], C[:, j], AR[:, j], M[:, j], FL[:, j], FR[:, j])

calculate observable

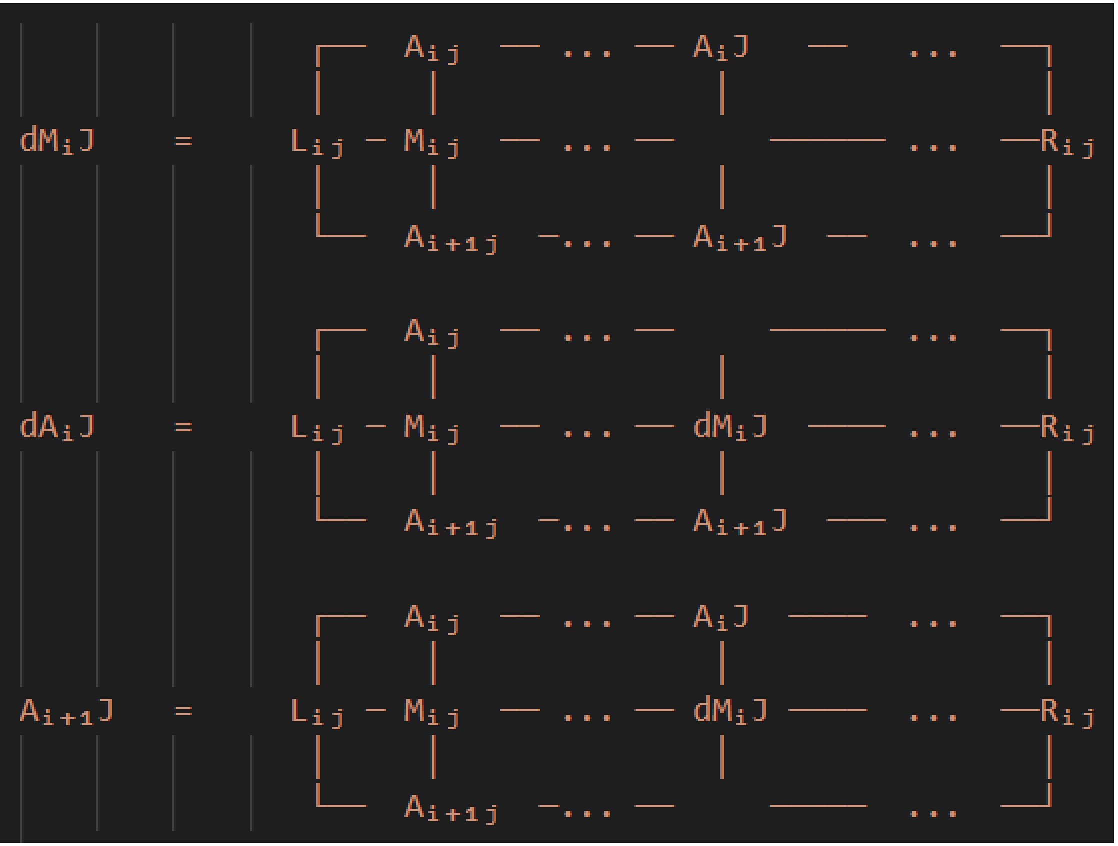
Obs = obser(O, AC, M, FL, FR)

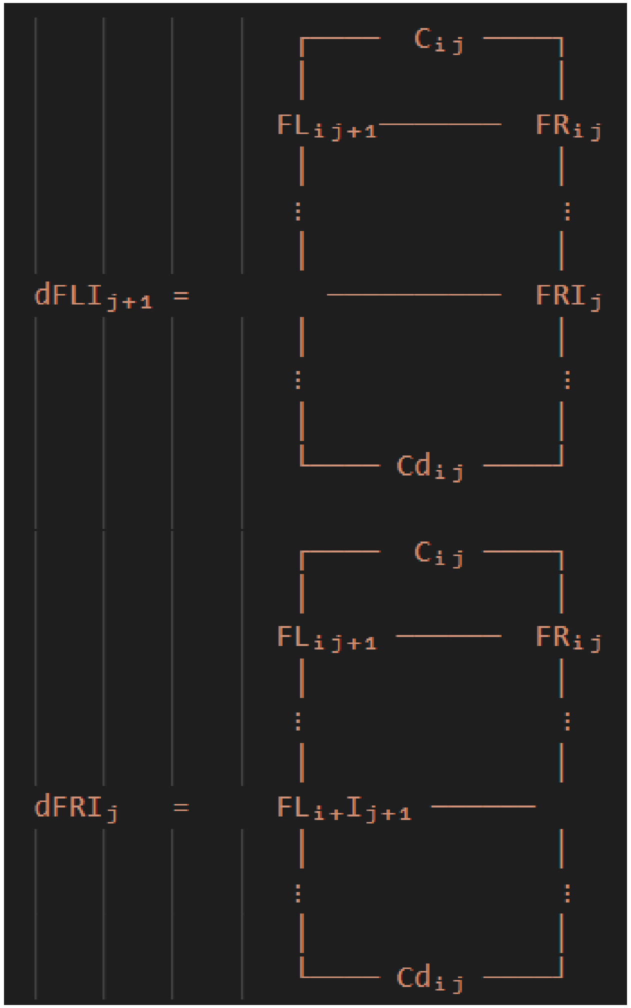


AD for array of array

```
# example to solve differential of array of array
# use `[]` list then reshape
A = Array{Array,2}(undef, 2, 2)
for j = 1:2,i = 1:2
    A[i,j] = rand(2,2)
end
function foo2(x)
    # B[i,j] = A[i,j].*x   # mistake
    B = reshape([A[i].*x for i=1:4],2,2)
    return sum(sum(B))
end
@test Zygote.gradient(foo2, 1)[1] ≈ num_grad(foo2, 1)
```

Be careful of index and replaced function!





AD-BC-VUMPS

Contents

some tricks → AD-VUMPS

- TFising
- Heisenberg

AD-BC-VUMPS

- 2×2 Heisenberg
- Honeycomb Kitaev

Subtle VUMPS

Some argument for up and down VUMPS

some Tricks

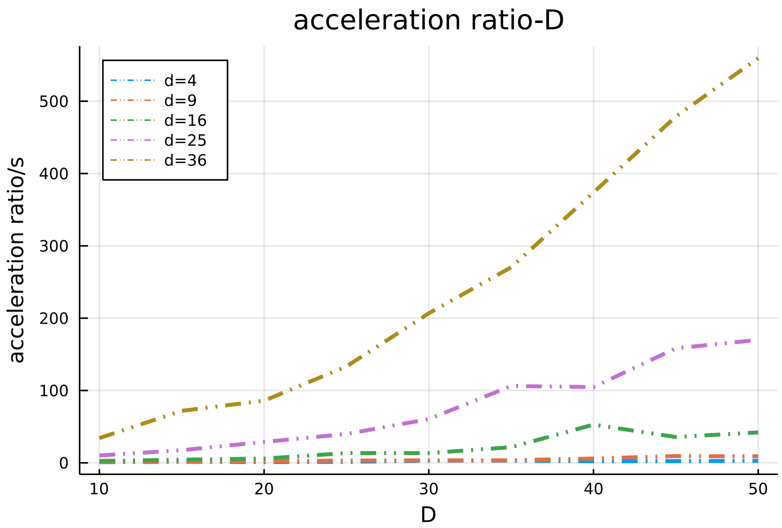
some tricks to accelerate

- Save vumps environment
- Optimize the contract order
- Set maxiter = 1 for linsolve but eigsolve not
- GPU

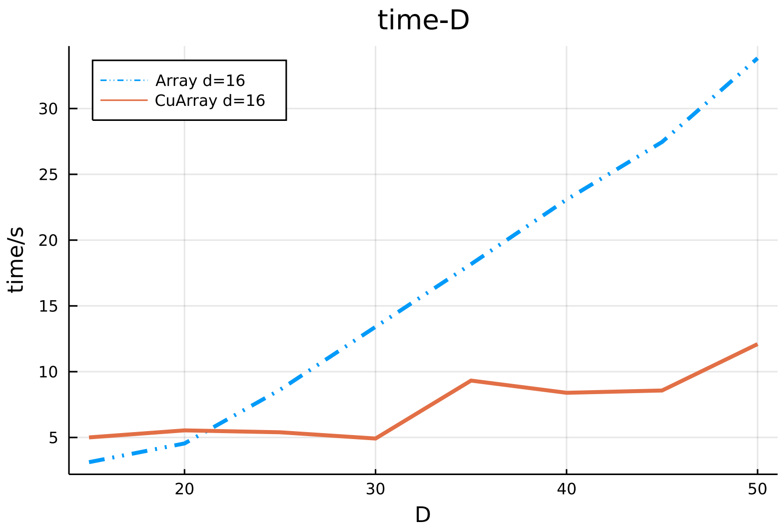
For Julia OMEinsum

optimize_greedy(; method=MinSpaceDiff())

OMEinsum



KrylovKit

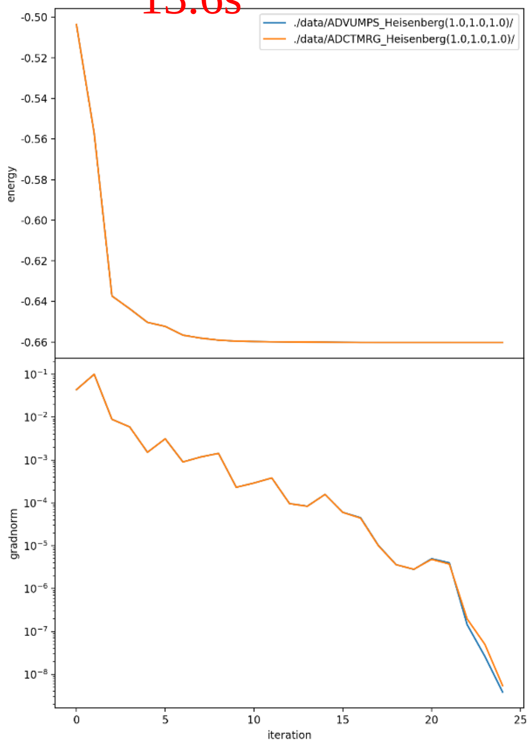


Finding the Ground State of infinite 2D Heisenberg model

ADVUMPS vs ADCTMRG $D = 2$ $\chi = 20$ $\delta = 10^{-12}$

33.76s vs 15.62s

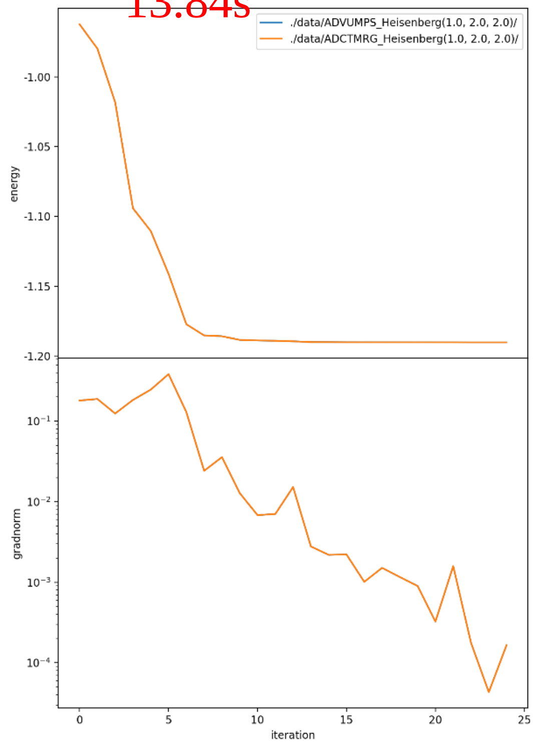
13.6s



Difference ~1e-14

29.11s vs 15.58s

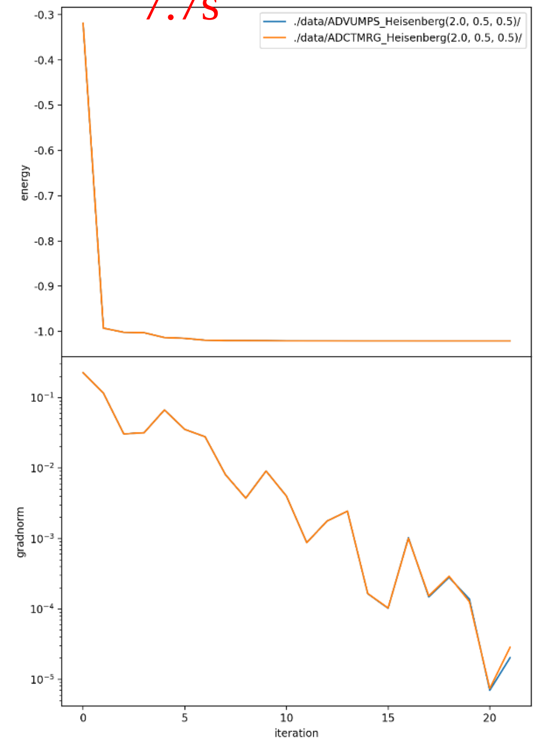
13.84s



Difference ~1e-9

16.61s vs 4.94s

7.7s



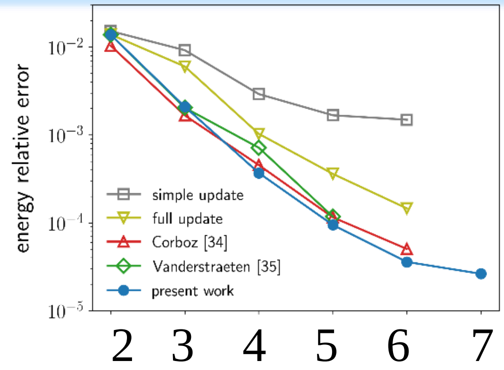
Difference ~1e-9

Backward	9.91s	72.3%	10.7s	77.1%	6.14s	76.9%
forward	3.79s	27.7%	3.19s	22.9%	1.84s	23.1%

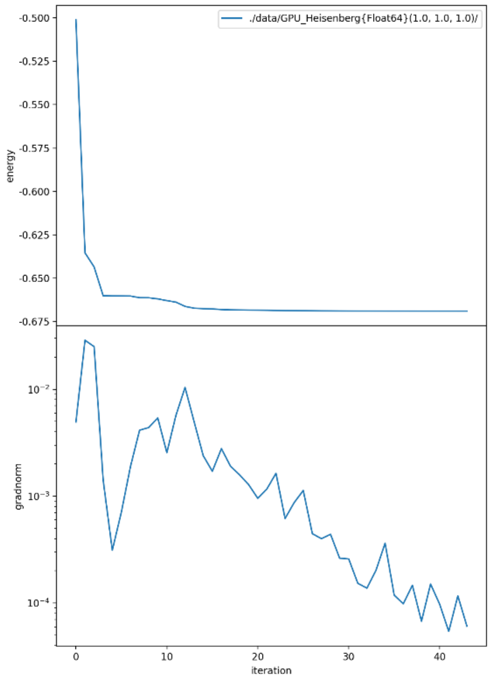
Heisenberg model $D = 4,5,6$

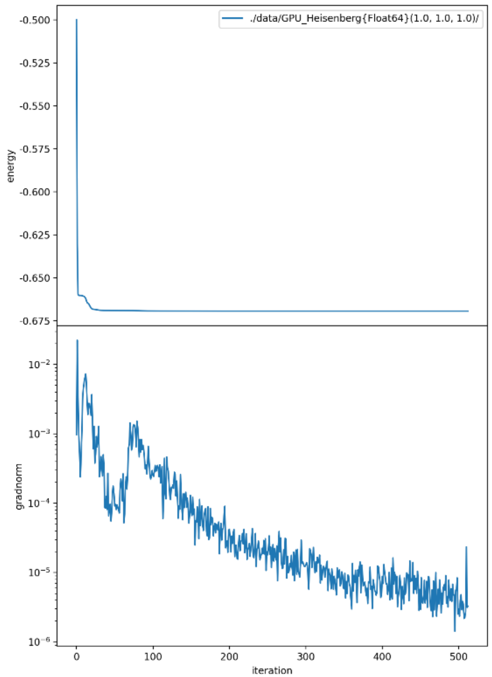
one RTX2060-6G

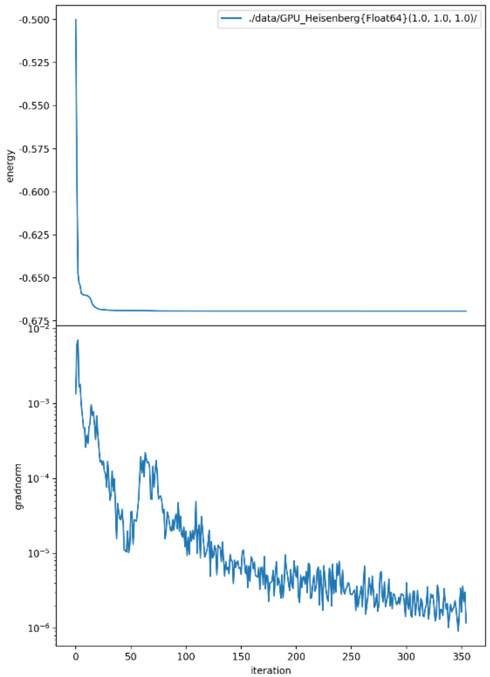
D	χ	iteration	time	E	error
4	30	44	22.7 min	-0.6689534	7.3×10^{-4}
5	30	513	6.6h	-0.6693341	1.6×10^{-4}
6	30	355	16.1h	-0.6693380	1.56×10^{-4}



MC: -0.6694421







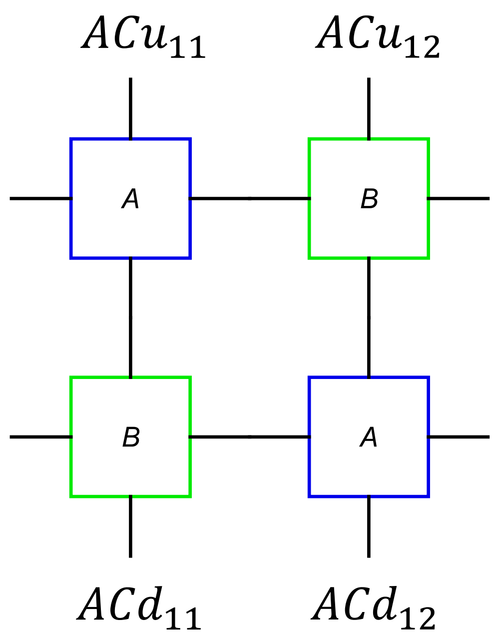
2×2 AD-BC-VUMPS

symmetry environment

- $H = \sum_{ij} S_i^z S_j^z + \frac{1}{2}(S_i^+ S_j^- + S_i^- S_j^+)$

$$\Big\downarrow \quad S_j \leftarrow \sigma_x S_j \sigma_x'$$

- $H = -\sum_{ij} S_i^z S_j^z + \frac{1}{2}(S_i^+ S_j^+ + S_i^- S_j^-)$



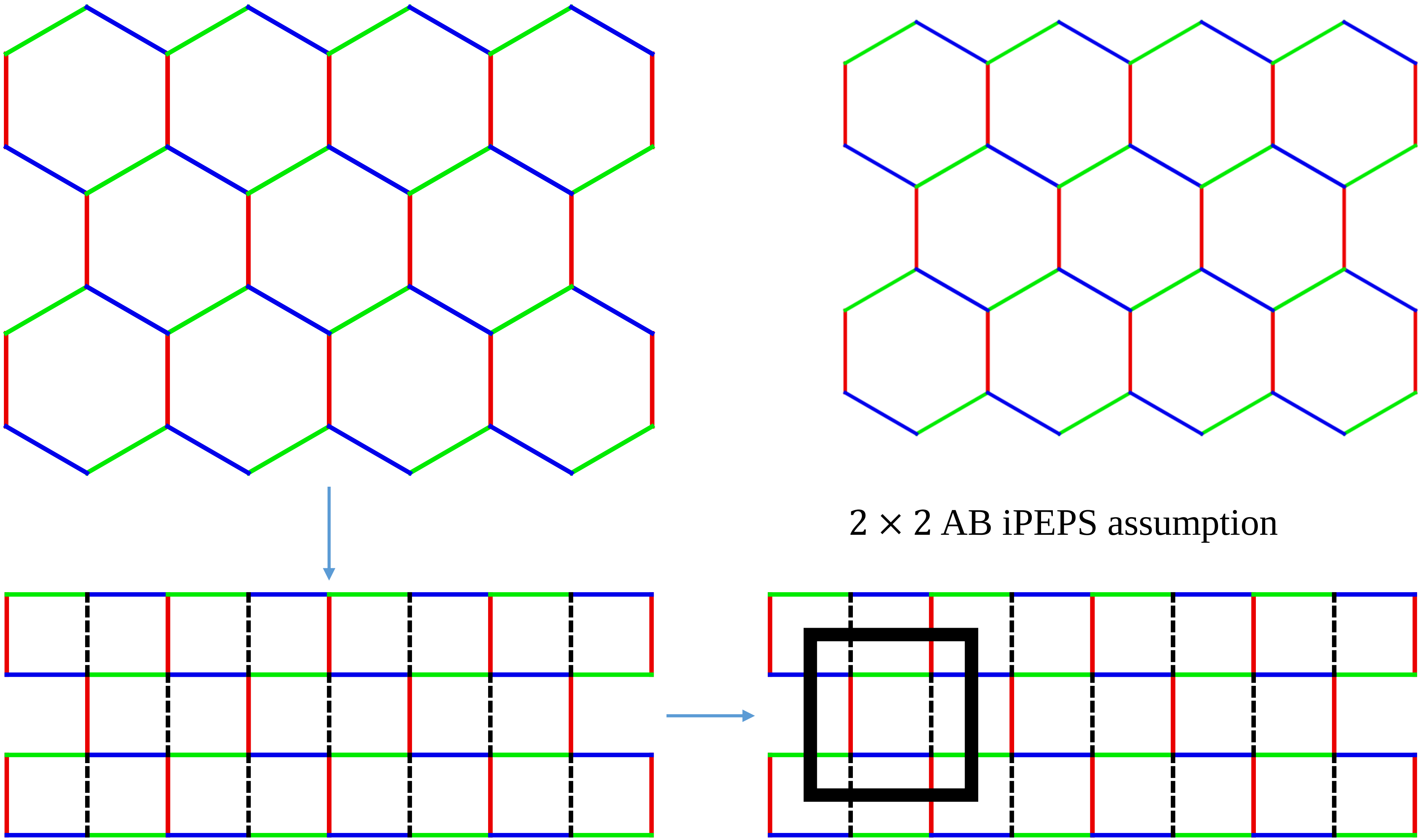
Up-and-down-symmetry assumption

$ACd_{11} = ACu_{12}$

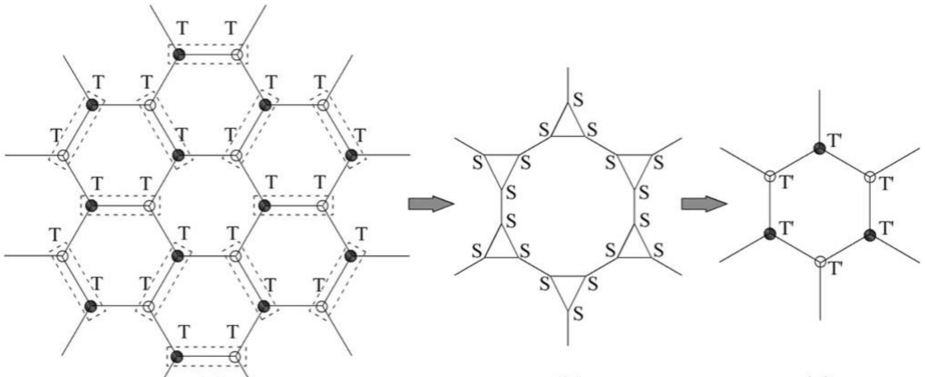
$ACd_{12} = ACu_{11}$

D	χ	iteration	time	E
2	20	25	122.9s	-0.6602265
4	32	134	5.5h	-0.6689648
4	32	86	2.5h	-0.6689539
4	30	513	22.7 min	-0.6689534

Honeycomb lattice to square



Honeycomb Heisenberg



(a) (b) (c)

D	χ	E	M
2	20	-0.5400	0.243
3	20	-0.5430	0.222
4	20	-0.5440	0.200

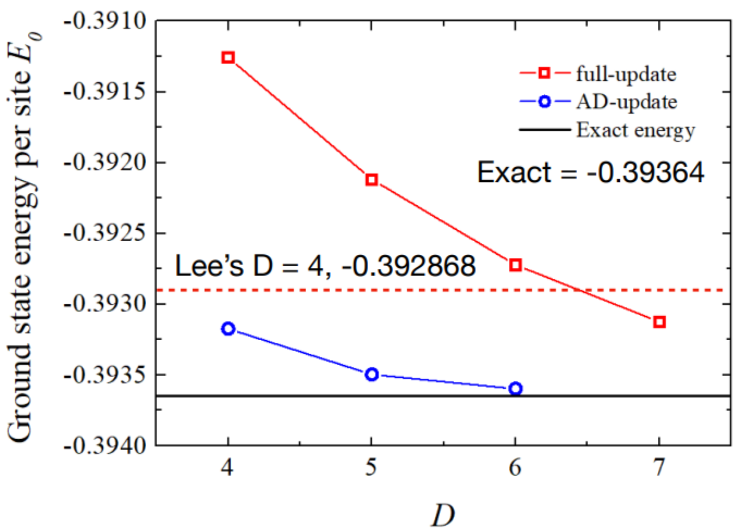
D	E	M
3	−0.5365	0.249
4	−0.5456	0.228
5	−0.5488	0.220
6	−0.5513	0.206
7	−0.5490	0.216
8	−0.5506	0.212

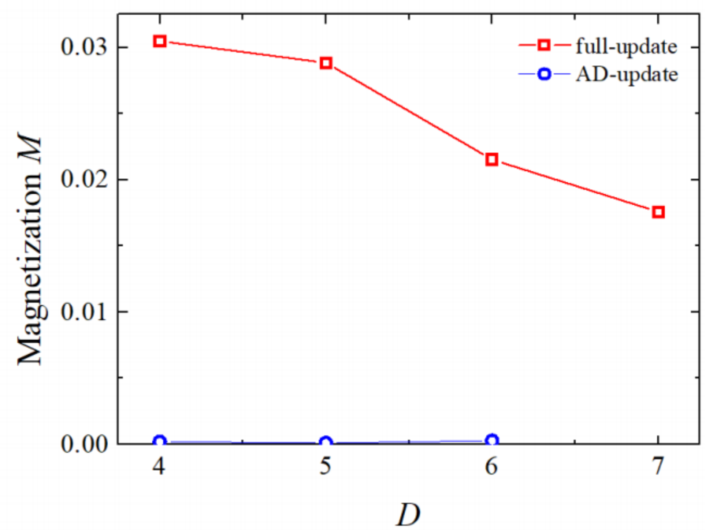
Method	E	M
Spin wave [12]	−0.5489	0.24
Series expansion [13]	−0.5443	0.27
Monte Carlo [14]	−0.5450±0.001	0.22±0.03
Ours $D = 8$	−0.5506	0.21 ± 0.01

PRL 101, 090603 (2008)

Honeycomb Kitaev

Iregui, *et. al.*, PRB 90, 195102 (2014)





CTMRG + imaginary-time evolution

vs.

TEBD + AD

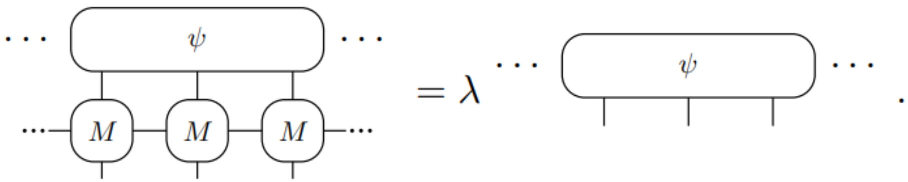
```
D  chi          En                      M
3  50    -0.3795995949421732    0.2721101290509978
4  80    -0.3931735662139908    0.0001872923118720
5  100   -0.3934970101949562    0.0000972301264798
6  100   -0.3935976940110233    0.0002659684803263
```

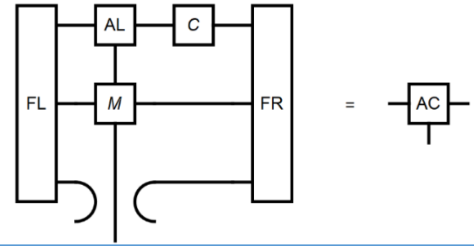
Theoretical value: −0.3936 0

D	χ	E	M
2	20	-0.3759726	
4	20	-0.3914..	0.02..

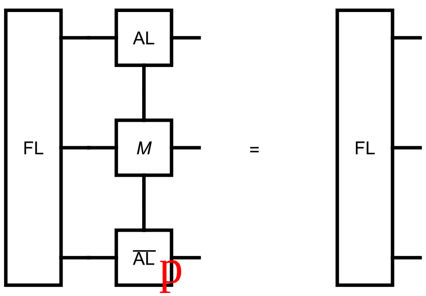
VUMPS doesn't converge for error ≈ 0.1

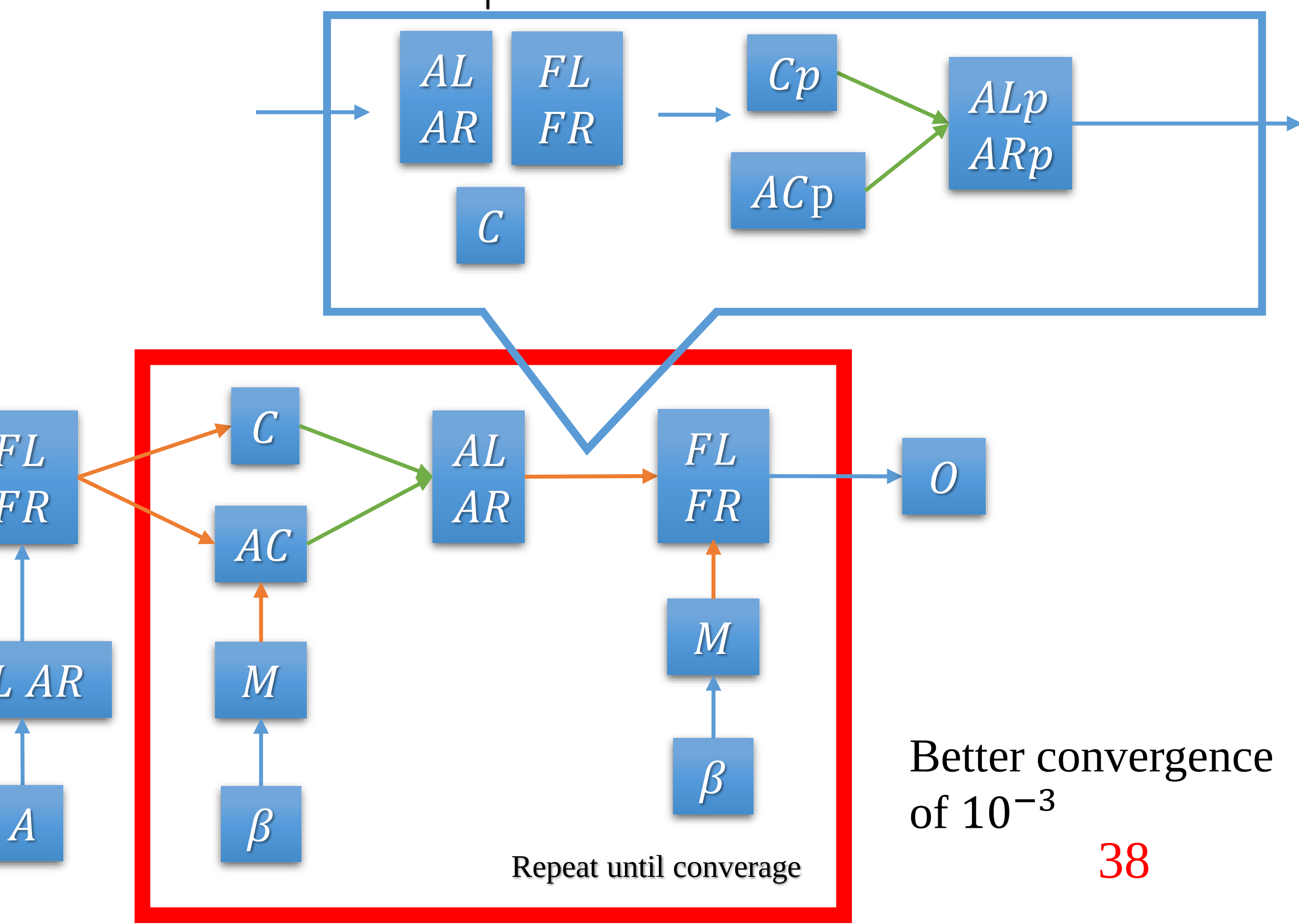
Computation Graphs





$$M\psi^n = \psi^{n+1}$$





Better convergence of 10^{-3}

result

D	chi	En	M
3	50	-0.3795995949421732	0.2721101290509978 5
4	80	-0.393173566213990 8	0.000187292311872 0
5	100	-0.3934970101949562	0.0000972301264798
6	100	-0.3935976940110233	0.0002659684803263

D	χ	E	M
4	20	-0.3933225282386825	0.0005
4	30	-0.3934358462431455 4	0.002

VUMPS up error ≈ 1e-6
down error ≈ 1e-6

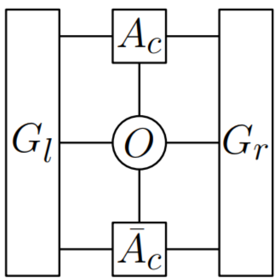
VUMPS up error ≈ 1e-7
down error ≈ 1e-2

$$E = -0.3933 \sim -0.3929$$

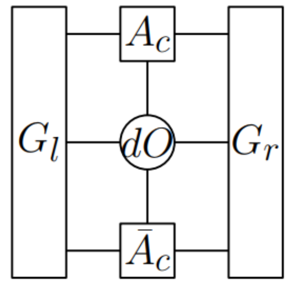
Some arguments for up and down VUMPS

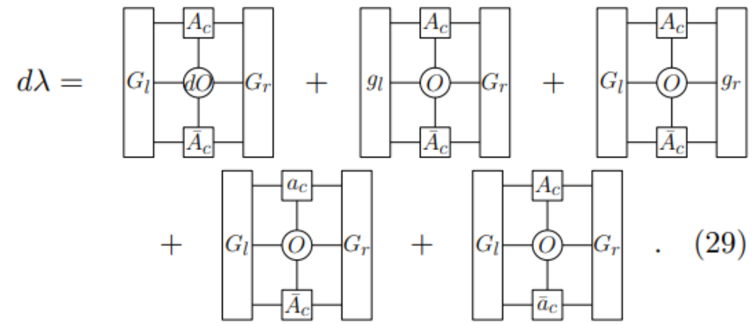
Differential for observable

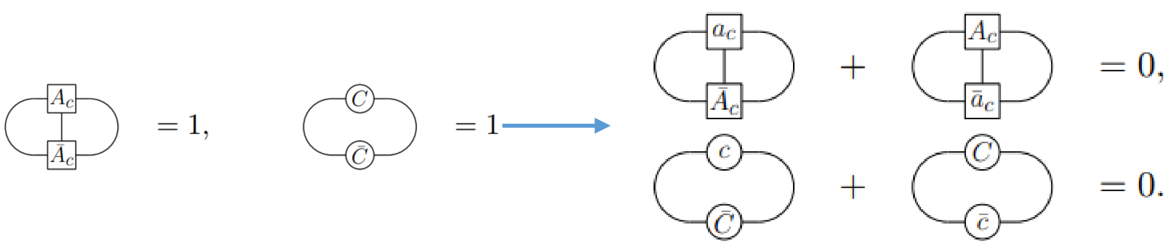
$\lambda =$

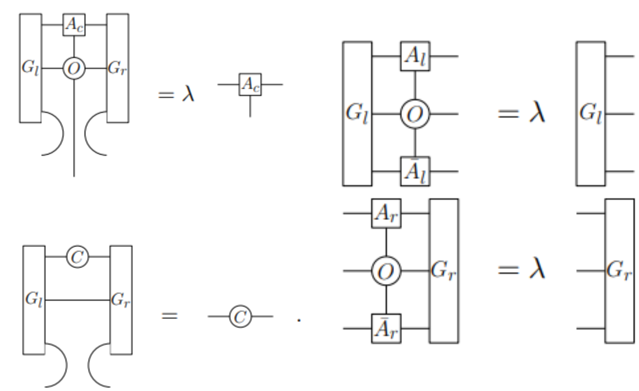


$d\lambda =$

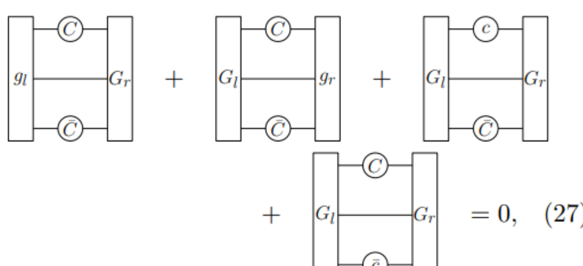


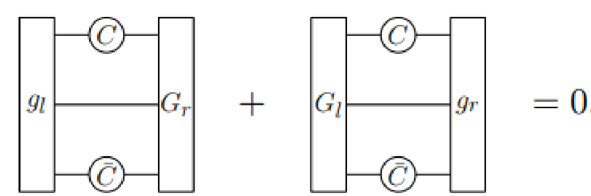






normalization of G_l and G_r





Contradictory result for up and down