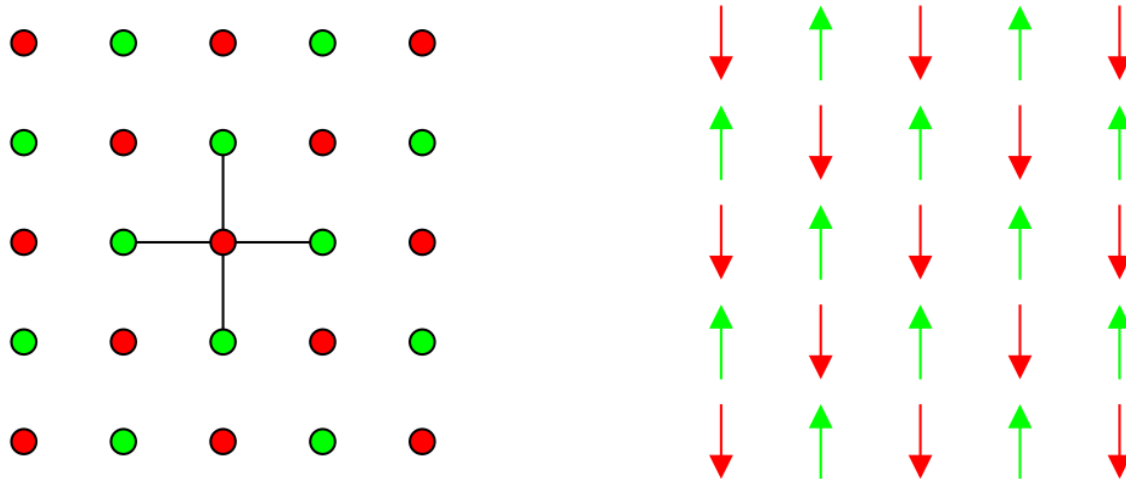

AD-U1-FPEPS

Xingyu Zhang
2022.07.22

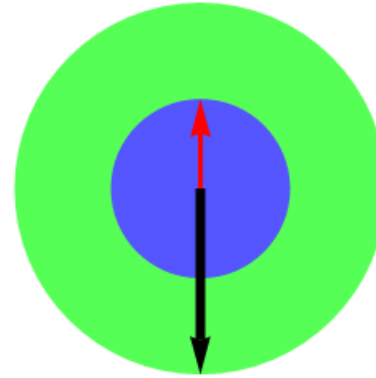
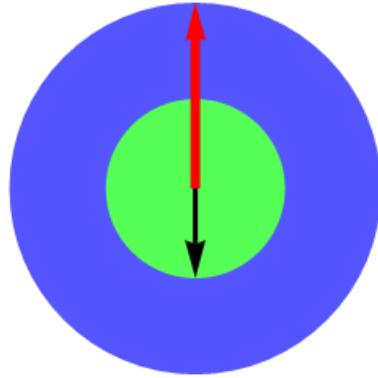
Hubbard model

$$\hat{H} = -t \sum_{\langle \mathbf{j}, \mathbf{l} \rangle \sigma} \left(c_{\mathbf{j}\sigma}^\dagger c_{\mathbf{l}\sigma} + c_{\mathbf{l}\sigma}^\dagger c_{\mathbf{j}\sigma} \right) + U \sum_{\mathbf{j}} n_{\mathbf{j}\uparrow} n_{\mathbf{j}\downarrow} - \mu \sum_{\mathbf{j}} (n_{\mathbf{j}\uparrow} + n_{\mathbf{j}\downarrow})$$

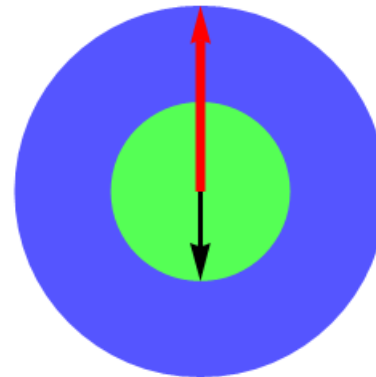
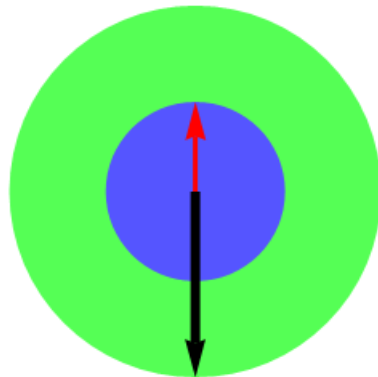
- $\mu = \frac{U}{2}$, Particle-hole symmetry



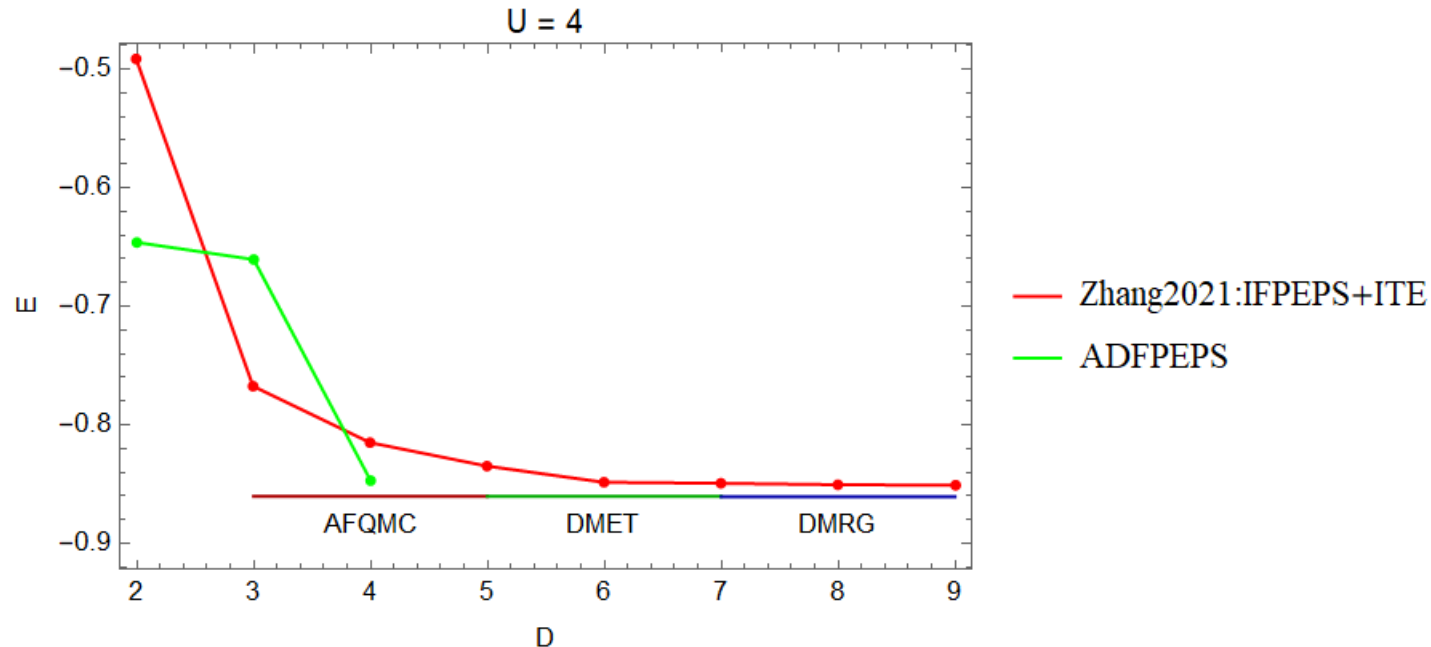
$U=4$ half-filled



$U=4$
 $\uparrow=0.675138$
 $\downarrow=0.325477$



U=4 half-filled



- $D = 2, 4$ is lower, but $D = 3$ is higher in contrast with ITE
- AD optimization is not stable at end (maybe for insufficient χ)
- 0.02 higher than other method, may need large D
- $D \geq 6$ is hard to achieve without symmetry for large memory usage

$$D = 4 \quad 3 - 4h$$

$$D = 5 \quad 24h$$

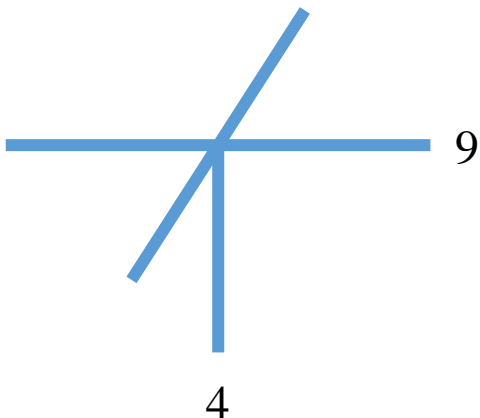
U1-tensor block division

- $Z2 \rightarrow U1$

$$\begin{array}{c} 0 \\ 1 \end{array} \begin{array}{cc} 0 & 1 \end{array} \left(\begin{array}{ccc|ccc} \cdot & \cdot & \cdot & & & \\ \cdot & \cdot & \cdot & & & \\ \cdot & \cdot & \cdot & & & \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ & & & \cdot & \cdot & \cdot \\ & & & \cdot & \cdot & \cdot \\ & & & \cdot & \cdot & \cdot \end{array} \right)$$

$$\begin{array}{c} 0 \\ 1 \\ 2 \end{array} \begin{array}{ccc} 0 & 1 & 2 \end{array} \left(\begin{array}{ccc|ccc|ccc} \cdot & \cdot & \cdot & & & & & & & \\ \cdot & \cdot & \cdot & & & & & & & \\ \cdot & \cdot & \cdot & & & & & & & \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & & & & \\ & & & \cdot & \cdot & \cdot & & & & \\ & & & \cdot & \cdot & \cdot & & & & \\ & & & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \\ & & & & & & \cdot & \cdot & \cdot & \\ & & & & & & \cdot & \cdot & \cdot & \\ & & & & & & \cdot & \cdot & \cdot & \end{array} \right)$$

- iPEPS

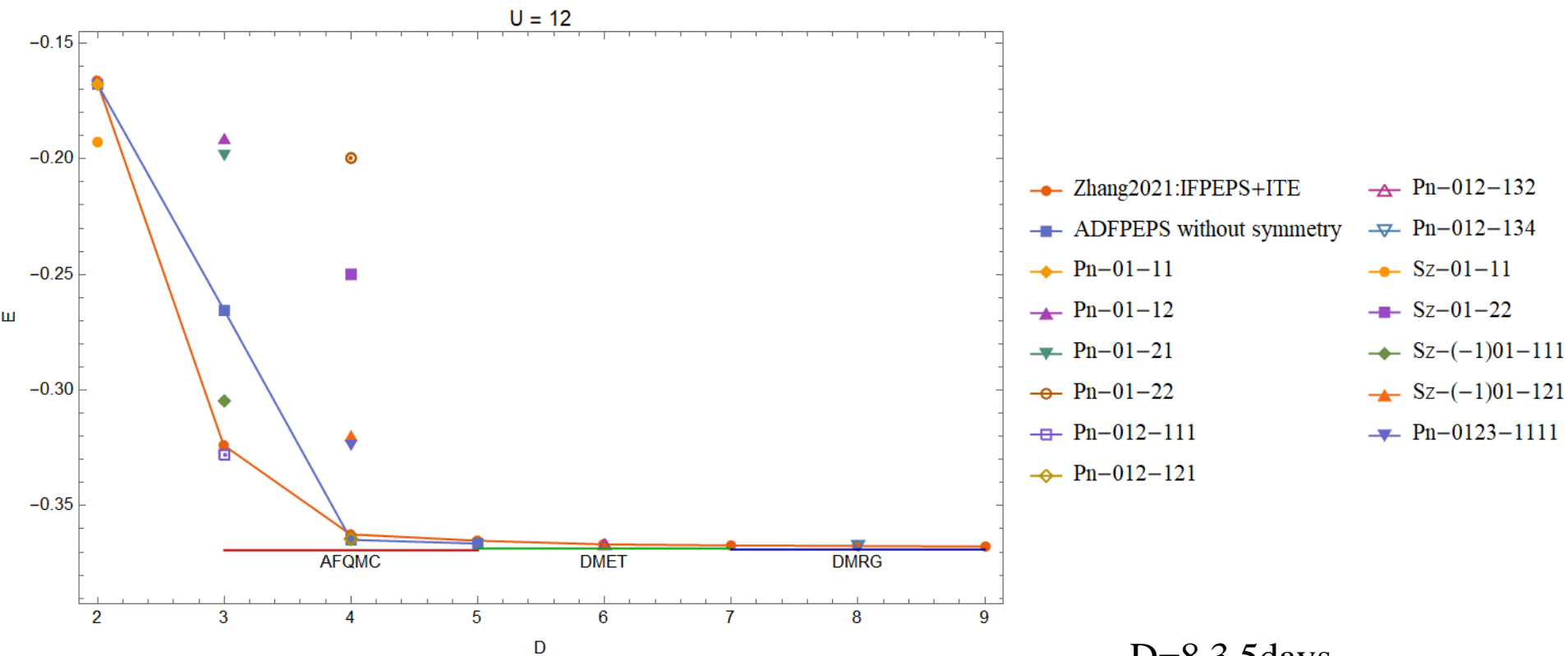


state	S_z	P_n
0	0	0
\uparrow	+1	1
\downarrow	-1	1
$\uparrow\downarrow$	0	2

$$\begin{array}{cccc} & 0 & 1 & 2 & 3 \\ 9 & = & 3 + 3 + 3 \\ & = & 1 + 4 + 3 + 1 \end{array}$$

Physical bond division is decided by Hamiltonian, but virtual bond division is arbitrary.

U=12 half-filled use U1-symmetry



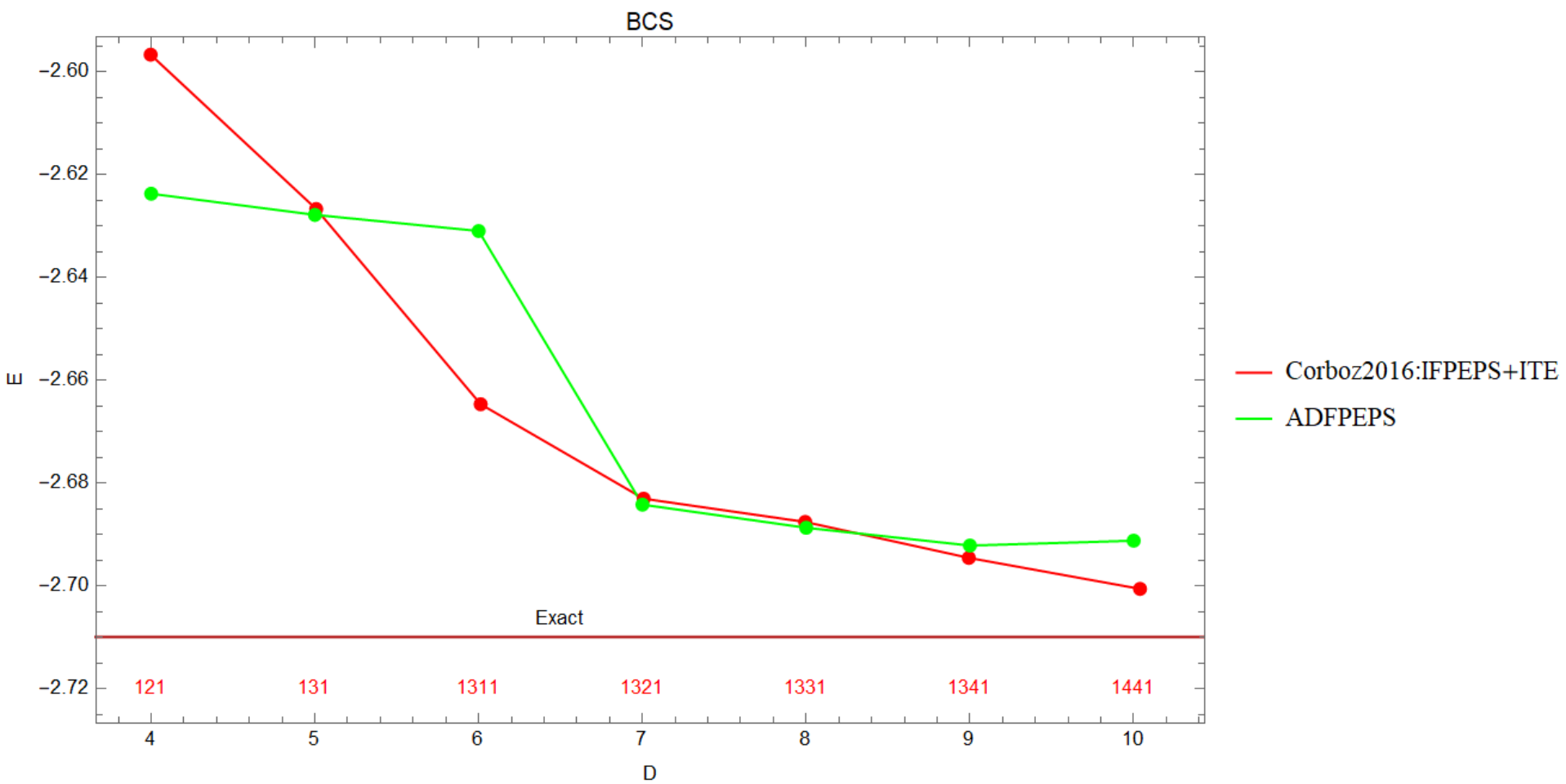
D=8 3.5days

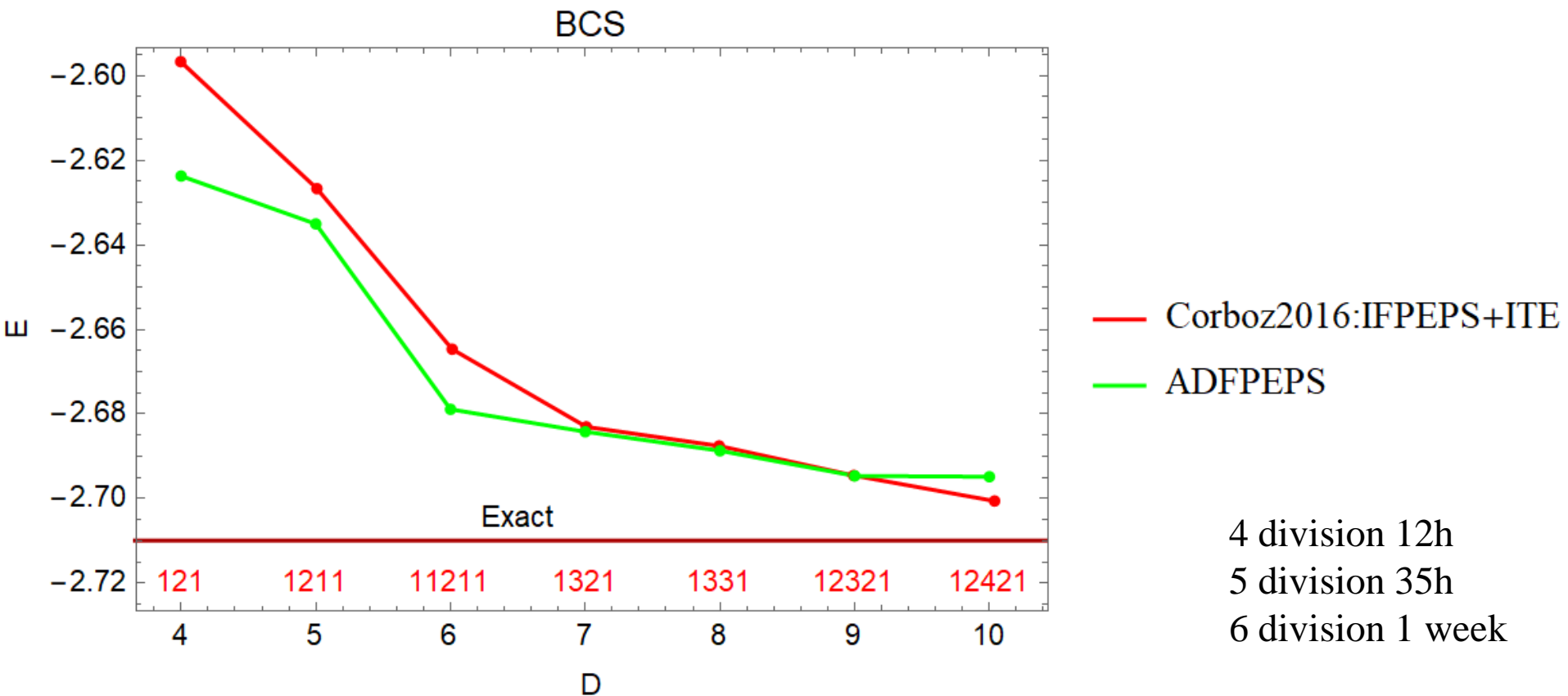
More division is better than bigger dimension

Mean field BCS

$$\hat{H} = -t \sum_{\langle i,j,\sigma \rangle} (\hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} + \text{H.c.}) + \sum_{\langle i,j \rangle} \gamma_{ij} (\hat{c}_{i\uparrow}^\dagger \hat{c}_{j\downarrow}^\dagger - \hat{c}_{i\downarrow}^\dagger \hat{c}_{j\uparrow}^\dagger + \text{H.c.})$$

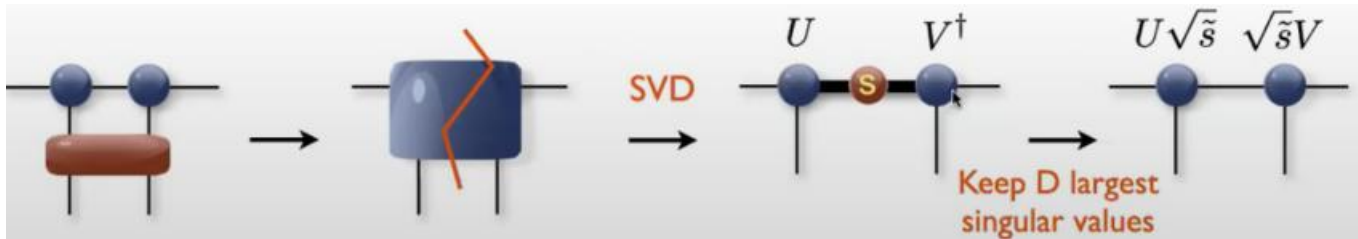
- Can be exact solved by diagonalized
- Only need 1×1 cell





- Small division converges quickly, more division is better but hard to achieve
- Difficulty for fermion model is ability expression rather than ability of optimization
- For large D , the difficulty is how to divide the U1 block

divide the U1 block use SVD



$$A = \begin{pmatrix} \cdot & \cdot & \cdot & & \\ \cdot & \cdot & \cdot & & \\ \cdot & \cdot & \cdot & & \\ & \cdot & \cdot & \cdot & \\ & \cdot & \cdot & \cdot & \cdot \\ & & \cdot & \cdot & \cdot \\ & & & \cdot & \cdot \\ & & & & \cdot \end{pmatrix} \times \begin{pmatrix} \cdot & & & & \\ & \cdot & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \cdot \\ & & & & & \cdot \\ & & & & & & \cdot \\ & & & & & & & \cdot \end{pmatrix} \times \begin{pmatrix} \cdot & \cdot & \cdot & & \\ \cdot & \cdot & \cdot & & \\ \cdot & \cdot & \cdot & & \\ & \cdot & \cdot & \cdot & \\ & \cdot & \cdot & \cdot & \cdot \\ & & \cdot & \cdot & \cdot \\ & & & \cdot & \cdot \\ & & & & \cdot \end{pmatrix}$$

U
 S
 V^\dagger

- Merge two legs of U1 tensor will **increase** number of blocks.
- Sort S from different blocks, keep largest D value may **decrease** number of blocks.
- So after split to origin leg, block would be **divided** differently from original one.

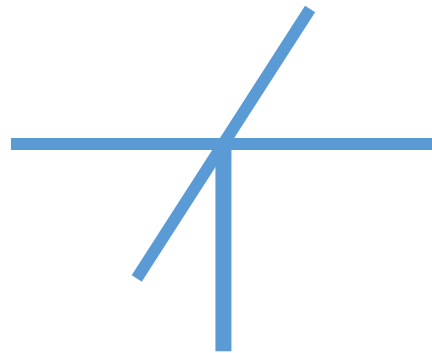
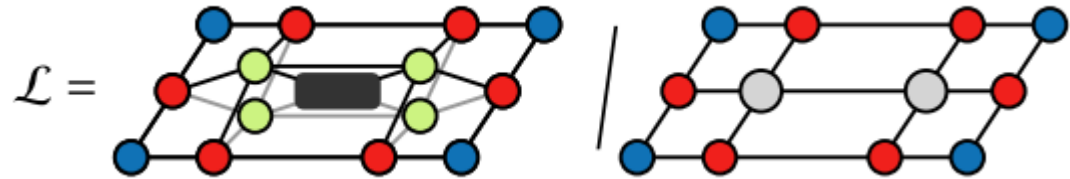
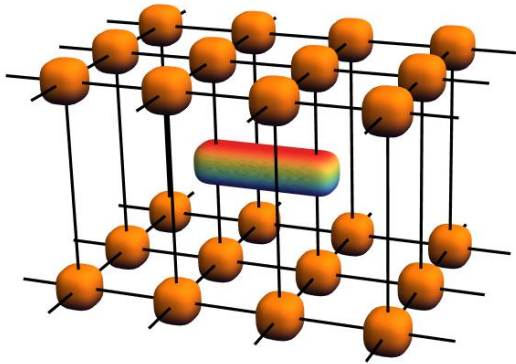
Open question

- Optimize U1 iPEPS block with AD
- Divide U1 boundary MPS block in VUMPS
- Optimize U1 code for large D with more division blocks

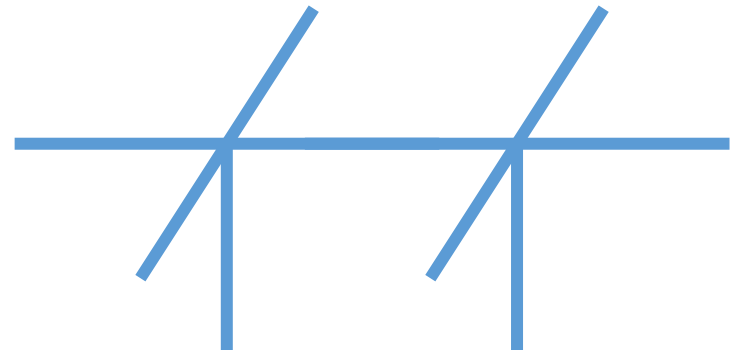
- Momentum space tensor network for fermion?

Optimize U1 iPEPS block with AD

- Merge two iPEPS \rightarrow ansatz: two-site iPEPS



16



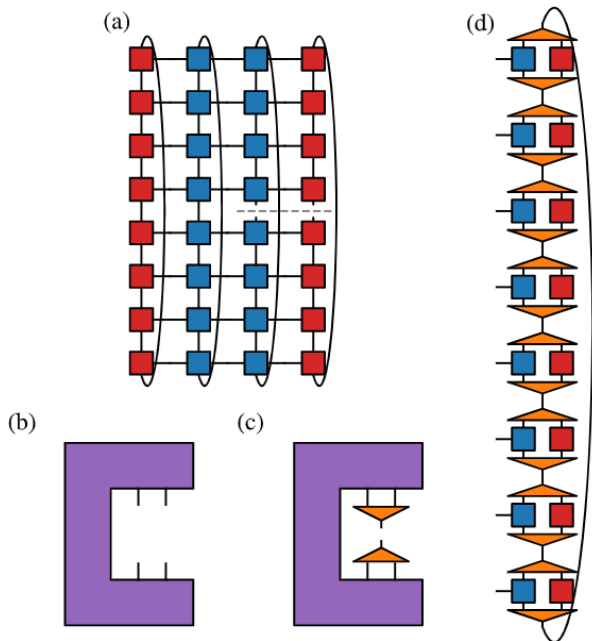
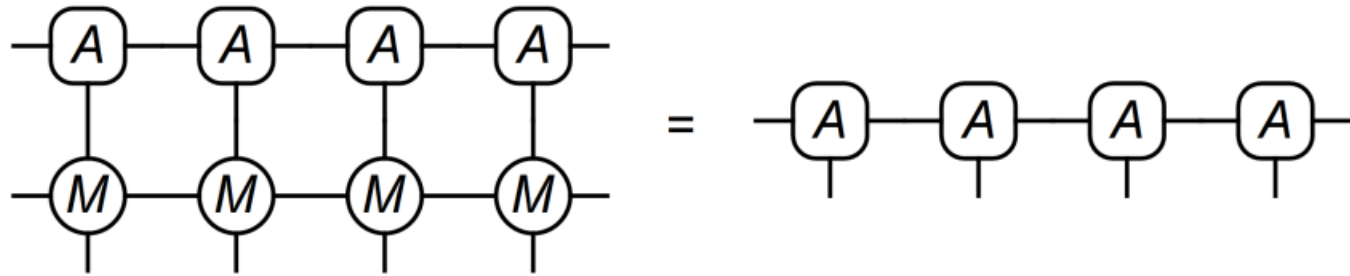
4

4

12

Divide U1 MPS block in VUMPS

- Last step TEBD



- Easy to realize but increased computation cost

3-leg FL,FR \rightarrow 4-leg FL,FR

- use this trick after one AD Optimization

Thank you for listening!

Q&A?