

PEPS contractions, needed for computing expectation values, are #P hard when performed exactly. Various schemes for performing PEPS contractions approximately have been and continue to be proposed (this is a question at the cutting edge of current research). We describe three schemes: (i) using purely MPS methodology, (ii) using 2D transfer matrices; and (iii) using corner transfer matrices.

1. PEPS via finite-size MPS

Goal: reduce PEPS computations to familiar MPS computations ('reuse code').

[Verstraete2004] (proof-of-principle application: spin 1/2 Heisenberg model on 4x4 lattice)

[Murg2007] Murg, Verstraete, Cirac: (2D hard-core bosons = 2D XY model in uniform field)

Strategy for PEPS ground state search:

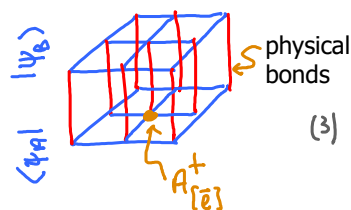
Trotter time step \rightarrow initial PEPS

(i) imaginary time evolution: $|\psi_B\rangle = e^{-H\tau} |\psi_A\rangle \quad (D \rightarrow Dd^2) \quad (1)$

(ii) truncate back to D , via variational compression (one tensor at a time) of

$$K = \|\underbrace{|\psi_A\rangle}_{\text{compressed PEPS}} - \underbrace{|\psi_B\rangle}_{\text{time-evolved PEPS}}\|^2 = \langle \psi_A | \psi_A \rangle + \langle \psi_B | \psi_B \rangle - 2 \text{Re} \langle \psi_A | \psi_B \rangle \quad (2)$$

$$0 = \frac{\partial K}{\partial A_{[i]}^+} \Rightarrow N_{[i]}^- A_{[i]}^- = W_{[i]}^- , \text{ yields new } A_{[i]}^-$$

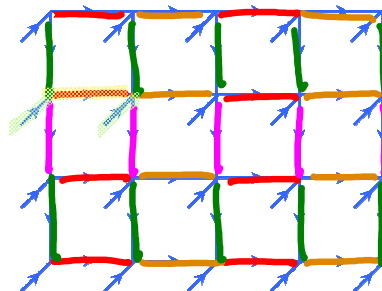


Computation of $N_{[i]}^-, W_{[i]}^-$ can be performed by repeated application of MPOs to MPS !

(i) Trotter time evolution [Murg2007, Sec. IV]

$$H = H_{eh} + H_{oh} + H_{ev} + H_{ov} \quad (1)$$

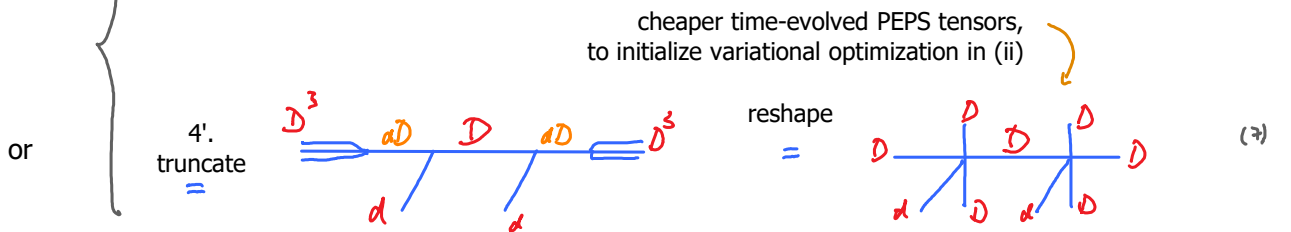
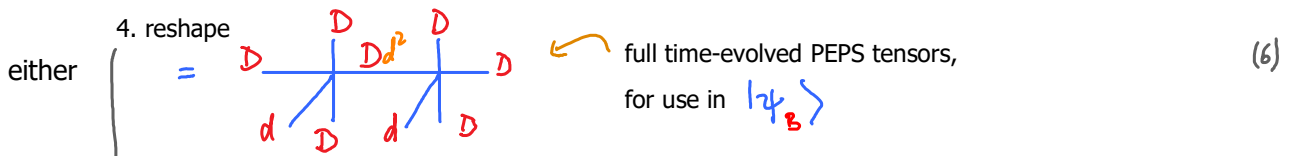
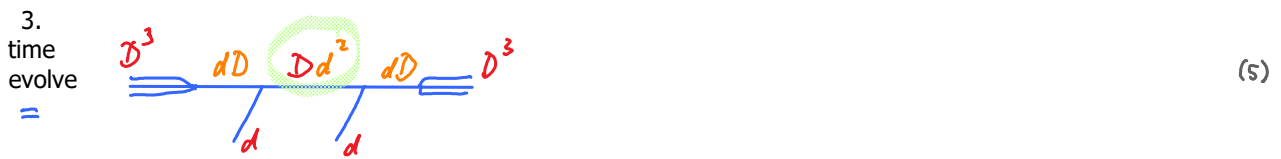
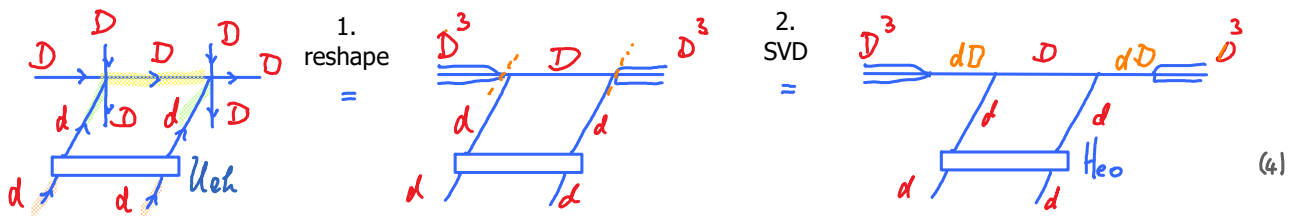
e : even, o : odd, h : horizontal, v : vertical



$$e^{-H\tau} = \underbrace{e^{-H_{eh}\tau}}_{\text{U}_{eh}} e^{-H_{oh}\tau} e^{-H_{ev}\tau} e^{-H_{ov}\tau} + \mathcal{O}(\tau^2) \quad (2)$$

Apply one such factor to all corresponding bonds: $|\psi_B\rangle = \prod_{eh} e^{-H_{eh}\tau} |\psi_A^o\rangle \quad (3)$

Following bad habit of PEPS literature, we mostly drop arrows in our diagrams below...



1. Reshape 5-leg PEPS tensors into 3-leg MPS form.

2. (suggested by A. Weichselbaum) SVD to project D^3 -dim space onto Dd (making step 3. cheaper)

3. Time-evolve. This causes increase in dimension of central bond from D to Dd^2 .

4. Reshape 3-leg MPS tensors back into 5-leg PEPS form. These define the time-evolved PEPS.

Its central bond dimension must still be decreased, using variational compression in step (ii) below.

Comment: when reshaping, a big D^3 -space is split into the direct product of 3 smaller D -spaces.

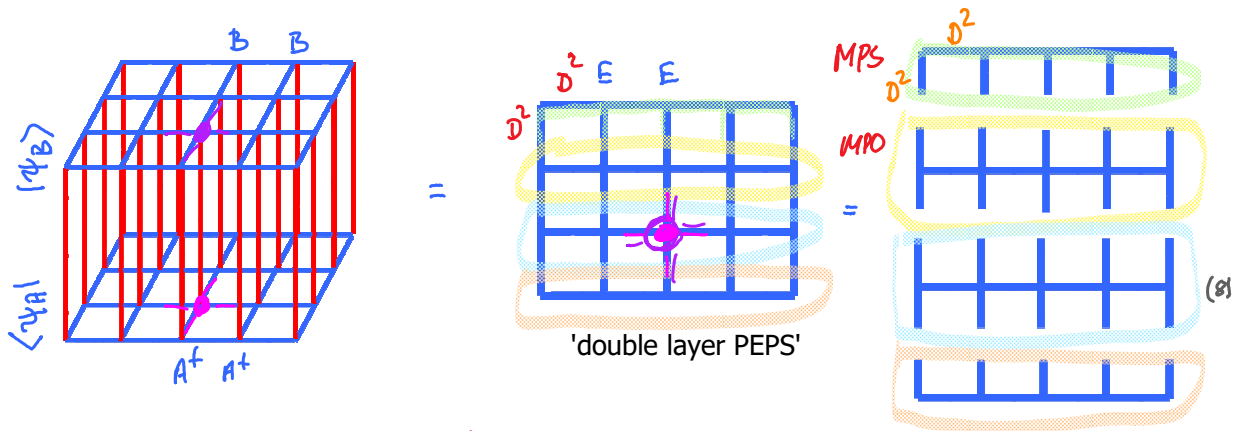
This involves much freedom, which should be used to minimize entanglement. That is partially achieved as a byproduct of the variational compression of (ii), but more direct strategies, involving 'disentanglers' (to be discussed in a subsequent lecture), might lead to further improvements.

4'. To get a good starting point for (ii), one may use SVD to truncate central bond from Dd to D .

(Note: performing only this truncation, without subsequently performing (ii), would yield a suboptimal result, since this truncation includes no information about environment of open legs. By contrast, such information is included during variational compression (ii), hence the latter is essential.

(ii) Variational compression [Meng2007, Sec. II]

To compute the overlaps in K , we need objects like $\langle \psi_A | \psi_B \rangle$, $\langle \psi_A | \psi_A \rangle$



where $E_{A+B} = \sum_{\sigma} A^{\sigma\dagger} B_{\sigma}$ is 4-leg tensor with bond dimension D^2

View first and last rows of double-layer PEPS as MPSs, and rows in between as MPOs.

Each application of an MPO increases bond dimension:



Iteratively apply MPOs. In last step, apply  to obtain  (10) which looks like a contracted MPS.

For $\frac{\partial K}{\partial A_{[\vec{r}]}}$ one can proceed similarly, but with one A^\dagger missing from lower (bra) layer, and correspondingly some open legs.

In this manner, compute $N_{[\vec{r}]}$ and $W_{[\vec{r}]}$, then solve $N_{[\vec{r}]} A_{[\vec{r}]} = W_{[\vec{r}]}$ for $A_{[\vec{r}]}$. (11)

Sweep until convergence, the converged result yields $|\psi_A\rangle$. Then return to (i), with next Trotter gate.

Results for ground state energy [Murg2007]

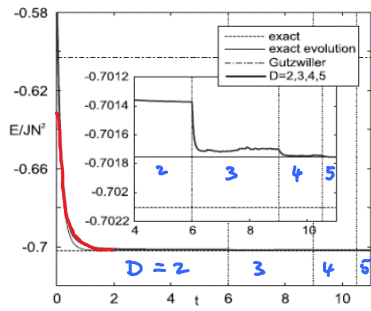


FIG. 3. Energy as a function of time for the imaginary time evolution of the system of hard-core bosons on a 4×4 lattice. The evolutions are performed sequentially with PEPS of virtual dimension $D=2$, $D=3$, $D=4$, and $D=5$. The times at which D is increased are indicated by vertical lines. For comparison, the exact ground-state energy, the exact imaginary time evolution, and the energy of the optimal Gutzwiller ansatz are included.

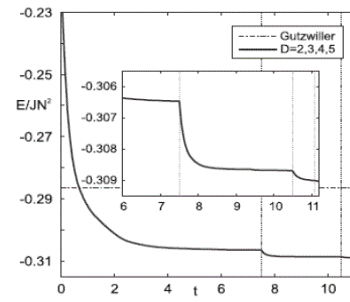


FIG. 4. Energy as a function of time for the imaginary time evolution of the system of hard-core bosons on a 11×11 lattice. The evolutions are performed sequentially with PEPS of virtual dimension $D=2$, $D=3$, $D=4$, and $D=5$. The times at which D is increased are indicated by vertical lines. For comparison, the energy of the optimal Gutzwiller ansatz is included.

Accuracy and numerical efficiency

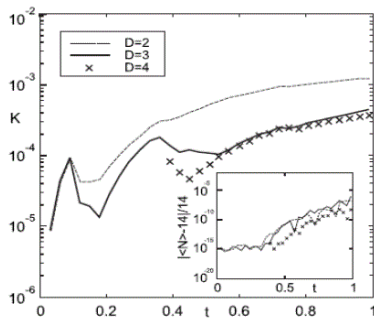


FIG. 9. Distance K between the time-evolved state and the state with reduced virtual dimension. The virtual dimensions $D=2$, $D=3$, and $D=4$ are included. The distance is plotted for the evolution of a Mott distribution with $N=14$, as explained in Fig. 8. From the inset, the deviation of the particle number from the value 14 can be gathered.

$$K \approx 10^{-3} \text{ for } D=2$$

$$10^{-4} \text{ for } D=3, 4$$

$$|\langle \hat{N} \rangle - 14| \approx 10^{-5}$$

particle number, should be conserved

[computations were performed on a workstation with 3.0 GHz Intel-Xeon processor]

Main bottleneck: # multiplications $\sim O(D^{12})$, memory $\sim O(D^8)$

\Rightarrow 11 x 11 lattice with $D=5$ took 55 hours for 1 time step, and required 2 GB memory.

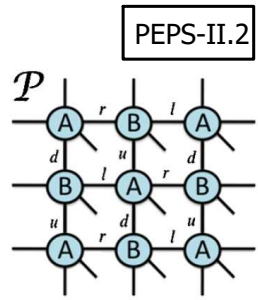
'going beyond $D=5$ is difficult at the moment' (2007).

By now, $D \lesssim 20$ is possible when exploiting non-Abelian symmetries.

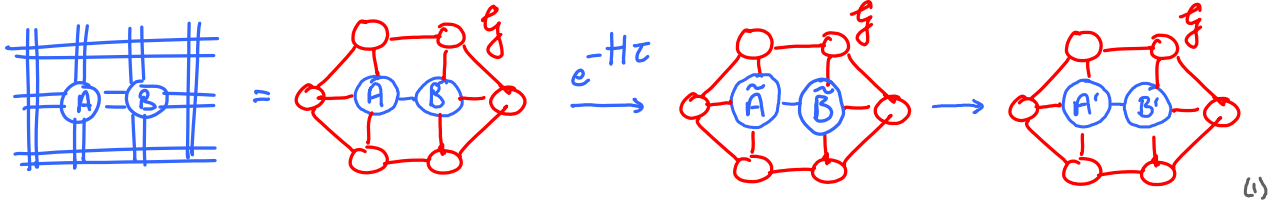
2. Infinite-size PEPS (iPEPS)

[Jordan2008]: Jordan, Orus, Vidal, Verstraete, Cirac, for 2D quantum Ising model

Goal: exploit translational invariance, by borrowing tools from iTEBD.



Strategy: represent entire lattice in terms of just two tensors, A, B



double-layer PEPS

system-environment

time-evolved version

variational compression

(i) Choose specific bond between two tensors, A, B forming unit cell.

Compute 'environment', $g(A, B)$ built from A, B too, using transfer matrix tools and iTEBD.

(ii) Perform imaginary time evolution on specified bond between active sites, $A, B \rightarrow \tilde{A}, \tilde{B}$

(iii) Use variational compression, involving $\tilde{A}, \tilde{B}, g(A, B)$, to compress \tilde{A}, \tilde{B} down to A', B' .

Then move to next bond and iterate...

Overall structure

Exploit translational invariance:

Use two-site 'active' unit cell,

for bipartite covering of lattice.

Then iPEPS depends on $\mathcal{O}(d^4)$ coefficients.

Single-layer iPEPS: $\mathcal{P}(A, B)$

Double-layer iPEPS: $\mathcal{E}(a, b)$

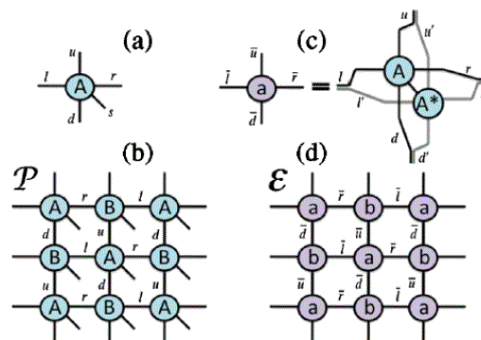


FIG. 1 (color online). Diagrammatic representations of (a) a PEPS tensor A_{sudlr} with one physical index s and four inner indices u, d, l and r ; (b) local detail of the tensor network \mathcal{P} for an iPEPS. Copies of tensors A and B are connected through four types of links; (c) reduced tensor a of Eq. (2); and (d) local detail of the tensor network \mathcal{E} .

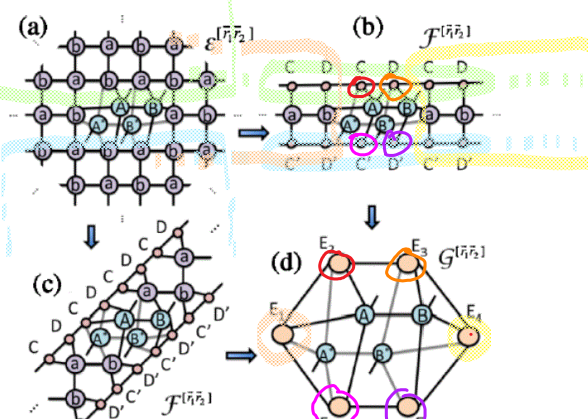
(i) Contract out environment for specified bond

e.g. bond r

Approximate the double-layer iPEPS, \mathcal{E} , by an infinite strip,

$$F = F(C, D, C', D', a, b) \quad (z)$$

(either horizontally, vertically, or diagonally), surrounding the active sites.



(i) Contract out environment for specified bond

e.g. bond r

Approximate the double-layer iPEPS, \mathcal{E} , by an infinite strip,

$$F = F(C, D, C', D', a, b) \quad (2)$$

(either horizontally, vertically, or diagonally), surrounding the active sites.

Approximate strip F by 'environment', encoded in six E-tensors,

$$g = g(E_1, \dots, E_6) \quad (3)$$

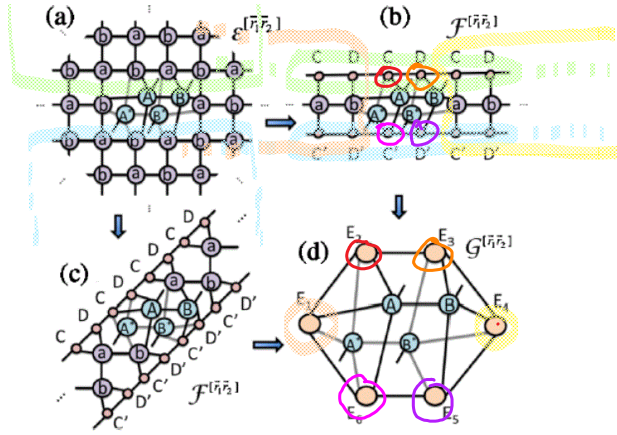


FIG. 2 (color online). The environment $\mathcal{E}^{[r_1, r_2]}$ for a link of type r is first approximated by an infinite strip $\mathcal{F}^{[r_1, r_2]}$ and then by a six-tensor network $\mathcal{G}^{[r_1, r_2]}$. These reductions can be performed according to either a vertical or horizontal scheme (b) or a diagonal scheme (c). Tensors $A, A^*, B,$ and B^* are not part of the environment.

Computation of strip F using transfer matrix methods

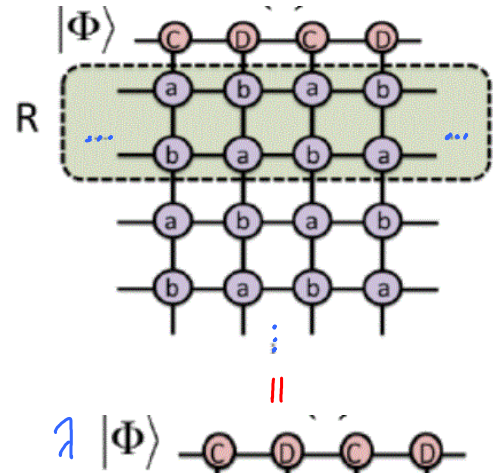
Define transfer matrix, R , built from two infinite rows of a, b :

Compute its dominant right eigenvector, $|\Phi\rangle$:

$$R|\Phi\rangle = \lambda|\Phi\rangle, \quad (4)$$

then $|\Phi\rangle \equiv$ (5)

represents entire 'upper half plane' (above active sites).



In practice: $|\Phi\rangle = \lim_{P \rightarrow \infty} \frac{R^P |\Phi_0\rangle}{\|R^P |\Phi_0\rangle\|} \quad (6)$

cost: $\mathcal{O}(\chi^3 D^6 + \chi^2 D^8 d)$

is computed using iTEBD methods, with R instead of e^{-Hz} as 'evolution' operator.

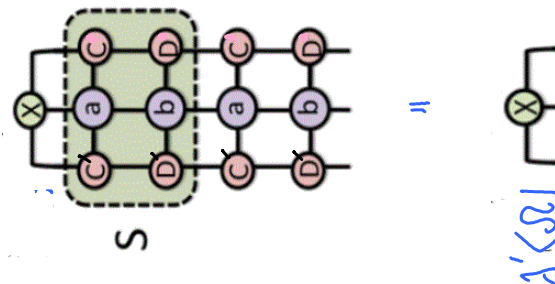
Similarly: compute dominant left eigenvector, $\langle\Phi| R = \lambda\langle\Phi| \quad (7)$

then $\langle\Phi| \equiv$ represents lower half-plane. (8)

Next, define transfer matrix S from the six tensors a, b, C, D, C', D' :

Compute its dominant left eigenvector, $\langle\Omega|$

$$\langle\Omega| S = \lambda'\langle\Omega| \quad (9)$$



then $\langle\Omega| \equiv$ represents left part of strip. (10)

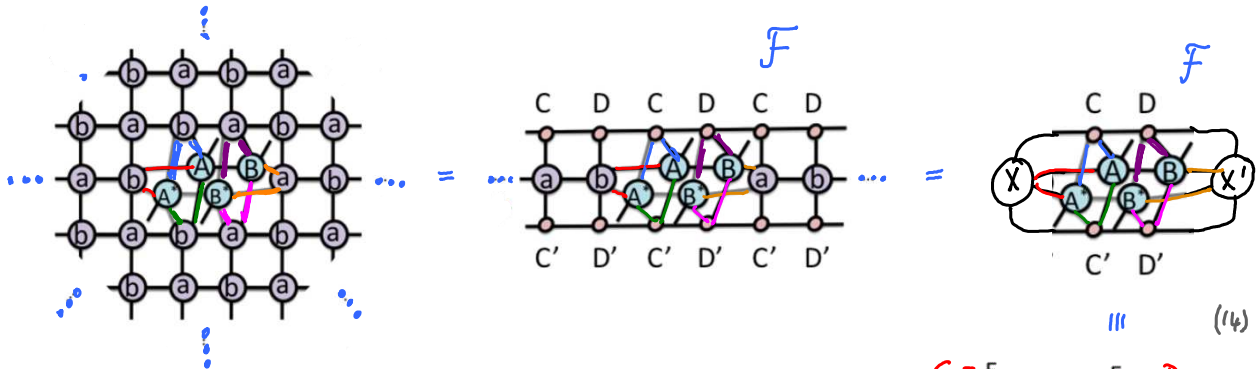
Similarly: compute its dominant right eigenvector. $S|\Omega\rangle = \lambda'|\Omega\rangle$

then $\langle \psi | \equiv \text{---} \bigcirc \text{---}$ represents left part of strip. (10)

Similarly, compute its dominant right eigenvector, $S | \Omega \rangle = \lambda' | \Omega \rangle$, (11)

then $| \Omega \rangle \equiv \text{---} \bigcirc \text{---}$ represents right part of strip. (12)

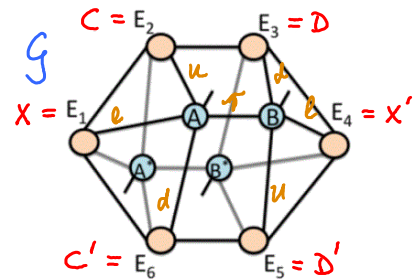
Construction of E-tensors representing environment (e.g. for r-bond)



Note: 'open up' those double-legs from double-layer tensors a, b (and C, D, C', D', X, X') which are connected with single-layer tensors A, B, A*, B*.

Thus, environment (computed from A, B) is represented by

$$g(E_1, E_2, E_3, E_4, E_5, E_6) = g(X, C, D, X', D', C') \quad (15)$$



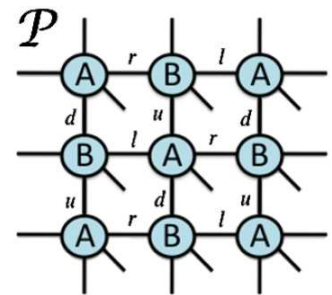
(ii) Imaginary time evolution of active basis sites

$$H = H_l + H_u + H_d + H_r \quad (16)$$

$$e^{-H\tau} = \underbrace{e^{-H_l\tau}}_{\prod_i g_i} e^{-H_u\tau} e^{-H_d\tau} e^{-H_r\tau} \quad (17)$$

$$\prod_i g_i = \prod_i e^{-h_i} \quad (18)$$

product over all left legs \uparrow \uparrow 'gate' acting only on a left-leg



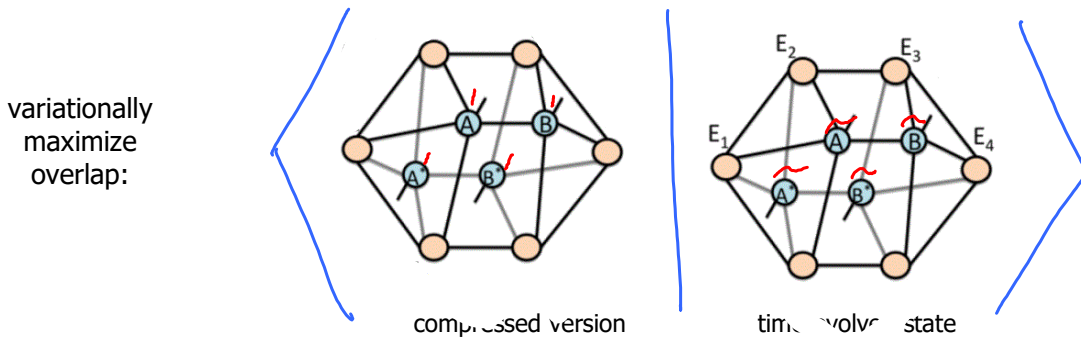
Time-evolve, e.g. for r-bond:

$$\text{---} \bigcirc \text{---} \text{---} \bigcirc \text{---} = \text{---} \bigcirc \text{---} \text{---} \bigcirc \text{---} \quad (19)$$

(PEPS-II.1.6)

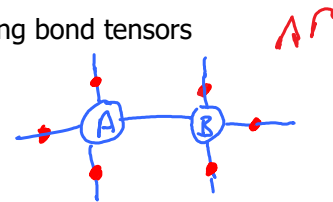
(iii) Compress \tilde{A}, \tilde{B} to A', B' via variational compression

Finding new tensors, A', B' (with bond dimensions D), which maximize overlap with time-evolved state.



The updated A', B' define the new iPEPS. Move to next bond, and iterate steps (i) to (iii) until convergence.

- 'Full update': recompute environment after each time step (most accurate, most expensive)
- 'Fast full update': use same environment for several time steps (less accurate, faster)
- 'Simple update': don't compute full environment, but approximate it using bond tensors (least accurate, but often still reasonable, by far the fastest)



Results for transverse field Ising model

[Jordan2008]

$$H(\lambda) = - \sum_{\langle i,j \rangle} \sigma_i^z \sigma_j^z - \lambda \sum_i \sigma_i^x$$

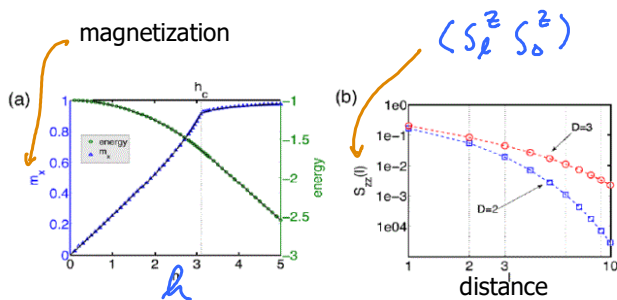


FIG. 4 (color online). (a) Transverse magnetization m_x and energy per site e as a function of the transverse magnetic field h . The continuous line shows series expansion results (to 26th and 16th order in perturbation theory) for h smaller and larger than $h_c \approx 3.044$ [13]. Increasing D leads to a lower energy per site e . For instance, at $h = 3.1$, $e(D = 2) \approx -1.6417$ and $e(D = 3) \approx -1.6423$. (b) Two-point correlator $S_{zz}(l)$ near the critical point, $\lambda = 3.05$. For nearest neighbors, the correlator quickly converges as a function of D , whereas for long distances we expect to see convergence for larger values of D .

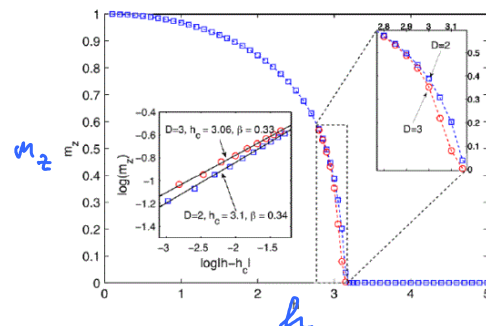
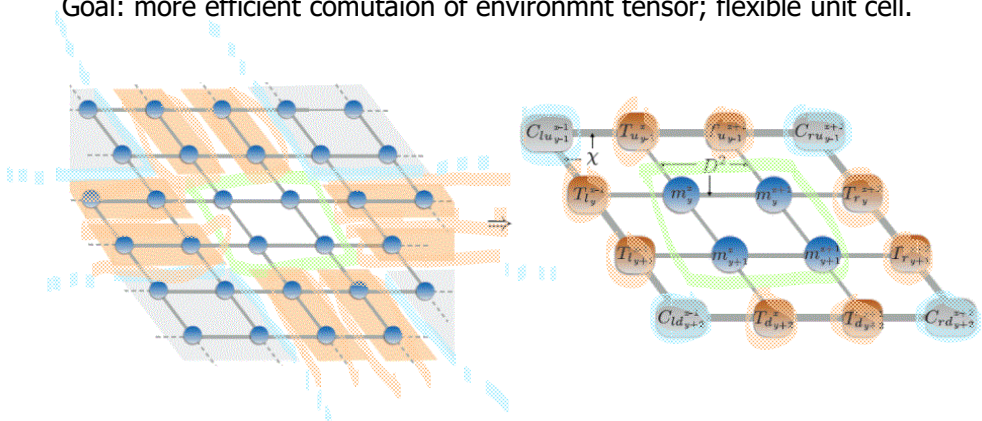


FIG. 5 (color online). Magnetization $m_z(\lambda)$ as a function of the transverse magnetic field λ . Dashed lines are a guide to the eye. We have used the diagonal scheme for $(D, \chi) = (2, 20)$, $(3, 25)$ and $(4, 35)$ [15] (the vertical or horizontal scheme leads to comparable results with slightly smaller χ). The inset shows a log plot of m_z versus $|\lambda - \lambda_c|$, including our estimate of λ_c and β . The continuous line shows the linear fit.

Goal: more efficient computation of environment tensor; flexible unit cell.



Figures from [Bruognolo2020]

Original idea of 'corner transfer matrix': [Baxter1968], [Baxter1978]

First application in tensor-network context: [Nishino1996], with Okunishi: 'corner transfer RG'

In PEPS context:

[Orus2009], with Vidal: proof-of-principle for 2D quantum Ising model

[Corboz2010], with Jordan, Vidal: spinless fermions with nearest-neighbor interactions; t-J model

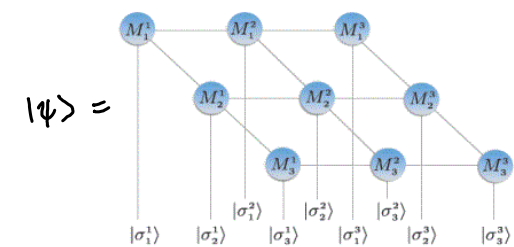
[Corboz2011], with White, Vidal, Troyer: stripes in t-J model

[Corboz2014], with Rice, Troyer: t-J model (competitive results!)

[Ponsioen2019], with Chung, Corboz (extended 2D Hubbard model)

[Bruognolo2020], with Li, von Delft (PEPS review for beginners) [notation of this review is used below]

Notation

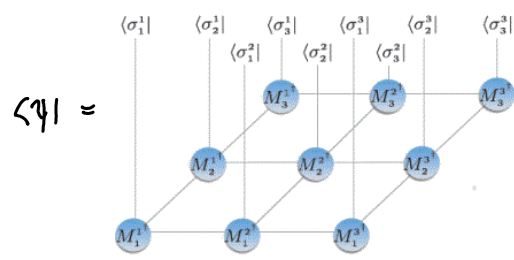


local tensor

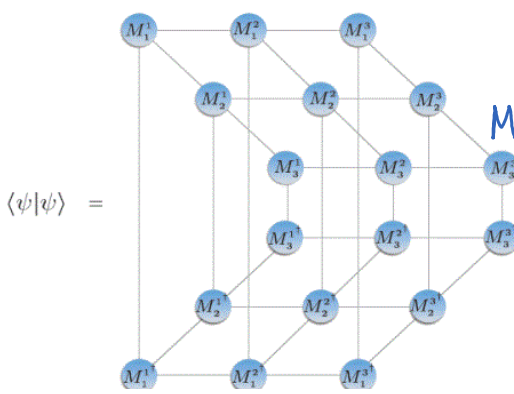
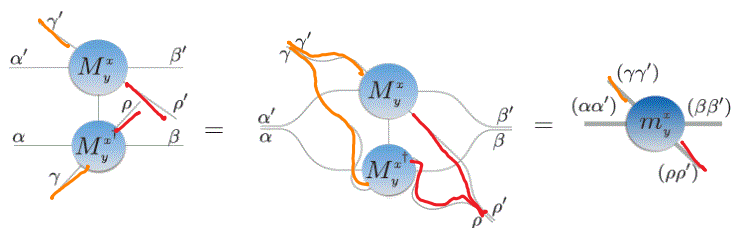
$$M_{\alpha\beta\gamma\rho}^{[\sigma_y^x]} = \alpha \text{---} \text{---} \text{---} \text{---} \beta \text{---} \text{---} \text{---} \text{---} \rho$$

conjugate local tensor

$$M_{\alpha\beta\gamma\rho}^{[\sigma_y^x]\dagger} = \alpha \text{---} \text{---} \text{---} \text{---} \beta \text{---} \text{---} \text{---} \text{---} \rho = M_{\rho\gamma\beta\alpha}^{*[\sigma_y^x]}$$

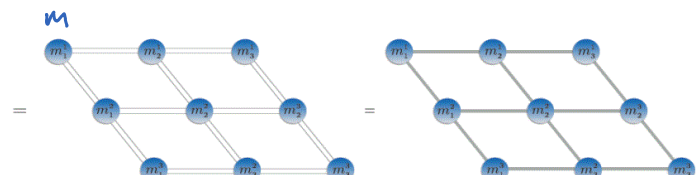


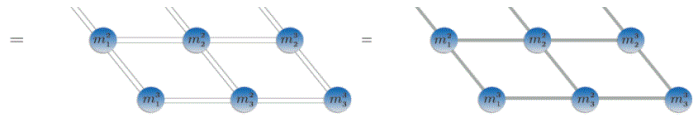
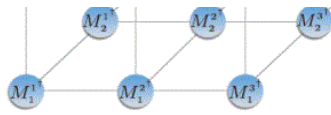
contracting out physical indices



detail relevant for fermions:

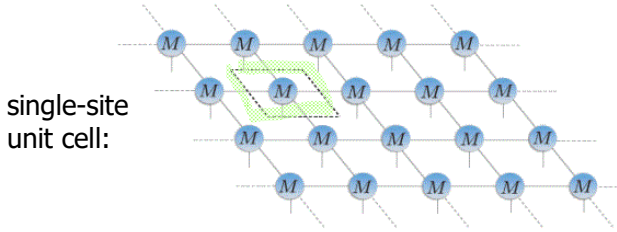
$$M_y^{\dagger} = \bar{M}_y^{\dagger}$$



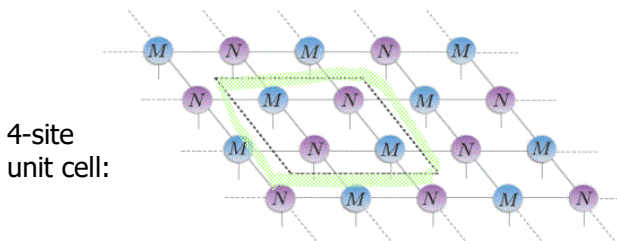


Variable unit cell [Corboz 2011]

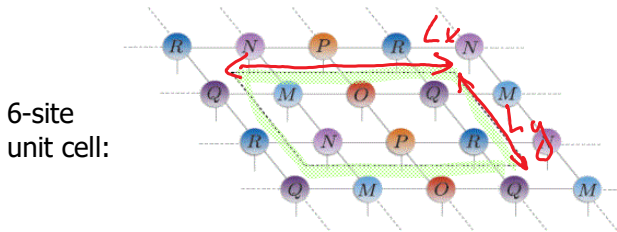
For systems with spontaneous symmetry breaking, one needs to vary size of unit cell, to accommodate various types of order...



cannot capture spontaneous symmetry breaking



compatible with antiferromagnetic order



could begin to show onset of striped order

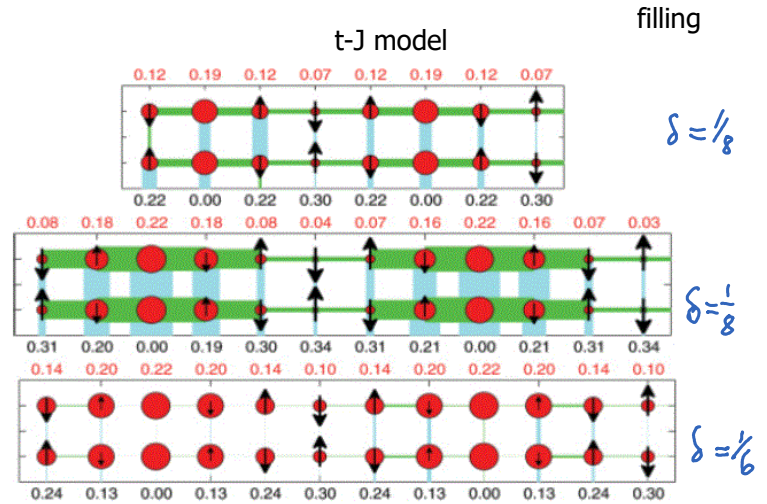
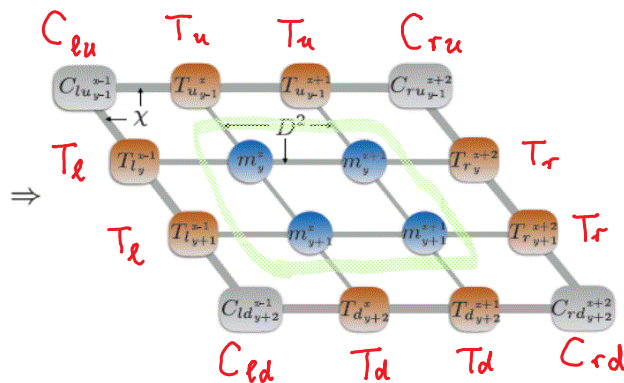
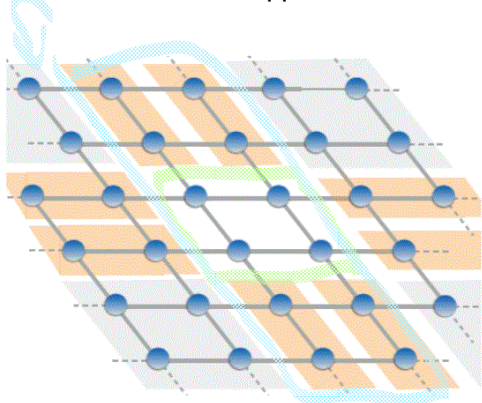


FIG. 4. (Color online) Examples of stripes running in the vertical direction obtained from various iPEPS simulations. Each panel shows one unit cell of the infinite lattice. The diameter of the dots scales with the local hole density with average values given by the (upper) red numbers. The arrows represent the local magnetic moment with average magnitude given by the (lower) black numbers. There is a π -phase shift in the antiferromagnetic order between adjacent stripes. The width of a bond between two sites scales with the (singlet) pairing amplitude on the bond with a positive (green/dark gray) or negative (cyan/light gray) sign. A pattern with predominantly d -wave order is visible, with maximal pairing amplitude 0.01, 0.03, and 0.003 in

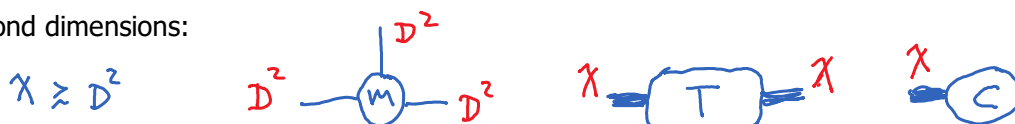
Corner transfer matrix approach: main idea



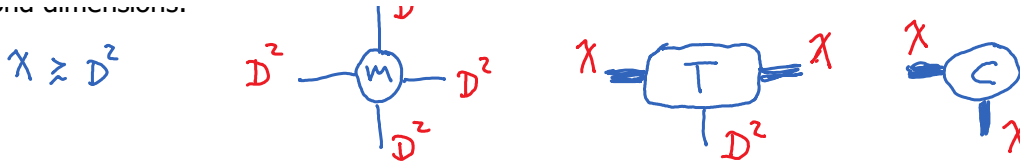
Represent environment of 2x2 'zooming window' in terms of 12 terms: tensors.

8 transfer tensors (T's), 4 corner tensor (C's). [Here: 2x2 unit cell. Generalization to larger ones is straightforward]

Bond dimensions:

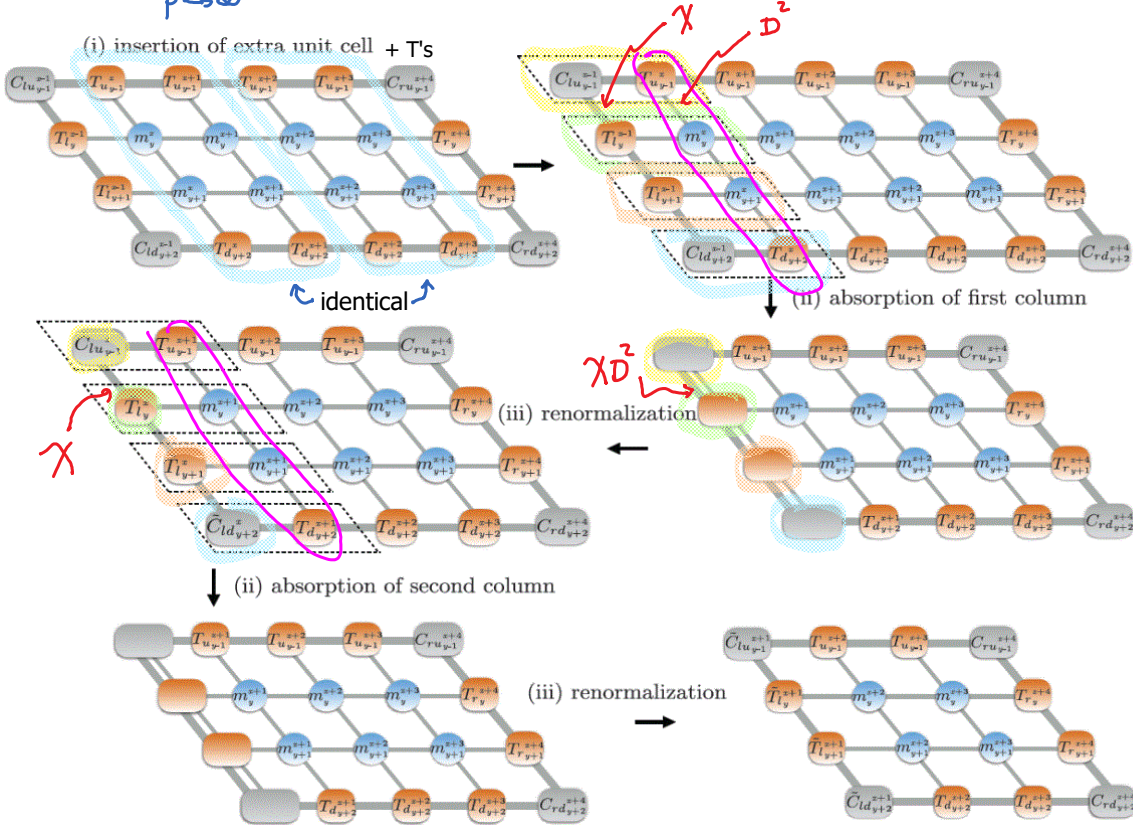


2D dimensions:

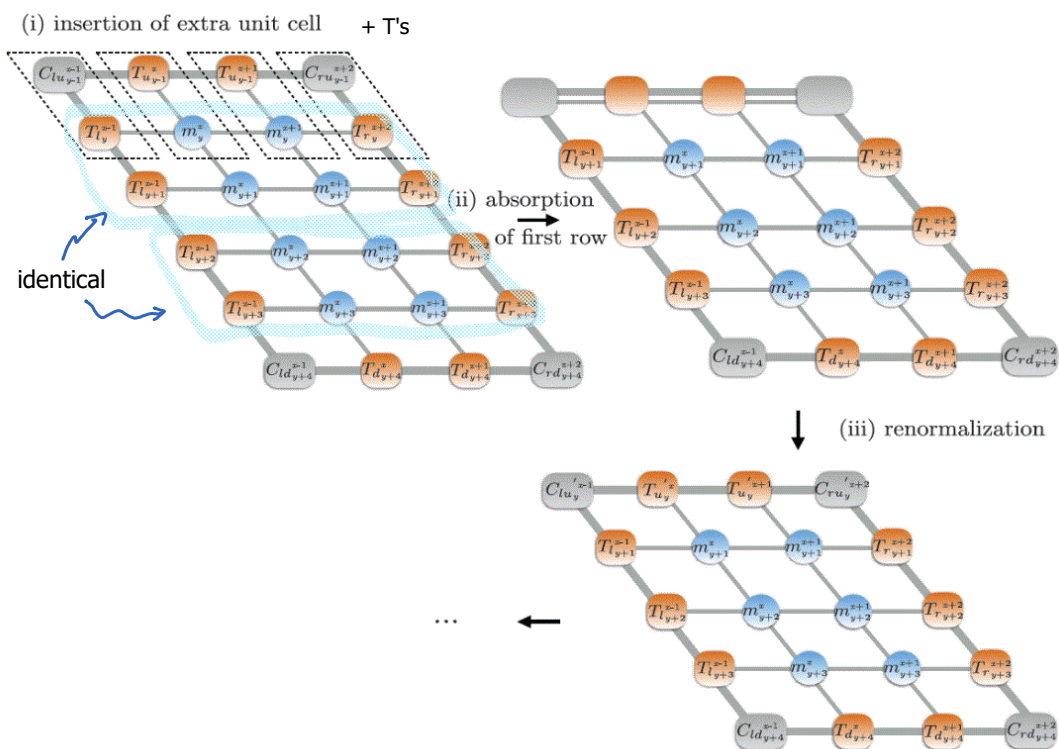


CTM coarse-graining move: insertion to the left

Instead of $\lim_{p \rightarrow \infty} R^p |\Phi\rangle$, repeatedly: (i) insert unit cell, (ii) absorb row, (iii) renormalize



CTM coarse-graining move: insertion at the top



One 'cycle' = insertion/absorption/renormalization at left, top, right, bottom.
 Repeat cycle numerous times until convergence (e.g. of local observables).

There is no clear 'optimal way' to perform the renormalization, due to lack of a canonical form.
 Trial and error, room for creativity!

see [Orus2009], [Corboz2010a], [Corboz2014-Supplement] = [Bruognolo2020]

Optimization of M-tensors in zoom-window

Most PEPS-workers use Trotter imaginary-time evolution, as proposed in [Jordan2008] for iPEPS.

$$|\psi_0\rangle = \lim_{\tau \rightarrow \infty} \frac{e^{-\tau H} |\psi\rangle}{\|e^{-\tau H} |\psi\rangle\|}$$

Factorize $e^{-\tau H}$ into nearest-neighbor two-site gates, etc.

After application of Trotter gate to a bond, truncation ('bond projection') is needed. Two options:

- full update, using CMT environment [most accurate, but computationally expensive]
- simple update, in spirit of iTEPD, with $\Lambda \Gamma$ treatment of neighboring bonds [less accurate (with systematic errors), much cheaper!]

Iteration scheme:

- start from random PEPS
- initially use large time steps, small bond dimensions
- gradually reduce time steps when SVD spectra converge, down to $\delta\tau \approx 10^{-5}$

Side remarks:

- Longer-ranged interactions are also possible. [Corboz2010], [Corboz2013]
- Alternative approaches: variational optimization of

$$\frac{\langle \psi_0 | H | \psi_0 \rangle}{\langle \psi_0 | \psi_0 \rangle}$$

[Corboz2016],
 [Vanderstraeten2016] 'gradient method'