Quantum simulation with string-bond states: Joining PEPS and Monte Carlo

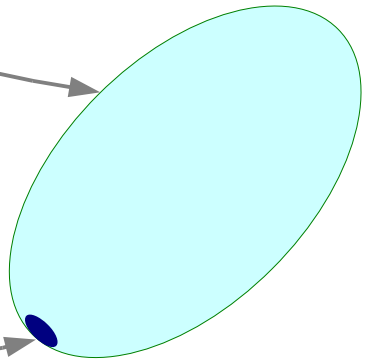
N. Schuch[1], A. Sfondrini[1,2], F. Mezzacapo[1],
J. Cerrillo[1,3], M. Wolf[1,4], F. Verstraete[5], I. Cirac[1]

[1] Max-Planck-Institute for Quantum Optics, Garching, Germany
[2] University of Padua, Italy
[3] University of Barcelona, Spain
[4] Niels-Bohr-Instititute, Copenhagen, Denmark
[5] University of Vienna, Austria

The key to quantum simulations

- Aim: Simulation of quantum many-body systems

Hilbert space dimension is huge:
exponentially in the number of spins.



But: The system is governed by a
local Hamiltonian – i.e. it can be
characterized by a *polynomial* number of parameters.

- Requirements on variational ansatz:
  - **it should capture the states of interest**
  - **it should allow for efficient computation of energies etc.**

Matrix Product States

- Matrix Product States (MPS):

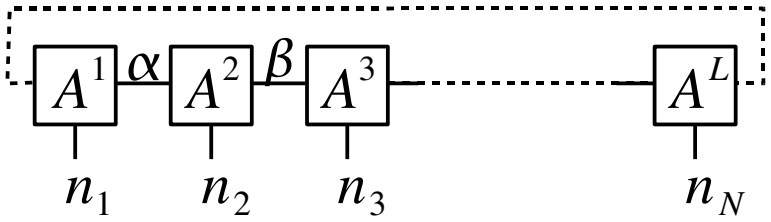
$D \times D$ matrices

$$|\psi\rangle = \sum_{n_1,\ldots,n_L} \mathrm{tr}\left[A^1_{n_1} A^2_{n_2} \cdots A^L_{n_L}\right] |n_1,\ldots,n_L\rangle$$

- Alternatively:

$$|\psi\rangle = \sum_{n_1,\ldots,n_L} c_{n_1,\ldots,n_L} |n_1,\ldots,n_L\rangle \text{ , with}$$

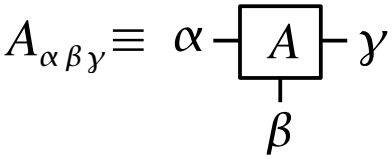
$c_{n_1,\ldots,n_L} \equiv$



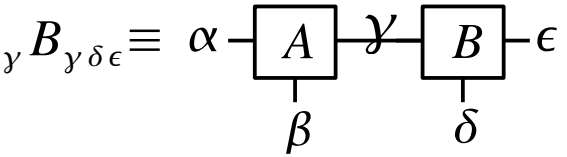
"Tensor network states"

Notation

$A_{\alpha\beta\gamma} \equiv$



$\sum_{\gamma} A_{\alpha\beta\gamma} B_{\gamma\delta\epsilon} \equiv$



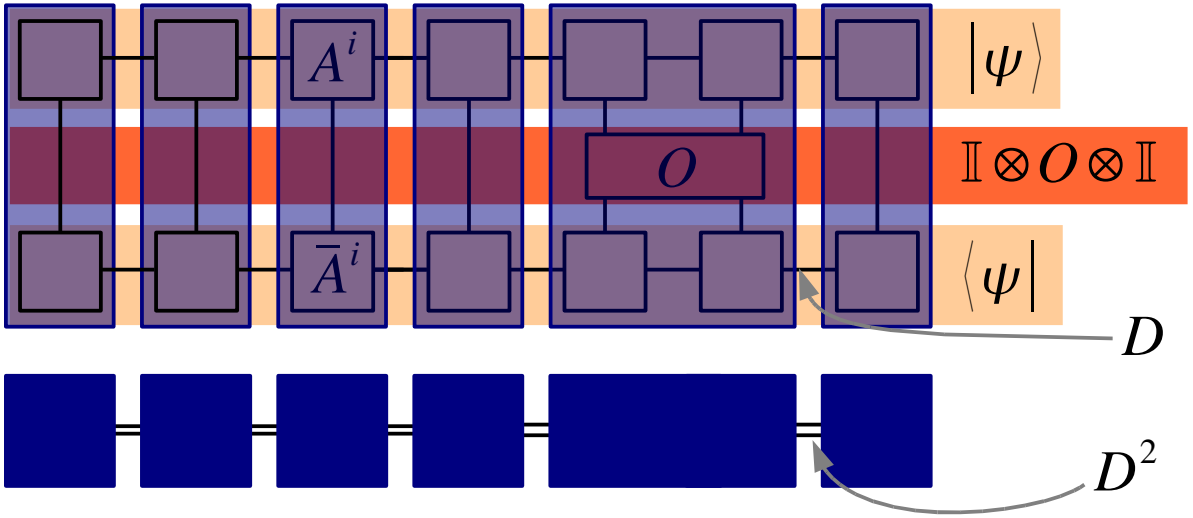
- MPS approximate well the ground states of gapped 1D Hamiltonians

[Hastings, JSTAT '07]

Computing expectation values on MPS

- On MPS, expectation values can be computed efficiently
- Local observables O:

$\langle\psi|O|\psi\rangle \quad =$



- Computation time $\propto D^4$ (D^6) for OBC (PBC) [can be improved to D^3 (D^5)].
- Energies $\langle\psi|H|\psi\rangle$, corr. functions $\langle\psi|O_i\otimes\tilde{O}_j|\psi\rangle$ can be computed efficiently.

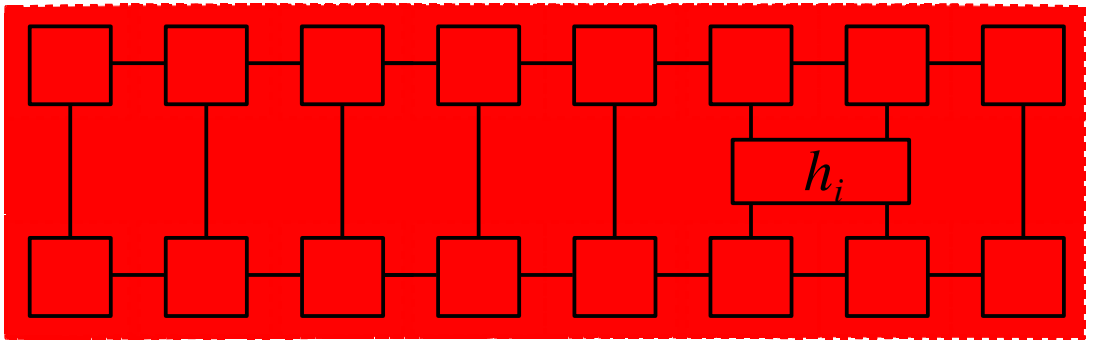
Variational method with MPS: DMRG

- $|\psi\rangle$ is linear in any $A^i \equiv X$: $|\psi_X\rangle = \sum \mathrm{tr}\left[A^1_{n_1} \cdots X_{n_k} \cdots A^L_{n_L}\right] |n_1, \cdots, n_L\rangle$

  $\Rightarrow$ The energy in state $|\psi_X\rangle$ is $E(\psi_X) = \dfrac{\langle\psi_X|H|\psi_X\rangle}{\langle\psi_X|\psi_X\rangle} = \dfrac{\vec{X}^\dagger M \vec{X}}{\vec{X}^\dagger N \vec{X}}$

  $\Rightarrow$ X which minimizes $E(\psi_X)$ can be found efficiently.

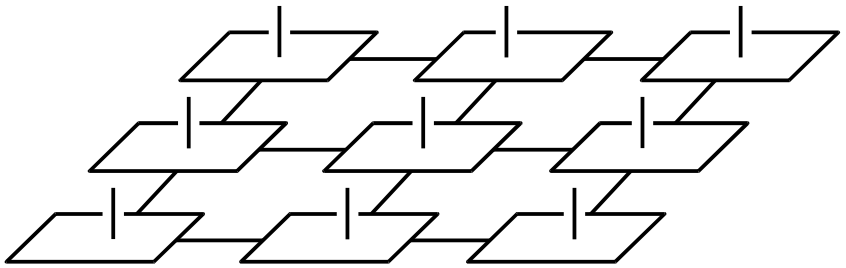
- Density Matrix Renormalization Group (DMRG) method: [White, PRL '92]
  Sweep through the MPS and minimize tensors locally.



- DMRG algorithm performs extremely well!
  Accuracy is tuned by increasing D (computation time $\propto D^3$).

- Idea extends to periodic boundary conditions (scales as D^5).

Extending DMRG to two dimensions



- The MPS ansatz naturally generalizes to two dimensions.

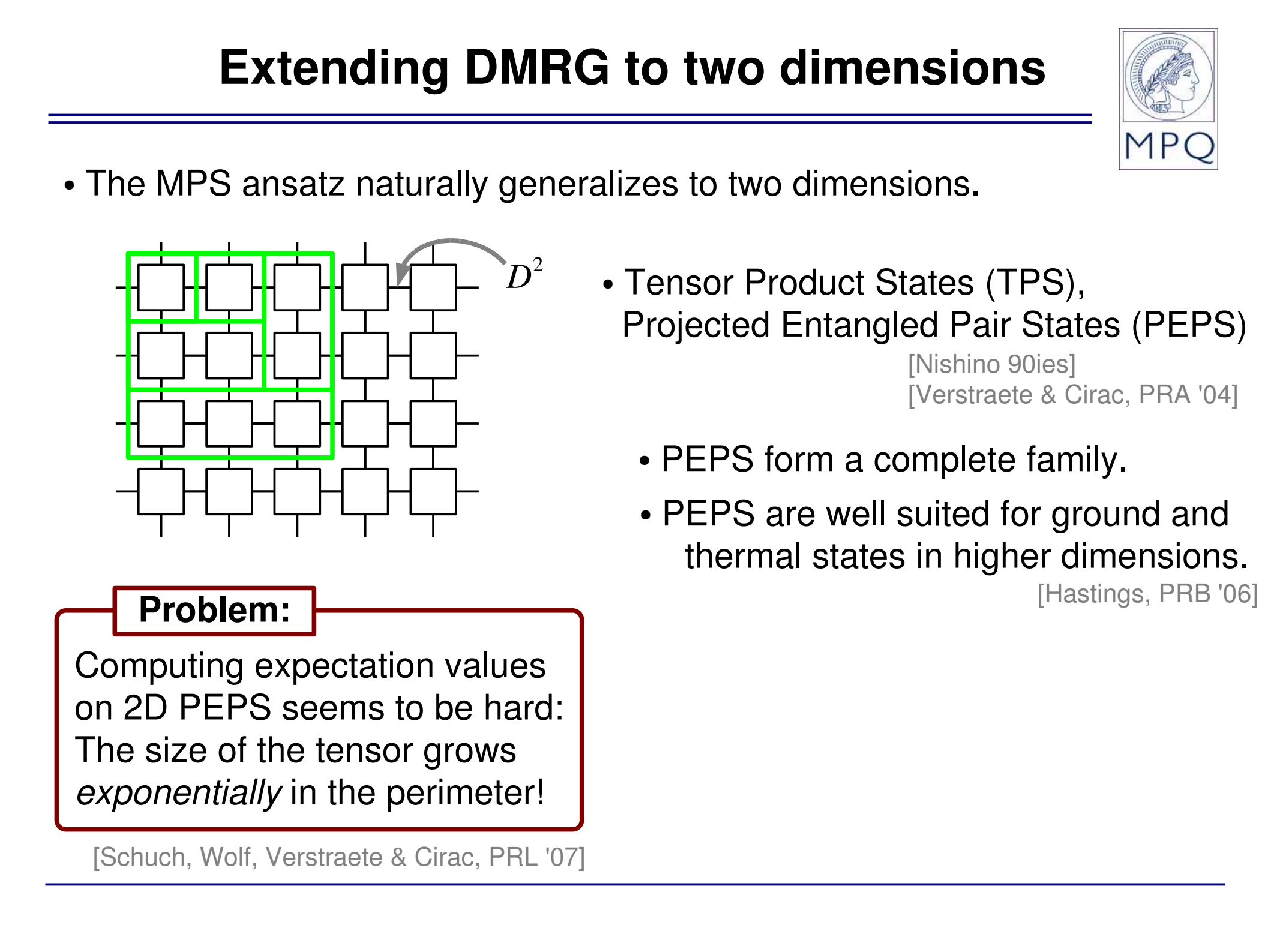
- Tensor Product States (TPS),
  Projected Entangled Pair States (PEPS)

  [Nishino 90ies]
  [Verstraete & Cirac, PRA '04]

  - PEPS form a complete family.
  - PEPS are well suited for ground and thermal states in higher dimensions.

    [Hastings, PRB '06]

Extending DMRG to two dimensions



- The MPS ansatz naturally generalizes to two dimensions.

- Tensor Product States (TPS), Projected Entangled Pair States (PEPS)

  [Nishino 90ies]
  [Verstraete & Cirac, PRA '04]

  - PEPS form a complete family.
  - PEPS are well suited for ground and thermal states in higher dimensions.

  [Hastings, PRB '06]

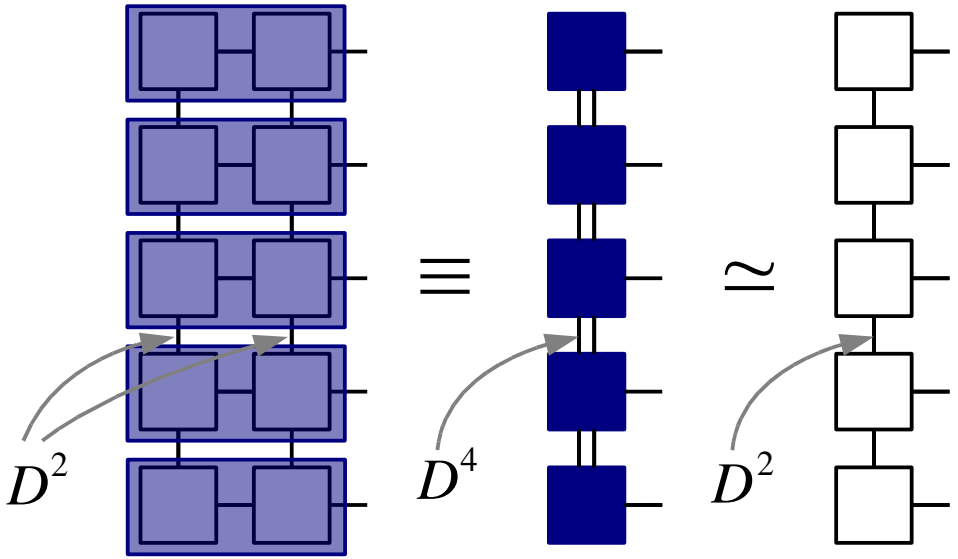
Problem:

Computing expectation values on 2D PEPS seems to be hard: The size of the tensor grows *exponentially* in the perimeter!

[Schuch, Wolf, Verstraete & Cirac, PRL '07]

PEPS algorithm with truncation

- Solution: proceed column-wise and truncate the bond dimension



- Corresponds to a DMRG-like optimization
- Works well in practice (due to symmetries of the system?)
- Known error

[Verstraete & Cirac, cond-mat/0407066]
[Murg, Verstraete & Cirac, PRA '07]
[Murg, Verstraete & Cirac '09]

- Works very well & outperforms other methods e.g. for frustrated systems
- Computation time scales as D^{12} (D^{18} for PBC) – can be reduced to D^{10} (D^{16})
- Up to D=5 (lattice up to 20×20).
- Not practical for PBC or higher dim. lattices, or irregular geometries.

The quest for other methods

- DMRG works well in 1D.
- PEPS extend DMRG to 2D and give very good results.
- PEPS contain the relevant states (ground and thermal states)
- However: - scaling of PEPS in D is unfavorable, in particular for PBC
  - truncation relies on lattice structure –
    not applicable to irregular systems

Is it possible to find a smaller/different class of states
- for which expectation values can be computed efficiently
- which has a favorable scaling in the accuracy parameter
- which does not rely on the geometry of the system
- which well approximates the states of interest

Variational Monte Carlo

- Given a state $|\psi\rangle$ on L spins, how can we sample exp. values?

$$\langle\psi|H|\psi\rangle = \sum_{\boldsymbol{n}} \langle\psi|\boldsymbol{n}\rangle\langle\boldsymbol{n}|H|\psi\rangle = \sum_{\boldsymbol{n}} \underbrace{|\langle\psi|\boldsymbol{n}\rangle|^2}_{\propto p(\boldsymbol{n})} \frac{\langle\boldsymbol{n}|H|\psi\rangle}{\langle\boldsymbol{n}|\psi\rangle}$$

$\boldsymbol{n}=(n_1,\ldots,n_L)$ basis state in some local basis
e.g. $0=\uparrow,\ 1=\downarrow$

$\Rightarrow \langle\boldsymbol{n}|\psi\rangle$ and $\langle\boldsymbol{n}|H|\psi\rangle$ need to be efficiently computable!

- $H=\sum h_i$ ← local terms

$$\Rightarrow \quad \langle\boldsymbol{n}|h_i = \sum_{\text{few } \boldsymbol{n}'} \phi(\boldsymbol{n}')\langle\boldsymbol{n}'| \qquad \Rightarrow \quad \langle\boldsymbol{n}|H|\psi\rangle = \sum_{\text{few more } \boldsymbol{n}'} \phi(\boldsymbol{n}')\langle\boldsymbol{n}'|\psi\rangle$$

$\langle n|\psi\rangle$ needs to be efficiently computable!

... the same works for products of Paulis, e.g. $\sigma^+\sigma^z\cdots\sigma^z\sigma^-$

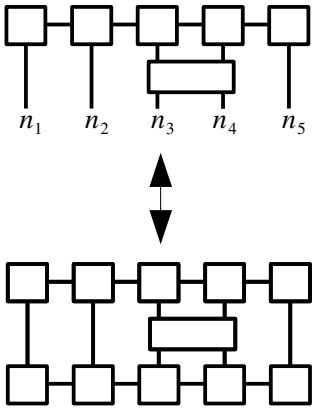
Classes for which this holds

- $\langle \boldsymbol{n} | \psi \rangle$ needs to be efficiently computable!

- Examples of such states:

  - Matrix Product States: $\langle \boldsymbol{n} | \psi \rangle = \mathrm{tr}\left[A^1_{n_1} \cdots A^L_{n_L} \right]$

    (note: computation of expectation values scales as D^2 $[D^3]$ instead of D^3 $[D^5]$ – could be used to speed up DMRG or PEPS methods; see also Sandvik & Vidal, PRL '07)



  - coherent version of classical thermal states:

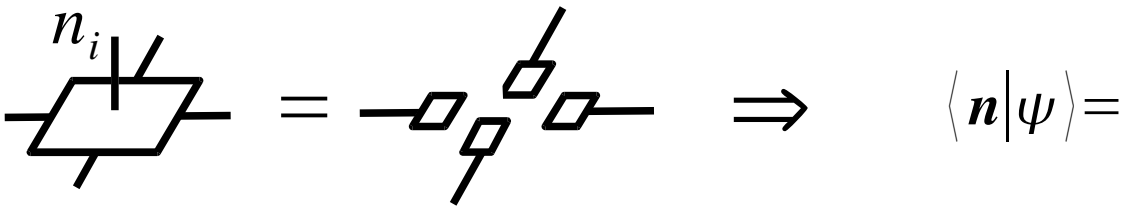
$$|\psi\rangle = \sum e^{-\beta H(\boldsymbol{n})/2} |\boldsymbol{n}\rangle \quad \text{for classical } H$$

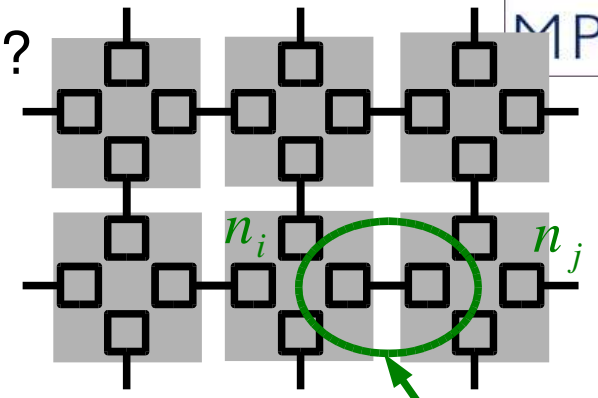
$$\Rightarrow \quad \langle \boldsymbol{n} | \psi \rangle = e^{-\beta H(\boldsymbol{n})/2}$$

Can we find new classes of suitable states (by extending these classes)?

Generalization of “classical” states

- What is the PEPS structure of “classical” states?

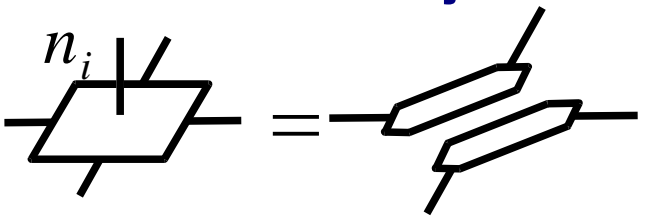


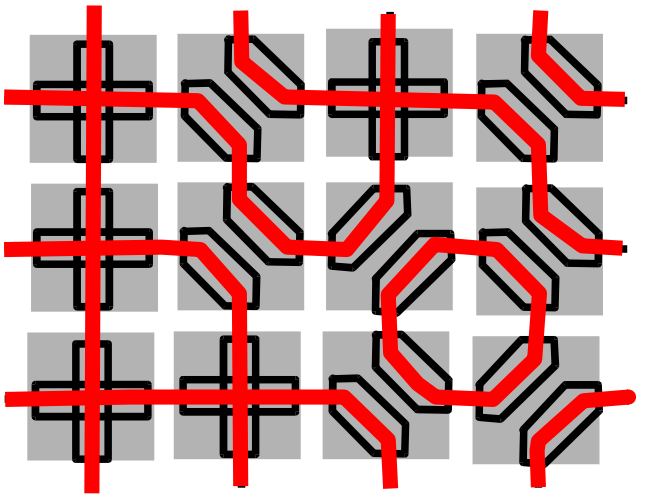


$\Rightarrow \langle n|\psi\rangle$ is product of **0-dimensional** objects!

$\Rightarrow$ It can be computed efficiently!

- Generalize to **1D objects**!





$\Rightarrow \langle n|\psi\rangle$ product of **1D** objects –
can still be efficiently computed!

- Generalization of “classical” states (or of Matrix Product States!)
- subclass of PEPS with eff. Monte Carlo sampling

String-bond states (SBS)

- **"String-bond states" (SBS)**

$$\langle \boldsymbol{n}|\psi\rangle = \prod_i f_i(\boldsymbol{n})$$

- defined on string (subset) of spins $(n_{i_1}, \ldots, n_{i_k})$
- efficiently computabe (e.g. matrix product trace)

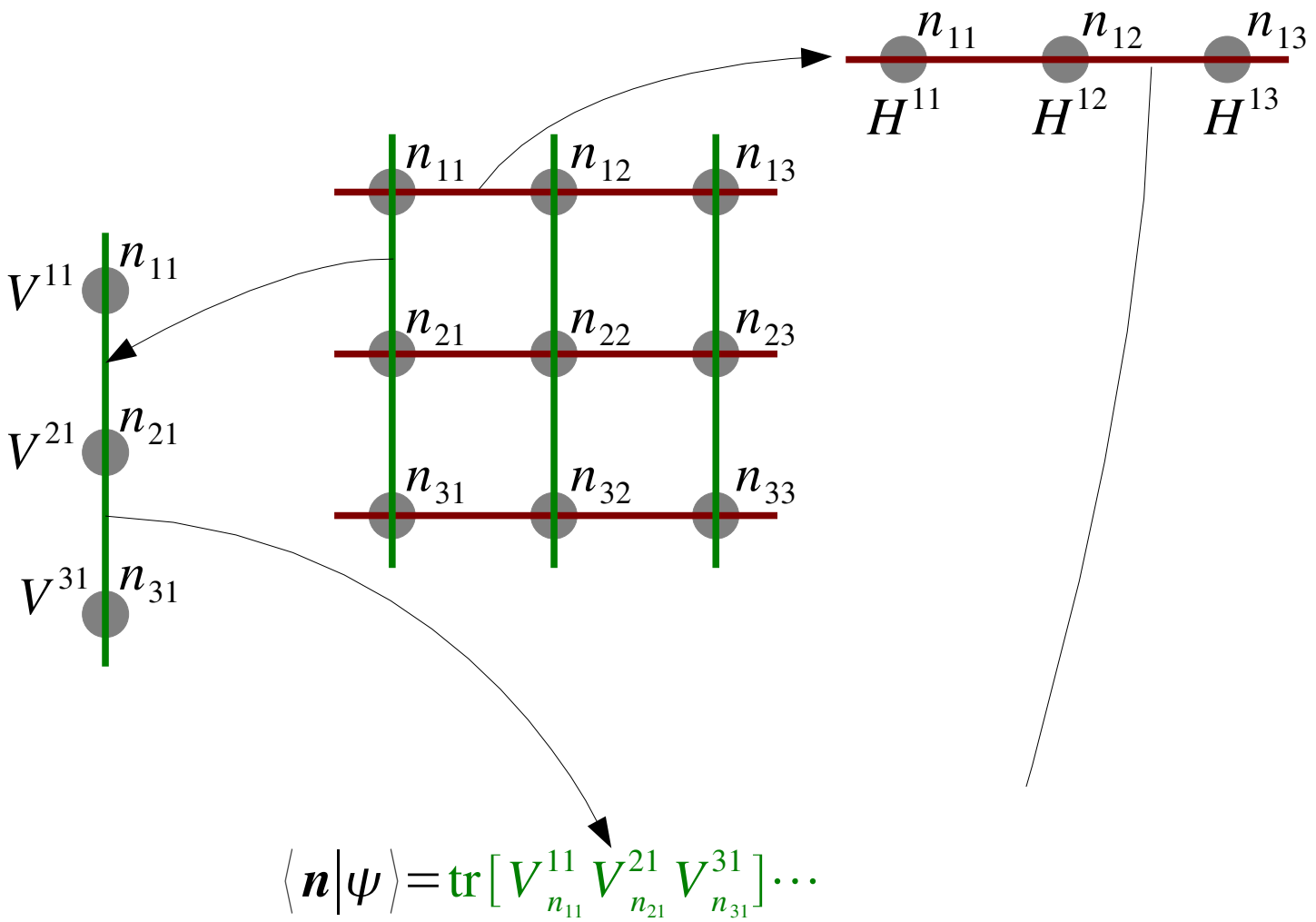
$\langle n|\psi\rangle$ is a product of efficiently computable functions defined on subsets of spins

- matrix product traces
- tree tensor networks
- on *small* subsets:
  - arbitrary state
  - mini-PEPS

- lines (horiz./vert./diagonal/...)
- 2x2 loops
- plaquettes (blocks) around each site

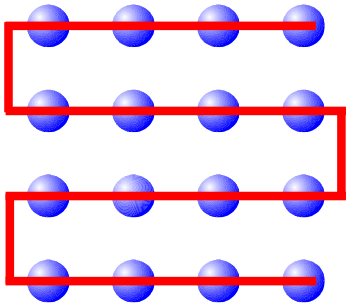
... and **any combination thereof!**

An example: Lines

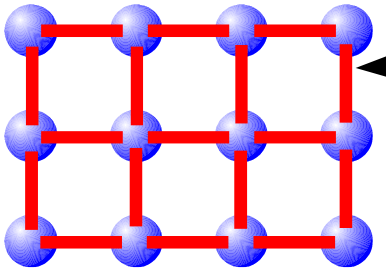


String-Bond States – Examples

- String-bond states form a **complete family** as they encompass MPS

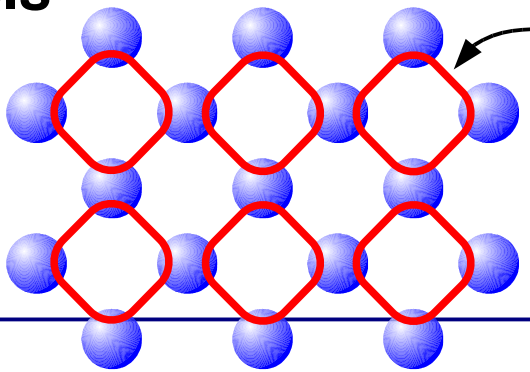


- Any (generalized, weighted) **graph state** is a SBS:
  - the **cluster state**
  - any **stabilizer state**



$$f(n_i, n_j) = (-1)^{n_i n_j}$$

- The **toric code state** and the **quantum double models** are SBS:



$$\begin{cases} 1 & \text{for even parity} \\ 0 & \text{for odd parity} \end{cases}$$

String-bond states: Properties

- SBS form a hierarchy of states which can be enlarged by
  - increasing D for the matrix products
  - adding new strings

- SBS include various common classes of states

- Computational resources scale favorably: D^2 (OBC) or D^3 (PBC)

  [cf. PEPS: D^{12} (2D OBC), D^{18} (2D PBC)]

- SBS can thus be used for PBC or 3D systems,
  and much higher D's can be used

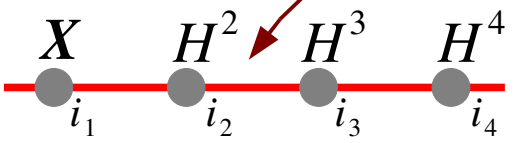
- no underlying geometry/locality necessary:
  also suitable for simulation of molecules,
  systems with non-local interactions, etc.

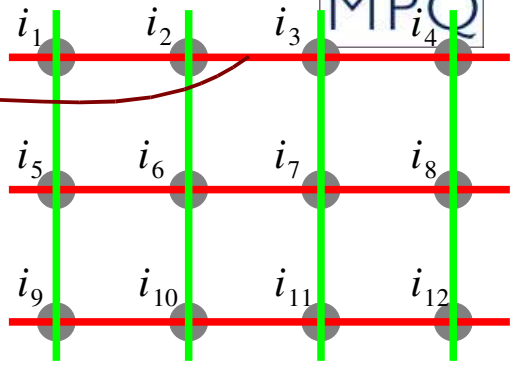
- each SBS is a PEPS, but PEPS-D exponential in #strings/site

Variational algorithm

- Basically as DMRG: pick *one* string, *one* tensor X on the string, and minimize

$$E(\psi_X) = \frac{\langle \psi_X | H | \psi_X \rangle}{\langle \psi_X | \psi_X \rangle} = \frac{\vec{X}^\dagger M \vec{X}}{\vec{X}^\dagger N \vec{X}}$$





- **Idea:** Determine M and N by MC sampling:

$$E(\psi_X) = \frac{\sum \langle \psi_X | \boldsymbol{n} \rangle \langle \boldsymbol{n} | H | \psi_X \rangle}{\sum \langle \psi_X | \boldsymbol{n}' \rangle \langle \boldsymbol{n}' | \psi_X \rangle} = \frac{\sum p_0(\boldsymbol{n}) \vec{X}^\dagger \cdot \vec{b}_{\boldsymbol{n}} \vec{a}_{\boldsymbol{n}}^\dagger \cdot \vec{X}}{\sum p_0(\boldsymbol{n}') \vec{X}^\dagger \cdot \vec{b}_{\boldsymbol{n}'} \vec{b}_{\boldsymbol{n}'}^\dagger \cdot \vec{X}}$$

$$M = \sum p_0(n) \vec{b}_n \vec{a}_n^\dagger$$

$$N = \sum p_0(n) \vec{b}_n \vec{b}_n^\dagger$$

$$\vec{b}_n^\dagger \cdot \vec{X} := \frac{\langle \boldsymbol{n} | \psi_X \rangle}{\langle \boldsymbol{n} | \psi_{X_0} \rangle}$$

$$\vec{a}_n^\dagger \cdot \vec{X} := \frac{\langle \boldsymbol{n} | H | \psi_X \rangle}{\langle \boldsymbol{n} | \psi_{X_0} \rangle}$$

$$p_0(\boldsymbol{n}) := \left| \langle \boldsymbol{n} | \psi_{X_0} \rangle \right|^2$$

→ Reweighting: one distribution samples all

However ...

$$\left.\begin{aligned} M &= \sum p_0(\boldsymbol{n})\,\vec{b}_{\boldsymbol{n}}\,\vec{a}^{\dagger}_{\boldsymbol{n}} \\ N &= \sum p_0(\boldsymbol{n})\,\vec{b}_{\boldsymbol{n}}\,\vec{b}^{\dagger}_{\boldsymbol{n}} \end{aligned}\right\} \quad \text{find } \vec{X} \text{ which minimizes } E(\psi_X) = \frac{\vec{X}^{\dagger} M \vec{X}}{\vec{X}^{\dagger} N \vec{X}}$$

However, there are some **problems** :

- M and N are **very large**: $(dD^2)^2$ – sampling rel. slow
- M and N less accurate far from $\vec{X}_0$ due to reweighting
- sampling error $\propto \sqrt{\text{MC sample length}}$

$\rightarrow$ small errors in kernel of N can have **big effect**

$\Rightarrow$ method **unstable!**

Gradient flow

- **Solution**: Move along gradient (in small steps), from initial value $X = X_0$:

$$\nabla_X E(\psi_X)\big|_{X=X_0} = 2[M - E(\psi_{X_0})N]\vec{X}_0$$

$$E(\psi_X) = \frac{\vec{X}^\dagger M \vec{X}}{\vec{X}^\dagger N \vec{X}}$$

$$\vec{X}_0{}^\dagger N \vec{X}_0 = 1$$

$\rightarrow$ well-behaved in N and M

- Gradient can be **sampled directly**:

$$\nabla_X E(\psi_X)\big|_{X=X_0} = 2\sum p_0(\boldsymbol{n})\vec{b}_{\boldsymbol{n}}[E_n - E(\psi_{X_0})]$$

$$E_{\boldsymbol{n}} = \frac{\langle \boldsymbol{n}|H|\psi_{X_0}\rangle}{\langle \boldsymbol{n}|\psi_{X_0}\rangle}$$

$$\vec{b}_n^\dagger \cdot \vec{X} := \frac{\langle \boldsymbol{n}|\psi_X\rangle}{\langle \boldsymbol{n}|\psi_{X_0}\rangle}$$

- Gradients indep. to first order:
  All tensors can be **updated simultaneously**
  (also allows for more eff. sampling)

The full algorithm

- Fix a string pattern & bond dimensions D.
- Choose initial configuration for the strings (random, guess, ...)
- Repeat: 1) Monte Carlo sample gradients (& Energy)
  2) Update tensors according to gradient and stepwidth ϵ
  (to this end: normalize gradients)
  3) Stop if energy has converged, otherwise go to 1)
- Increase precision: either
  - decrease stepwidth ϵ
  - increase D
  - add new strings

  and restart algorithm from obtained optimum,
  until energy does not improve any more

Testing the algorithm

- How can we benchmark SBS?

Quantum Monte Carlo

+ PBC, 3D possible
- no frustrated systems

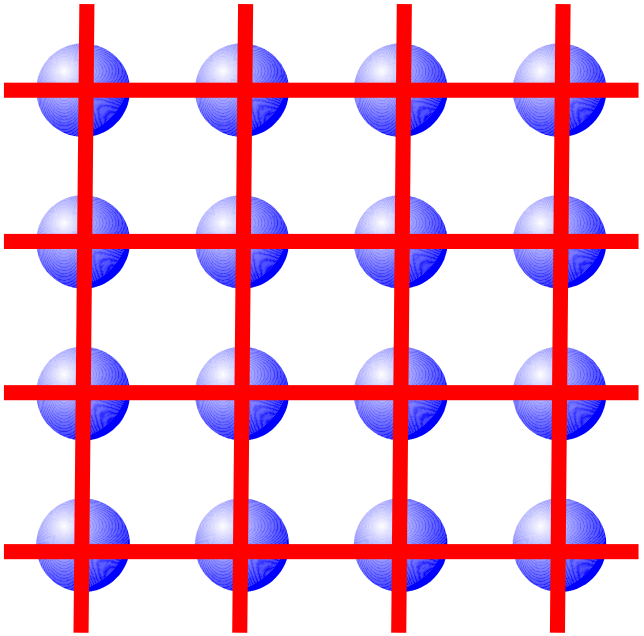
PEPS

+ can do frustr. systems
- only 2D OBC

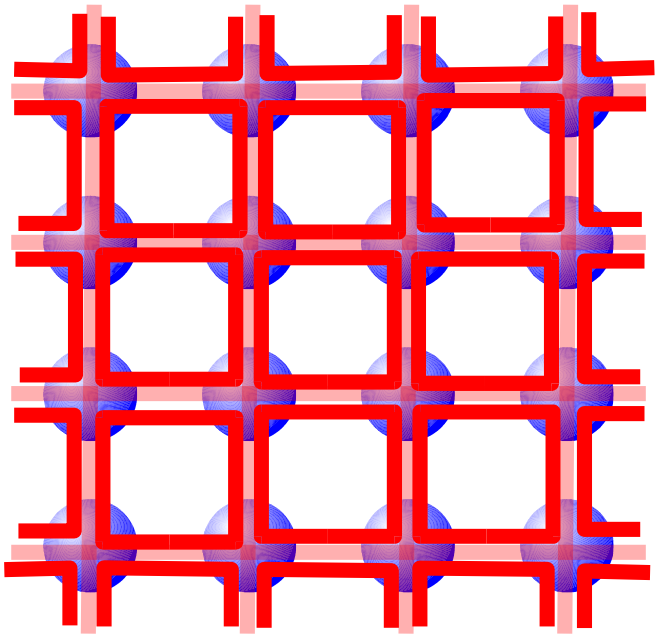
- **Note: Aim – simulating systems neither QMC nor PEPS can do!**

Implemented patterns

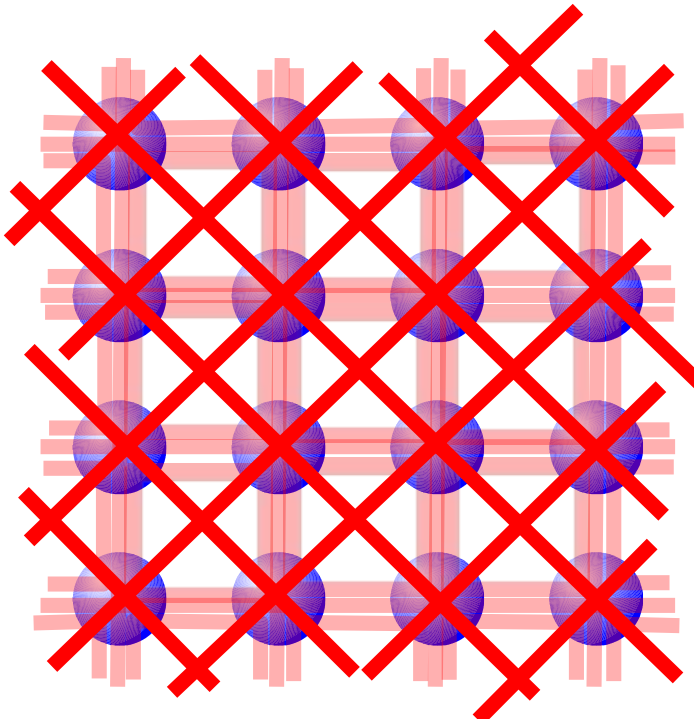
lines



loops

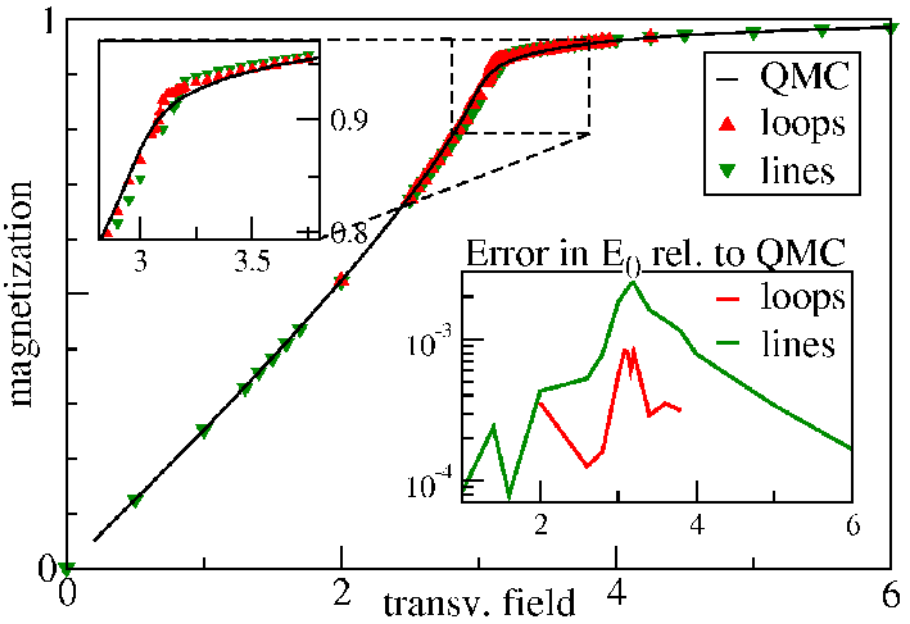


diagonals



Comparison to QMC: transverse Ising model

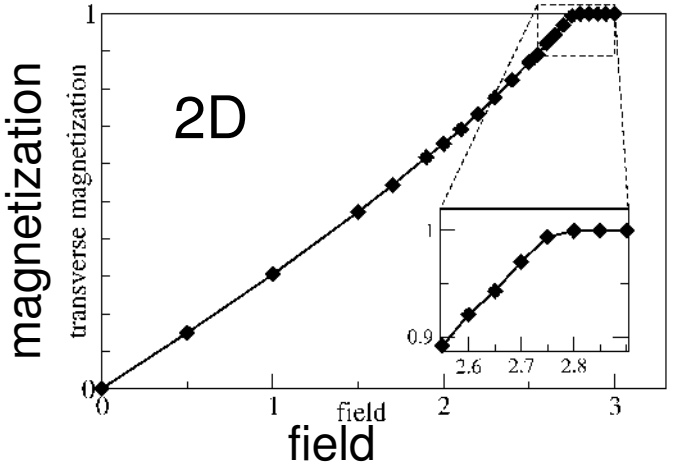
2D transverse Ising model (PBC):



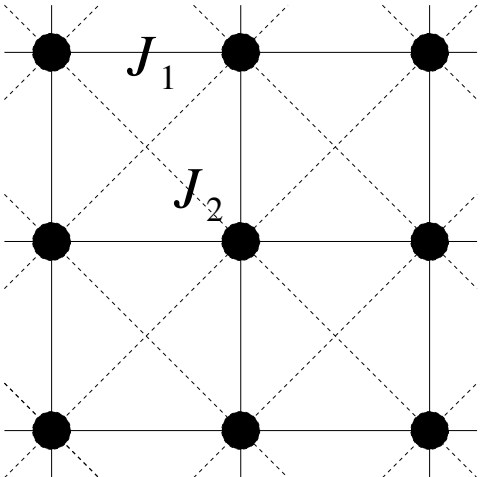
QMC vs. lines and lines+loops

Comparison to PEPS: frustrated systems

- first comparisons: frustrated XX model – SBS can outperform PEPS ($D = 12$ vs. $D = 4$).
- simulation of 2D frustrated PBC XX model w. transv. field:



- extensive comparison (and extension to PBC): $J_1 - J_2$ model



$$H = J_1 \sum_{<i,j>} \vec{\sigma}_i \cdot \vec{\sigma}_j + J_2 \sum_{\ll i,j \gg} \vec{\sigma}_i \cdot \vec{\sigma}_j$$

Projected String Bond States

- Can we incorporate symmetries of H in the ansatz?
- $\mathrm{SU}(2)$ invariant Hamiltonian:

  There exists ground state $|\psi_0\rangle$ with $\hat{M}|\psi_0\rangle = 0$ (where $\hat{M} = \sum_i \sigma_i^z$)

- For *any* candidate $|\psi_0\rangle$, $\boldsymbol{|\bar{\psi}_0\rangle = \Pi_0 |\psi_0\rangle}$ is at least as good!

- Implementation:

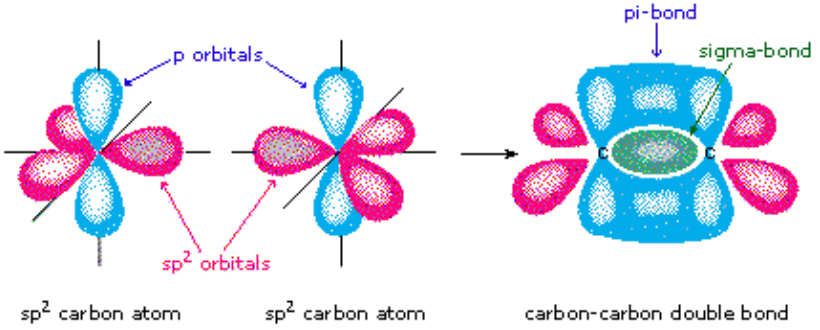
  (this is a SBS – $\langle \boldsymbol{n}|\psi\rangle$ can be computed efficiently)

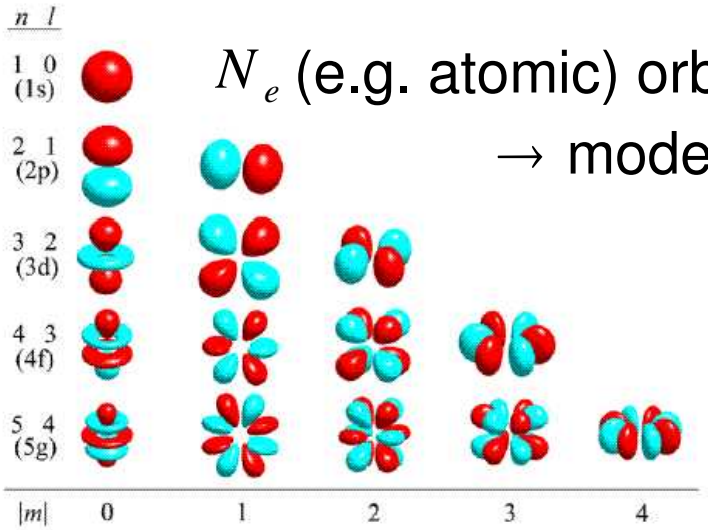
  $\rightarrow$ enforce by sampling only from the $M = 0$ subspace.

- In practice: start from configuration with $M = 0$ and swap a random pair of spins

Molecules

- Simulation of molecules:





N_e (e.g. atomic) orbitals

→ modes c_{is}

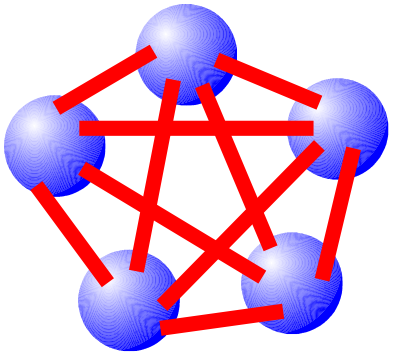
$$H = \sum_{i\,j\,s} T_{ij}\, c^{\dagger}_{is} c_{js} + \frac{1}{2} \sum_{i\,j\,k\,l\,s\,s'} V_{ijkl}\, c^{\dagger}_{is} c^{\dagger}_{is'} c_{ks'} c_{ls}$$

- Problem: no natural ordering (geometry) of modes

Molecules

- Ansatz: put strings between **all** pairs of modes

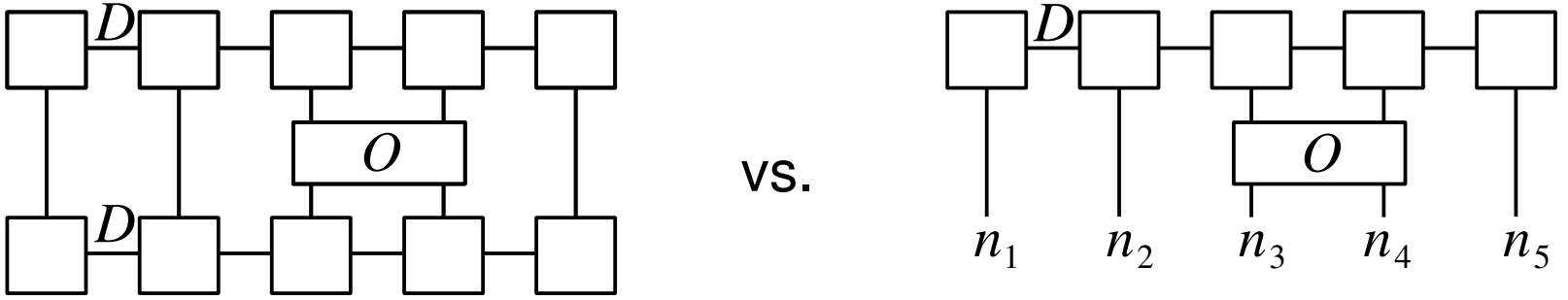
$$|\psi\rangle = \prod_{\alpha>\beta} f_{\alpha,\beta}(n_\alpha, n_\beta) \underbrace{(c_L^\dagger)^{n_L} \cdots (c_1^\dagger)^{n_1} |\mathrm{vac}\rangle}_{\equiv |n_1, \ldots, n_L\rangle}$$



- **And:** sample only in subspace with N_e fermions
- Advantages:
  - all modes treated on equal footing
  - ansatz invariant under relabeling of modes
- gradient flow optimization of the $f_{\alpha\beta}$
- possible to include canonical transformations $\tilde{c}_i = \sum O_{ij} c_j$ in the gradient optimization
- extension: use $f_{\alpha\beta\gamma\delta}(n_\alpha, n_\beta, n_\gamma, n_\delta)$ on quadruples of modes
- results soon!

Speeding up DMRG/PEPS with Monte Carlo

- The computational cost is D^2 / D^3 times the number of strings
  (each string can be evaluated by sequence of matrix multiplications)
- This is better than the scaling of DMRG (=a special SBS) – D^3 / D^5 .
- Reason: We don't have to contract the "sandwich" $\langle\psi|O|\psi\rangle$
  but only $\langle \boldsymbol{n}|\psi\rangle$ and $\langle \boldsymbol{n}|O|\psi\rangle$.



- Monte Carlo sampling can thus be used to speed up DMRG
  and, more importantly, the PEPS algorithm (D^6 instead of D^{12}).

[see also: Sandvik and Vidal, PRL '07]

Summary

- Monte Carlo sampling can help in tensor contraction
- String-bond states: $\langle \boldsymbol{n} | \psi \rangle$ prod. of eff. computable “local” objects
- String pattern can reflect geometry/ent. structure of system
- Enlarge family by larger D or more strings
- Cost of contraction: $D^3 \times$#strings
- SBS allow for efficient simulation of frustrated systems
  also with 2D PBC and in 3D
- Monte Carlo can speed up any tensor network based method
- <u>Current & future direction:</u>
  - better understand & characterize class of SBS
  - simulate molecules
  - string patterns using plaquettes
  - (imag.) time evolution
  - smarter sampling

[N.Schuch, M. Wolf, F. Verstraete, I. Cirac, PRL '08]
[A. Sfondrini *et al.*, in preparation]