1. PEPS - V: gauge fixing

Recall gauge fixing for 1D PEPS:

\[
\begin{array}{ccccccc}
\end{array}
\]

To left-normalize:

\[
\frac{1}{m} \begin{bmatrix} 1 & 1 \\ 1 & m \end{bmatrix} \frac{1}{A R M} = \begin{bmatrix} 1 & 1 \\ 0 & m \end{bmatrix} = \begin{bmatrix} A \hat{A} \\ \hat{A} \end{bmatrix}
\]

To right-normalize:

\[
\frac{1}{m} \begin{bmatrix} 1 & 1 \\ 1 & m \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & m \end{bmatrix} = \begin{bmatrix} m L & \alpha \\ \hat{A} & \hat{B} \end{bmatrix}
\]

The step of "gauge fixing", i.e. factoring out unitary matrix, \( U \), on the bond, introduces an orthogonal state space on the left or right:

\[
\frac{A^*}{A} = C
\]

norm matrix

\[
\frac{B^*}{B} = C
\]

norm matrix

norm matrix of environment = 1.

---

Gauge fixing for PEPS

"left-environment"

"right-environment"

\[\text{Lubasch 2014a}\]

\[\text{Brunsgaard 2017, Sec. 2.5}\]

We would like to have:

\[ E_{\text{full}} C = C \quad \text{and} \quad C E_{\text{full}} = C \]

So, try to do some type of QR decomposition that achieves this:

First diagonalize:

\[
\frac{E_{\text{full}}}{2} \approx \begin{bmatrix} W & \epsilon & 1 \\ & & Z \end{bmatrix}
\]

Retain only positive eigenvalues:

\[ Z = \sqrt{\epsilon} W \]
Now factorize out a unitary matrix from $E$, in two ways:

$$E_{\text{full}} = R \quad \text{orthogonal state space}$$

$$L^{-1} \quad \text{orthogonal state space}$$

Then

$$\mathcal{C}_{w} = \mathcal{C}_{y'} = 1_{3'} \quad \text{for } w'$$

$$\mathcal{D} = 1_{1''}$$

Now, absorb $-R$- and $-L^{-1}$- and the tensors that contact with

Finally, also adjust $X$ and $Y$: Recall:

$$M'_{x} = X_{x}^{i} = X_{x}^{i} \quad \text{and } Y_{y}^{i + 1} = Y_{y}^{i + 1}$$

Hence, obtain:

All this was just "orthonormalization of environment". Now come optimization of tensors:

Apply two-side gate to $\tilde{v}, \tilde{w}$, instead of $v, w$.

Then compress $\tilde{v}, \tilde{w}$ to $v', w'$, using $E$, and finally update $M$-tensors:
Gauge fixing improves numerical stability.

Gauge fixing reduces condition number of norm matrix by orders of magnitude!

\[
\text{condition number} = \frac{\text{largest eigenvalue}}{\text{smallest eigenvalue}}
\]

This measures sensitivity of solution of \( Ax = b \).

**TABLE I.** We show the mean condition number of the norm matrix with its standard deviation in the reduced tensor update without our gauge fixing, using only the positive approximant, and with our gauge fixing during the FU imaginary time evolution of \( D = 2 \) (a) and \( D = 4 \) (b) PEPS of size \( N = 11 \times 11 \) for the Ising model and of size \( N = 10 \times 10 \) for the Heisenberg model. The values were obtained averaging over 10 time steps and all tensors in the lattice.

<table>
<thead>
<tr>
<th>Model</th>
<th>Positive approximant</th>
<th>Gauge fixing</th>
</tr>
</thead>
<tbody>
<tr>
<td>( B = 1.0 ) Ising</td>
<td>( (2 \pm 3) \times 10^2 )</td>
<td>( 1.1 \pm 0.1 )</td>
</tr>
<tr>
<td>( B = 3.0 ) Ising</td>
<td>( (2 \pm 3) \times 10^3 )</td>
<td>( 1.6 \pm 0.1 )</td>
</tr>
<tr>
<td>Heisenberg</td>
<td>( (8 \pm 5) \times 10^2 )</td>
<td>( 1.08 \pm 0.02 )</td>
</tr>
</tbody>
</table>

**FIG. 13.** (Color online) Mean value of the relative change 
\( c_\beta (u) := |d(u) - d(u - 1)|/|d_{\text{init}}| \) of the cost function \( d \), Eq. (1), after consecutive update sweeps \( u \) over a tensor pair computed with respect to the initial value of the cost function \( d_{\text{init}} \), for the \( D = 4 \) reduced tensor update setting of Table I. We compare the FU evolution without our gauge fixing using only the positive approximant (open symbols) to the same propagation with our gauge fixing (filled symbols), for a \( 11 \times 11 \) Ising model at \( B = 1.0 \) (circles) and 3.0 (squares), and for a \( 10 \times 10 \) Heisenberg model (triangles).

10. **Fermionic PEPS**

Can be accomplished using two "fermionization rules":

- First introduced in context of fermionic MERA,

  [Corboz2009], with Vidal,

  [Corboz2010a], with Evenbly, Verstraete, Vidal,

  Then adapted to PEPS in [Corboz2010b], with Orús, Bauer, Vidal.

This is the approach described in [Bruneau2017].

Key ingredients:

(i) use only positive parity tensors

(ii) replace line crossings by fermionic swap gates

Equivalent formulations have also been developed by:

- [Barthel2009] with Pineda, Eisert,

  [Pineda2010], with Barthel, Eisert,

  [Kraus2010] with Schuch, Verstraete, Cirac,

  [Shi2010] with Li, Zhao, Zhou.
(i) Parity conservation

fermionic Hamiltonians preserve parity of electron number: \( \hat{P} = (-1)^{\hat{c}^\dagger\hat{c}} \)

\[ \hat{H} = c^\dagger c + c c^\dagger + c^\dagger + c \]

\[ [\hat{H}, \hat{P}] = 0 \Rightarrow \text{all energy eigenstates are parity eigenstates, too:} \]

\[ \hat{H} |\alpha, p\rangle = E_{\alpha, p} |\alpha, p\rangle, \quad \hat{P} |\alpha, p\rangle = p |\alpha, p\rangle, \quad p = \pm \text{ ("Z}_2\text{-symmetry")} \]

So, we may agree to work only with states of well-defined parity.

Eq. local state space for spinful fermions: \( |\alpha_1, \alpha_2, \alpha_3, \alpha_4\rangle \)

\( |10\rangle \equiv |10, 0, +\rangle \); \( c_1^\dagger |10\rangle \equiv |11, 0, -\rangle \), \( c_4^\dagger |10\rangle \equiv |10, 1, -\rangle \), \( c_4^\dagger c_5^\dagger |10\rangle \equiv |11, 1, +\rangle \)

\[ \Rightarrow \text{Every line in tensor network diagram also carries parity index } \alpha, p \]

---

Enforcing \( \text{Z}_2 \) symmetry [Corboz 2010 b, Sec. II.F.]

To enforce \( \text{Z}_2 \) symmetry in tensor network: choose all tensors to be "parity preserving"

**Rule (i):** \( \Gamma_0, \alpha_2, ... \alpha_n = 0 \text{ if } P_{\alpha_1, \alpha_2, ... \alpha_n} \equiv \text{p}(\alpha_1)\text{p}(\alpha_2) ... \text{p}(\alpha_n) \neq 1 \)

**Examples:**

\[
\begin{align*}
|10, 1, +\rangle & = 10, 1, + \quad P_{\alpha_1, \alpha_2, \alpha_3} = (+1)(-1)(-1) = 1 \quad \checkmark \\
|10, 0, +\rangle & = 10, 0, + \quad P_{\alpha_1, \alpha_2, \alpha_3} = (+1)(+1)(+1) = 1 \quad \checkmark \\
|11, 1, +\rangle & = 11, 1, + \quad P_{\alpha_1, \alpha_2, \alpha_3} = (+1)(-1)(-1) = 1 \quad \checkmark \\
|10, 1, -\rangle & = 10, 1, - \quad P_{\alpha_1, \alpha_2, \alpha_3} = (+1)(-1)(+1) = 1 \quad \checkmark \\
|10, 0, -\rangle & = 10, 0, - \quad P_{\alpha_1, \alpha_2, \alpha_3} = (+1)(+1)(-1) = 1 \quad \checkmark \\
\end{align*}
\]

\( c^\dagger \) and \( c \) both change parity by \((-1)\)

so overall \( \text{h}_2 \equiv (-1)^2 \).
(ii) Fermionic signs

\[ c_i c_j = - c_j c_i, \quad c_i c_j^+ = - c_j^+ c_i, \quad c_i^+ c_j = \delta_{ij} - c_j c_i^+ \]

To keep track of these signs, one chooses an ordering convention, say \(1, 2, 3, ..., N\) and defines:

\[ |1, 2, ..., N\rangle = \overbrace{c_1 c_2^+ ... c_N^+}^{10} |10\rangle \]

We have to keep this order in mind when evaluating matrix elements:

Example: consider \(N = 3\), \(14\rangle = 10, 1, 1\rangle = c_2 c_1^+ |10\rangle \]
\(12\rangle = 11, 1, 0\rangle = c_1 c_2^+ |10\rangle \]

\[ \langle 4' | c_1^+ c_3 | 4 \rangle = \langle 0 | c_2 c_1 c_1^+ c_3 c_3^+ c_2^+ c_3 10 \rangle \]
\[ = - \langle 0 | c_2 c_1 c_3 c_3^+ c_2^+ 10 \rangle = - 1 \]

In MPS language: [Corboz 2009, App:A]

\[ \langle 4 | = \langle 1101 \rangle = \langle 0 | \mathbb{I}_3 c_2 c_1 \]

\[ \hat{O} = c_1^+ c_3 \]

\[ \mathbb{I} c_1^+ c_1 \mathbb{I} |10\rangle \]

\[ \text{Filled-filled crossing: } (-1) \]

\[ \mathbb{I} c_3^+ c_3 |10\rangle \]

\[ \text{Filled-empty crossing: } (+1) \]

Line crossings keep track of operator rearrangements needed to get the two operators that “talk to \(\hat{O}\)” to become neighbors.
SWAP gates

Here crossing keep track of operator orderings.

(1) needed only for exchanging $c^t$ with $c$, or $c$ with $c^t$, or $c$ with $c$:  

\[ \begin{array}{cccccc}
 & 1_i & 1_j & \rightarrow & 1_i & 1_j \\
- & 1_i & c_i & (1) & c_i & 1_j \\
- & 1_i & c_i & (1) & 1_j & c_i \\
+ & 1_i & c_i & (1) & c_i & 1_j
\end{array} \]

Question: how can we encode this compactly?

Answer: SWAP gate that depends on parity of is commuting lines!

\[ \alpha \rightarrow \beta \quad S_{\alpha \beta} = S_{\alpha \beta} \delta_{\alpha \alpha'} S(\alpha, \beta) \]

Rule (ii): \( S(\alpha, \beta) = \begin{cases} +1 & \text{if } \rho(\alpha) = \rho(\beta) = -1 \\ -1 & \text{otherwise} \end{cases} \)

Fermionic quasiparticles: parity conserving \([\text{Cornell 2010a}], \text{Sec. II.F.} \)

Some matrix elements of operators involving fermions need minus signs:

Example: spinless fermions, consider two sites \( i, j \) with local basis \( | \sigma_i \sigma_j \rangle = \left( c_i^\dagger \sigma_i \right) \left( c_j^\dagger \sigma_j \right) | 0 : 0 \rangle \), \( \sigma_i \in \{ 0, 1 \} \)

General operator: \( \delta = \sum \langle \sigma_i \sigma_j | \delta | \sigma_i \sigma_j \rangle | \sigma_i \sigma_j \rangle \), with matrix elements:

\[ O_{\sigma_i \sigma_j}^{\sigma_i' \sigma_j'} = \langle \sigma_i \sigma_j | \delta | \sigma_i' \sigma_j' \rangle = \langle 0 : 0 | (c_i^\dagger \sigma_i') (c_j^\dagger \sigma_j') \delta (c_i \sigma_i') (c_j \sigma_j') | 0 : 0 \rangle \]

Example:

\[ \begin{array}{cc}
\text{Hopping:} & \delta = c_i^\dagger c_j \\
\text{Pairing:} & \delta = c_i c_j
\end{array} \]

\[ O_{0 : 0}^{1 : 1} = \langle 0 : 0 | c_i^\dagger c_j c_i c_j^\dagger | 0 : 0 \rangle = +1 \]

\[ O_{0 : 0}^{-1 : -1} = \langle 0 : 0 | c_i^\dagger c_j c_i c_j^\dagger | 0 : 0 \rangle = -1 \]
Fermionic operators: parity-changing [Corboz 2010 b, Sec. III F (iii)]

A or \( \bar{c} \) change parity, but note (i) demands: use only parity-carrying terms!

Remedy: add additional leg with index \( S \) taking just a single value, \( S = (0, p) = (1, -1) \), with parity assignment \( p(S) = -1 \):

which compensates for parity change induced by \( c \).

\[
(c_i^\dagger)_{i', S} = \left( \begin{array}{c} c_i^\dagger \vspace{0.1cm} \end{array} \right)_{i', S}
\]

only non-zero matrix element:

\[
\Phi^\dagger (-) \left( \begin{array}{c} c_i^\dagger \vspace{0.1cm} \end{array} \right)_{i'} = \left( \begin{array}{c} c_i^\dagger \vspace{0.1cm} \end{array} \right)_{i'}
\]

and its parity is

\[
P_{i', S} = P(c_i^\dagger) P(c_i) P(S) = (+1)(-1)(-1) = +1
\]

The new operator is written as:

\[
c_{i'} c_i = \left( \begin{array}{c} c_i \vspace{0.1cm} \end{array} \right)_{i'} \overset{\text{"contact"}}{\longrightarrow} \left( \begin{array}{c} c_i \vspace{0.1cm} \end{array} \right)_{i'}
\]

Since \( S \) carries just a single value, swap can be simplified to local operator that acts on local physical sites.

---

Jump move [Corboz 2009], App. C, [Corboz 2010 b, p.9]

Lines can be "dragged over tensors", because tensor preserves parity.

Finally, true for \( p' = + \), since then all swap signs are +1: \( S(p, +) = +1 \) for all \( p \).

Consider \( p' = - \):

1-leg tensor: \( - \)

Swap signs:

\[
\begin{array}{c|c}
\text{before} & \text{after} \\
S(p, -) & S(p, -) \\
\end{array}
\]

2-leg tensor:

\[
\begin{array}{c|c}
(p_1, p_2) & S(p_1, p_2, -) \\
\end{array}
\]

3-leg tensor:

\[
\begin{array}{c|c|c}
(p_1, p_2, p_3) & S(p_1, p_2, p_3, -) & S(p_1, p_2, -) \\
\end{array}
\]

General argument: parity-preserving tensor has even number of minus lines.

\[
\begin{align*}
\text{(Sign) before} & = \prod_{\beta} S(p_\beta, -1) \\
\text{(Sign) after} & = \prod_{\beta, \alpha} S(p_\beta, -1) \\
\text{even } \beta & \rightarrow (\text{Sign) before} = (\text{Sign) after}
\end{align*}
\]
MPS Examples [Bouguelia 2017]

Nearest-neighbor expectation value needs no swap gates.

Time evolution of nearest-neighbor operator:

Due to jump moves, the two ways of drawing red lines are equivalent.

Fermionic case via a PEPS

Choose some ordering for spin indices and stick to it.

Choose some ordering for spin indices and stick to it.

\(|\psi\rangle\) has some effect as Jordan-Wigner string.

Due to jump moves, the two ways of drawing red lines are equivalent.
Absorbing SWAP gates

\[ \langle \psi | \hat{O} | \psi \rangle = E_1 \]

SWAP gates for environmental tensor

\[ E_{0,\text{full}} = E_1 \]

Swap gates act differently on M and \( \hat{M} \):

\[ M_1 \rightarrow M' \]

Restoration of updated tensor:

\[ X' \rightarrow M' \]