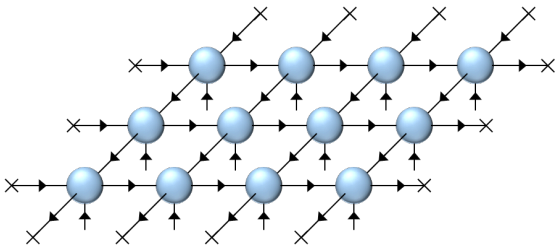
Contraction of finite PEPS

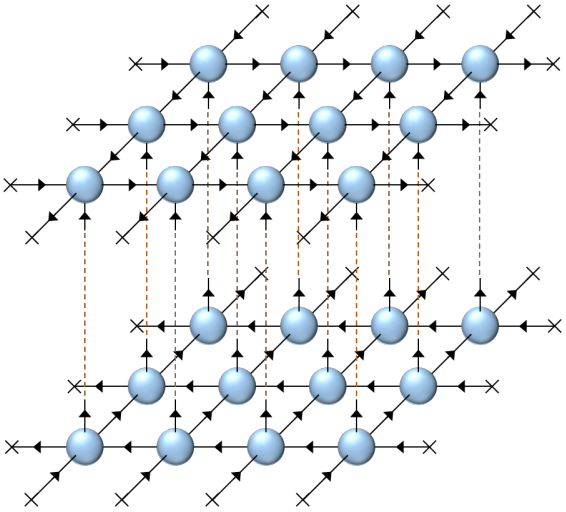
Author: Seung-Sup Lee

In this tutorial, we will compute the correlation functions of projected entangled-pair states (PEPSs) on a finite-sized square lattice with open boundary condition. For example, a ket PEPS on a 3×4 square lattice is represented by:

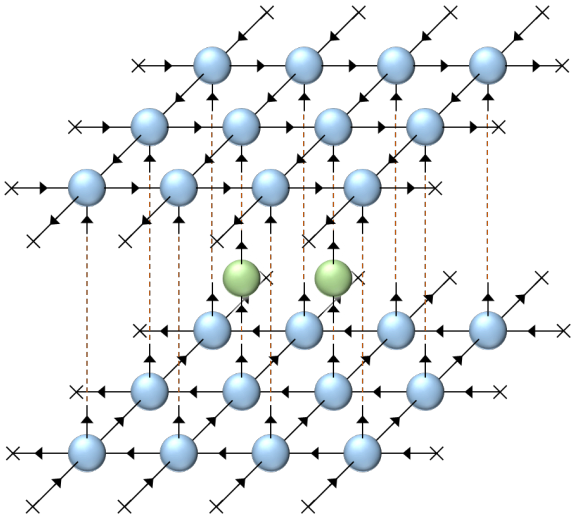


Here rank-5 tensors (blue balls) are arranged on a square grid of three rows and four columns. We index the tensors by its row and column indices: the upper-left corner is $(1,1)$ and the lower-right corner is $(3,4)$. We use the leg order convention of left-up-physical-down-right. The right and down legs are outward, and the rest are inward. The open legs at the boundary, which do not connect tensors, are dummy legs.

The squared norm of the PEPS is obtained by the contraction of tensor network that consists of the ket (upper layer) and bra (lower layer) PEPSs:

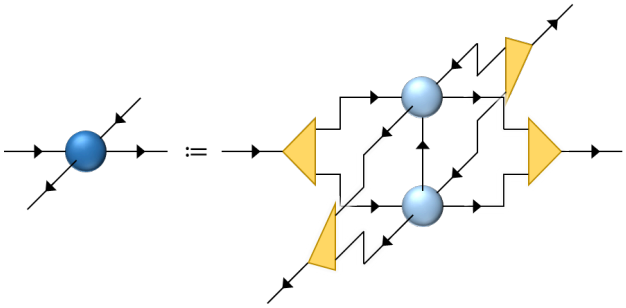


The spin-spin correlation function $\langle \hat{S}^z_{(i,j)} \hat{S}^z_{(i',j')} \rangle$ is obtained by the contraction of the ket and bra layers of PEPS with spin-z operators acting on sites (i,j) and (i',j'):

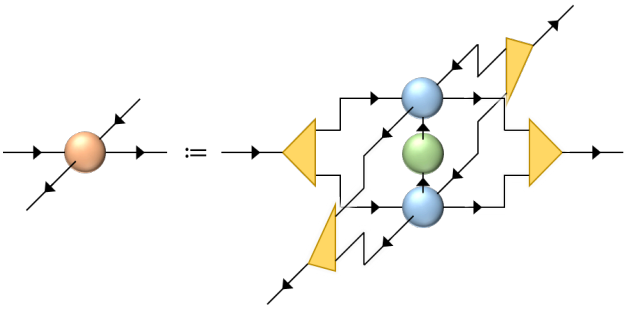


Here the green balls indicate the rank-2 form of spin-z operators.

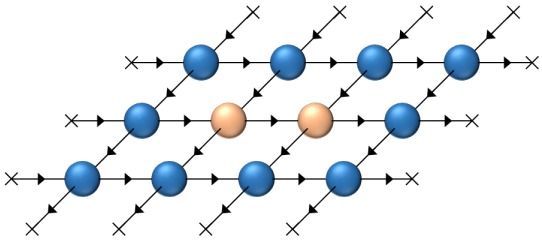
To contract such tensor networks, we use the following strategy. First, we contract all the tensors associated with the same lattice site, and fuse the in-plane legs by using isometries (yellow triangles).



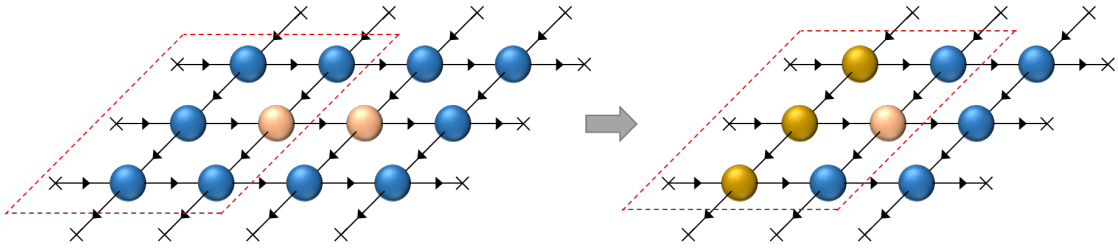
The result is called double tensor or transfer operator. Similarly, for the sites on which local spin operators act, we have



Then the tensor networks become purely two-dimensional. For example, the tensor network with spin-z operators, which is shown above, reduces to:



We contract this network column by column. The first (left-most) column is equivalent to a matrix product state (MPS) since the left open legs of the column are dummy legs. And the second and third columns are equivalent to matrix product operators (MPOs). We first contract the first and second columns, then contract the result with the third column, and so on.



After contracting with another column, we apply the downward and upward sweeps of truncations: the bonds lying along column directions are truncated via the singular value decomposition on the rank-4 tensor which is the contraction of two nearest-neighbor tensors.

Exercise (a): Complete `normFinPEPS_Ex.m`

There is a function `normFinPEPS_Ex.m` which is zipped together with this tutorial material. This function is designed to perform the contraction of the tensor networks made of the bra and ket PEPSs and, if given, local operators, following the strategy described above.

In this function, we consider the ket PEPS is uniform in the bulk, i.e., is maded of the same local tensor `A` as input. The tensors at the boundary are obtained by projecting the space of the open legs onto their first bases. That is, the bond space of local tensor `A` should be defined in a way that the first basis corresponds to the vacuum space.

Complete the parts which are enclosed by the comments `TODO - Exercise (a).` Once you complete the function, you can follow the demonstration below.

Resonating valence bond (RVB) state

As a paradigmatic PEPS, we consider the RVB state on a square lattice. In this tutorial, we use the PEPS representation of the RVB state introduced in F. Verstraete, M. M. Wolf, D. Perez-Garcia, and J. I. Cirac, Phys. Rev. Lett. **96**, 220601 (2006) (or its arXiv version).

Each lattice site has spin-1/2 degree of freedoms, having two spin states, $|\uparrow_i\rangle$ and $|\downarrow_i\rangle$. Each lattice site is decomposed into four auxiliary sites, each of which corresponds to the bond direction: left, up, down, and right. Each auxiliary site has the three-dimensional Hilbert space: empty $|e\rangle$, spin-up $|\Uparrow\rangle$, and spin-down $|\Downarrow\rangle$. The local physical states of the lattice site are given by the projection of the product space of four auxiliary sites associated with the same lattice site.

The valence bond state is the entangled state of two auxiliary sites that are associated with the bond connecting nearest-neighbor lattice sites. For example, the valence bond state connecting a site i and the other site j (that lies on the right of site i) is:

$$|E_{ir,jl}\rangle = |e_{ir}, e_{jl}\rangle + |\Uparrow_{ir}, \Downarrow_{jl}\rangle - |\Downarrow_{ir}, \Uparrow_{jl}\rangle,$$

where r and l mean that the virtual sites for the right and left bonds, respectively. The valence bond is represented by a rank-2 tensor:

```
clear
% left(in)-right(out) for horizontal valence bond,
% or up(in)-down(out) for vertical valence bond
VB = blkdiag(1,[0,1;-1,0]);
```

Here we define the first, second, and third basis states as $|e\rangle$, $|\Uparrow\rangle$, and $|\Downarrow\rangle$, respectively.

The projection onto local physical space is applied onto four auxiliary sites at the same lattice site:

$$\begin{aligned}\hat{P}_i = |\uparrow_i\rangle(\langle\Uparrow_{il}, e_{iu}, e_{id}, e_{ir}| + \langle e_{il}, \Uparrow_{iu}, e_{id}, e_{ir}| + \langle e_{il}, e_{iu}, \Uparrow_{id}, e_{ir}| + \langle e_{il}, e_{iu}, e_{id}, \Uparrow_{ir}|) \\ + |\downarrow_i\rangle(\langle\Downarrow_{il}, e_{iu}, e_{id}, e_{ir}| + \langle e_{il}, \Downarrow_{iu}, e_{id}, e_{ir}| + \langle e_{il}, e_{iu}, \Downarrow_{id}, e_{ir}| + \langle e_{il}, e_{iu}, e_{id}, \Downarrow_{ir}|).\end{aligned}$$

The numerical construction of $\hat{P}_i$ is done as:

```
P = zeros(3,3,2,3,3); % left(in)-up(in)-physical(in)-down(out)-right(out)

for it1 = (1:3) % left (in)
    for it2 = (1:3) % up (in)
        for it3 = (1:3) % down (out)
            for it4 = (1:3) % right (out)
                it_tot = [it1 it2 it3 it4];
                % spin-up in the physical space
                if (sum(it_tot == 1) == 3) && (sum(it_tot == 2) == 1)
                    P(it1,it2,1,it3,it4) = 1;
                end

                % spin-down in the physical space
                if (sum(it_tot == 1) == 3) && (sum(it_tot == 3) == 1)
                    P(it1,it2,2,it3,it4) = 1;
                end
            end
        end
    end
end
```

Then the local tensor is obtained by contracting (i) the down leg of the projector `P` with the up leg of vertical valence bond `VB` and (ii) the right leg of the contraction result with the left leg of horizontal valence bond `VB`. (**Quick exercise**: how about the left and up legs of `P`?)

```
A = contract(P,5,4,VB,2,1,[(1:3) 5 4]);
A = contract(A,5,5,VB,2,1);
```

Spin-spin correlation of the RVB state

Let's compute the spin-spin correlation $\langle\hat{S}^z_{(i,j)}\hat{S}^z_{(i,j+1)}\rangle$ for the nearest neighbors (i,j) and $(i,j+1)$. We choose the row and column indices i and j so that the sites lie at the center of the lattice. With this choice, we can minimize the boundary effect.

Compute the spin-spin correlation for different system sizes.

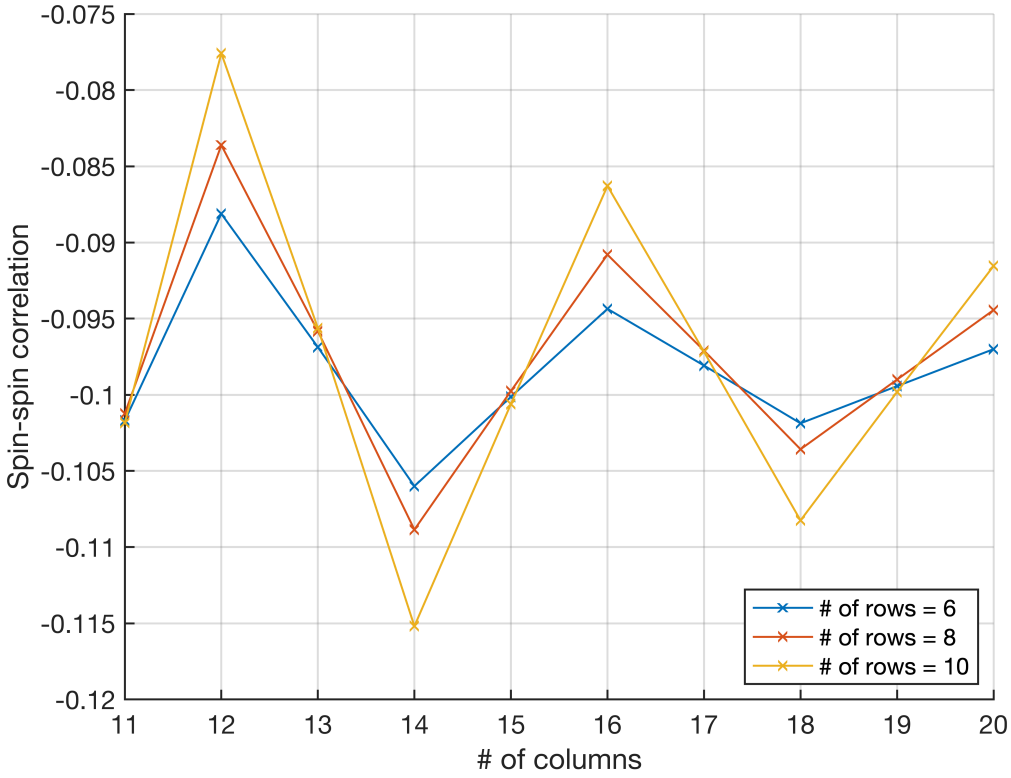
```
[S,I] = getLocalSpace('Spin',1/2);
Sz = squeeze(S(:,3,:));

Nkeep = 30; % maximum bond dimension
Nrow = (6:2:10); % number of rows
Ncol = (11:20); % number of columns
Cvals = zeros(numel(Nrow),numel(Ncol)); % correlation

for itr = (1:numel(Nrow))
    for itc = (1:numel(Ncol))
        % location of spin operators
        rO = [round(Nrow(itr)/2),round((Ncol(itc)+1)/2)-1];
        rO = [rO;rO+[0 1]];

        val1 = normFinPEPS_Ex (A,Nrow(itr),Ncol(itc),Nkeep);
        val2 = normFinPEPS_Ex (A,Nrow(itr),Ncol(itc),Nkeep,Sz,rO);

        Cvals(itr,itc) = val2/val1;
    end
end
```

```
Finite PEPS contraction: Nrow = 6, Ncol = 11, Nkeep = 30
Elapsed time: 0.6171s, CPU time: 5.23s, Avg # of cores: 8.475
Finite PEPS contraction: Nrow = 6, Ncol = 11, Nkeep = 30
  Act local operator(s) onto (x, y) = (3, 5), (3, 6)
Elapsed time: 0.5025s, CPU time: 4.09s, Avg # of cores: 8.139
Finite PEPS contraction: Nrow = 6, Ncol = 12, Nkeep = 30
Elapsed time: 0.5487s, CPU time: 4.61s, Avg # of cores: 8.401
Finite PEPS contraction: Nrow = 6, Ncol = 12, Nkeep = 30
  Act local operator(s) onto (x, y) = (3, 6), (3, 7)
Elapsed time: 0.5083s, CPU time: 4.27s, Avg # of cores: 8.401
Finite PEPS contraction: Nrow = 6, Ncol = 13, Nkeep = 30
Elapsed time: 0.6137s, CPU time: 5.1s, Avg # of cores: 8.31
Finite PEPS contraction: Nrow = 6, Ncol = 13, Nkeep = 30
  Act local operator(s) onto (x, y) = (3, 6), (3, 7)
Elapsed time: 0.6103s, CPU time: 4.96s, Avg # of cores: 8.127
Finite PEPS contraction: Nrow = 6, Ncol = 14, Nkeep = 30
Elapsed time: 0.7107s, CPU time: 5.72s, Avg # of cores: 8.048
Finite PEPS contraction: Nrow = 6, Ncol = 14, Nkeep = 30
  Act local operator(s) onto (x, y) = (3, 7), (3, 8)
Elapsed time: 0.6811s, CPU time: 5.52s, Avg # of cores: 8.104
Finite PEPS contraction: Nrow = 6, Ncol = 15, Nkeep = 30
Elapsed time: 0.8017s, CPU time: 6.39s, Avg # of cores: 7.97
Finite PEPS contraction: Nrow = 6, Ncol = 15, Nkeep = 30
  Act local operator(s) onto (x, y) = (3, 7), (3, 8)
Elapsed time: 0.7376s, CPU time: 6.36s, Avg # of cores: 8.622
Finite PEPS contraction: Nrow = 6, Ncol = 16, Nkeep = 30
Elapsed time: 0.76s, CPU time: 6.31s, Avg # of cores: 8.303
Finite PEPS contraction: Nrow = 6, Ncol = 16, Nkeep = 30
  Act local operator(s) onto (x, y) = (3, 8), (3, 9)
Elapsed time: 0.7736s, CPU time: 6.31s, Avg # of cores: 8.157
Finite PEPS contraction: Nrow = 6, Ncol = 17, Nkeep = 30
Elapsed time: 0.8386s, CPU time: 6.9s, Avg # of cores: 8.228
Finite PEPS contraction: Nrow = 6, Ncol = 17, Nkeep = 30
  Act local operator(s) onto (x, y) = (3, 8), (3, 9)
Elapsed time: 0.8649s, CPU time: 7.2s, Avg # of cores: 8.325
Finite PEPS contraction: Nrow = 6, Ncol = 18, Nkeep = 30
Elapsed time: 0.8418s, CPU time: 7.03s, Avg # of cores: 8.351
```

```
Finite PEPS contraction: Nrow = 6, Ncol = 18, Nkeep = 30
  Act local operator(s) onto (x, y) = (3, 9), (3, 10)
Elapsed time: 0.8785s, CPU time: 7.37s, Avg # of cores: 8.389
Finite PEPS contraction: Nrow = 6, Ncol = 19, Nkeep = 30
Elapsed time: 0.8971s, CPU time: 7.85s, Avg # of cores: 8.751
Finite PEPS contraction: Nrow = 6, Ncol = 19, Nkeep = 30
  Act local operator(s) onto (x, y) = (3, 9), (3, 10)
Elapsed time: 0.8996s, CPU time: 7.54s, Avg # of cores: 8.382
Finite PEPS contraction: Nrow = 6, Ncol = 20, Nkeep = 30
Elapsed time: 0.9742s, CPU time: 8.05s, Avg # of cores: 8.263
Finite PEPS contraction: Nrow = 6, Ncol = 20, Nkeep = 30
  Act local operator(s) onto (x, y) = (3, 10), (3, 11)
Elapsed time: 0.9581s, CPU time: 8.08s, Avg # of cores: 8.433
Finite PEPS contraction: Nrow = 8, Ncol = 11, Nkeep = 30
Elapsed time: 0.9967s, CPU time: 8.05s, Avg # of cores: 8.076
Finite PEPS contraction: Nrow = 8, Ncol = 11, Nkeep = 30
  Act local operator(s) onto (x, y) = (4, 5), (4, 6)
Elapsed time: 1.019s, CPU time: 8.15s, Avg # of cores: 7.998
Finite PEPS contraction: Nrow = 8, Ncol = 12, Nkeep = 30
Elapsed time: 1.141s, CPU time: 9.29s, Avg # of cores: 8.14
Finite PEPS contraction: Nrow = 8, Ncol = 12, Nkeep = 30
  Act local operator(s) onto (x, y) = (4, 6), (4, 7)
Elapsed time: 1.133s, CPU time: 9s, Avg # of cores: 7.945
Finite PEPS contraction: Nrow = 8, Ncol = 13, Nkeep = 30
Elapsed time: 1.232s, CPU time: 9.81s, Avg # of cores: 7.963
Finite PEPS contraction: Nrow = 8, Ncol = 13, Nkeep = 30
  Act local operator(s) onto (x, y) = (4, 6), (4, 7)
Elapsed time: 1.217s, CPU time: 9.74s, Avg # of cores: 8.007
Finite PEPS contraction: Nrow = 8, Ncol = 14, Nkeep = 30
Elapsed time: 1.312s, CPU time: 10.73s, Avg # of cores: 8.176
Finite PEPS contraction: Nrow = 8, Ncol = 14, Nkeep = 30
  Act local operator(s) onto (x, y) = (4, 7), (4, 8)
Elapsed time: 1.32s, CPU time: 10.68s, Avg # of cores: 8.093
Finite PEPS contraction: Nrow = 8, Ncol = 15, Nkeep = 30
Elapsed time: 1.476s, CPU time: 11.72s, Avg # of cores: 7.939
Finite PEPS contraction: Nrow = 8, Ncol = 15, Nkeep = 30
  Act local operator(s) onto (x, y) = (4, 7), (4, 8)
Elapsed time: 1.631s, CPU time: 13.28s, Avg # of cores: 8.143
Finite PEPS contraction: Nrow = 8, Ncol = 16, Nkeep = 30
Elapsed time: 1.962s, CPU time: 15.53s, Avg # of cores: 7.914
Finite PEPS contraction: Nrow = 8, Ncol = 16, Nkeep = 30
  Act local operator(s) onto (x, y) = (4, 8), (4, 9)
Elapsed time: 2.032s, CPU time: 15.96s, Avg # of cores: 7.855
Finite PEPS contraction: Nrow = 8, Ncol = 17, Nkeep = 30
Elapsed time: 2.181s, CPU time: 17.08s, Avg # of cores: 7.833
Finite PEPS contraction: Nrow = 8, Ncol = 17, Nkeep = 30
  Act local operator(s) onto (x, y) = (4, 8), (4, 9)
Elapsed time: 2.294s, CPU time: 17.49s, Avg # of cores: 7.624
Finite PEPS contraction: Nrow = 8, Ncol = 18, Nkeep = 30
Elapsed time: 2.501s, CPU time: 19.09s, Avg # of cores: 7.634
Finite PEPS contraction: Nrow = 8, Ncol = 18, Nkeep = 30
  Act local operator(s) onto (x, y) = (4, 9), (4, 10)
Elapsed time: 2.494s, CPU time: 19.13s, Avg # of cores: 7.67
Finite PEPS contraction: Nrow = 8, Ncol = 19, Nkeep = 30
Elapsed time: 2.93s, CPU time: 22.64s, Avg # of cores: 7.726
Finite PEPS contraction: Nrow = 8, Ncol = 19, Nkeep = 30
  Act local operator(s) onto (x, y) = (4, 9), (4, 10)
Elapsed time: 2.831s, CPU time: 21.74s, Avg # of cores: 7.679
Finite PEPS contraction: Nrow = 8, Ncol = 20, Nkeep = 30
Elapsed time: 2.902s, CPU time: 21.9s, Avg # of cores: 7.547
Finite PEPS contraction: Nrow = 8, Ncol = 20, Nkeep = 30
  Act local operator(s) onto (x, y) = (4, 10), (4, 11)
Elapsed time: 2.919s, CPU time: 22.1s, Avg # of cores: 7.571
Finite PEPS contraction: Nrow = 10, Ncol = 11, Nkeep = 30
Elapsed time: 2.315s, CPU time: 17.72s, Avg # of cores: 7.655
```

```
Finite PEPS contraction: Nrow = 10, Ncol = 11, Nkeep = 30
  Act local operator(s) onto (x, y) = (5, 5), (5, 6)
Elapsed time: 2.292s, CPU time: 17.31s, Avg # of cores: 7.551
Finite PEPS contraction: Nrow = 10, Ncol = 12, Nkeep = 30
Elapsed time: 2.508s, CPU time: 18.68s, Avg # of cores: 7.45
Finite PEPS contraction: Nrow = 10, Ncol = 12, Nkeep = 30
  Act local operator(s) onto (x, y) = (5, 6), (5, 7)
Elapsed time: 2.573s, CPU time: 19.21s, Avg # of cores: 7.465
Finite PEPS contraction: Nrow = 10, Ncol = 13, Nkeep = 30
Elapsed time: 2.769s, CPU time: 20.64s, Avg # of cores: 7.455
Finite PEPS contraction: Nrow = 10, Ncol = 13, Nkeep = 30
  Act local operator(s) onto (x, y) = (5, 6), (5, 7)
Elapsed time: 2.773s, CPU time: 20.77s, Avg # of cores: 7.49
Finite PEPS contraction: Nrow = 10, Ncol = 14, Nkeep = 30
Elapsed time: 3.063s, CPU time: 22.99s, Avg # of cores: 7.506
Finite PEPS contraction: Nrow = 10, Ncol = 14, Nkeep = 30
  Act local operator(s) onto (x, y) = (5, 7), (5, 8)
Elapsed time: 2.971s, CPU time: 22.11s, Avg # of cores: 7.441
Finite PEPS contraction: Nrow = 10, Ncol = 15, Nkeep = 30
Elapsed time: 3.411s, CPU time: 25.59s, Avg # of cores: 7.502
Finite PEPS contraction: Nrow = 10, Ncol = 15, Nkeep = 30
  Act local operator(s) onto (x, y) = (5, 7), (5, 8)
Elapsed time: 3.509s, CPU time: 26.41s, Avg # of cores: 7.526
Finite PEPS contraction: Nrow = 10, Ncol = 16, Nkeep = 30
Elapsed time: 3.727s, CPU time: 27.58s, Avg # of cores: 7.4
Finite PEPS contraction: Nrow = 10, Ncol = 16, Nkeep = 30
  Act local operator(s) onto (x, y) = (5, 8), (5, 9)
Elapsed time: 3.956s, CPU time: 29.82s, Avg # of cores: 7.538
Finite PEPS contraction: Nrow = 10, Ncol = 17, Nkeep = 30
Elapsed time: 3.866s, CPU time: 28.57s, Avg # of cores: 7.389
Finite PEPS contraction: Nrow = 10, Ncol = 17, Nkeep = 30
  Act local operator(s) onto (x, y) = (5, 8), (5, 9)
Elapsed time: 3.918s, CPU time: 28.67s, Avg # of cores: 7.318
Finite PEPS contraction: Nrow = 10, Ncol = 18, Nkeep = 30
Elapsed time: 4.049s, CPU time: 29.83s, Avg # of cores: 7.368
Finite PEPS contraction: Nrow = 10, Ncol = 18, Nkeep = 30
  Act local operator(s) onto (x, y) = (5, 9), (5, 10)
Elapsed time: 4.173s, CPU time: 30.57s, Avg # of cores: 7.325
Finite PEPS contraction: Nrow = 10, Ncol = 19, Nkeep = 30
Elapsed time: 4.352s, CPU time: 32.03s, Avg # of cores: 7.359
Finite PEPS contraction: Nrow = 10, Ncol = 19, Nkeep = 30
  Act local operator(s) onto (x, y) = (5, 9), (5, 10)
Elapsed time: 4.43s, CPU time: 32.44s, Avg # of cores: 7.322
Finite PEPS contraction: Nrow = 10, Ncol = 20, Nkeep = 30
Elapsed time: 4.608s, CPU time: 33.83s, Avg # of cores: 7.341
Finite PEPS contraction: Nrow = 10, Ncol = 20, Nkeep = 30
  Act local operator(s) onto (x, y) = (5, 10), (5, 11)
Elapsed time: 4.655s, CPU time: 34.04s, Avg # of cores: 7.313
```

```
figure;
hold on;
legs = cell(1,numel(Nrow));
for itr = (1:numel(Nrow))
    plot(Ncol,Cvals(itr,:),'LineWidth',1,'Marker','x');
    legs{itr} = ['# of rows = ',sprintf('%g',Nrow(itr))];
end
grid on;
set(gca,'LineWidth',1,'FontSize',13);
legend(legs(:),'Location','southeast');
xlabel('# of columns');
ylabel('Spin-spin correlation');
```



The spin-spin correlation, as a function of the number of columns with fixed number of rows, shows an oscillatory behaviour with period 4, and converges slowly to a value between -0.095 and -0.1, in the limit of infinite columns. A Monte Carlo calculation predicts the nearest-neighbour correlator $\langle \widehat{S}^z_{(i,j)} \widehat{S}^z_{(i,j+1)} \rangle \simeq -0.0987$ in the thermodynamic limit [A. F. Albuquerque and F. Alet, Phys. Rev. B **82**, 180408(R) (2010) or its arXiv version].

Exercise (b): Different shape of square lattice

In the above demonstration, we have considered the lattices whose rows are longer than columns. How about the lattices whose columns are longer than rows? Which result would be more accurate? Why?

Exercise (c): Toric code

Compute the spin-spin correlator for the Kitaev's toric code.

(*Hint*: it should be always zero, since the action of two $\widehat{S}^z$ operators onto the ground state yields an excited energy eigenstate.)

Exercise (d): Resonating AKLT loop (RAL) state

Compute the spin-spin correlator for the RAL state. It is the generalization of the RVB state: the RVB state is for the lattice of spin-1/2's, and the RAL state is for the lattice of spin-1's, having three states $|+1_i\rangle$, $|0_i\rangle$, and $|-1_i\rangle$. Similarly as in the PEPS construction of the RVB state, each local tensor is given by the contraction of

valence bond states and local projectors. The valence bond state here is the same as one for the RVB state. The local projector is different:

$$\begin{aligned}\widehat{P}_i = &|+1_i\rangle(\langle e_{il}, e_{iu}, \Uparrow_{id}, \Uparrow_{ir}| + \langle e_{il}, \Uparrow_{iu}, e_{id}, \Uparrow_{ir}| + \cdots) \\ &+ |-1_i\rangle(\langle e_{il}, e_{iu}, \Downarrow_{id}, \Downarrow_{ir}| + \langle e_{il}, \Downarrow_{iu}, e_{id}, \Downarrow_{ir}| + \cdots) \\ &+ \frac{1}{\sqrt{2}}|0_i\rangle(\langle e_{il}, e_{iu}, \Uparrow_{id}, \Downarrow_{ir}| + \langle e_{il}, e_{iu}, \Downarrow_{id}, \Uparrow_{ir}| + \langle e_{il}, \Uparrow_{iu}, e_{id}, \Downarrow_{ir}| + \langle e_{il}, \Downarrow_{iu}, e_{id}, \Uparrow_{ir}| + \cdots)\end{aligned}$$

For details, refer to Wei Li, Shuo Yang, Meng Cheng, Zheng-Xin Liu, and Hong-Hao Tu, Phys. Rev. B **89**, 174411 (2014), its arXiv version, or the local copy in our group website.