Tensor renormalization group and simple update ground-state search

Author: Seung-Sup Lee

In this tutorial, we implement the **tensor renormalization group (TRG)** method to compute the correlation functions of a projected entangled-pair state (PEPS) on a square or honeycomb lattice with periodic boundary condition. Also, we study the **simple update** method of finding the ground state of a Heisenberg model on a honeycomb lattice. We will largely refer to Secs. II A and II C of Gu2008 [Z.-C. Gu, M. Levin, and X.-G. Wen, Phys. Rev. B **78**, 205116 (2008) or its arXiv version] for the TRG, and to Jiang2008 [H. C. Jiang, Z. Y. Weng, and T. Xiang, Phys. Rev. Lett. **101**, 090603 (2008) or its arXiv version].

Exercise (a): Complete `TRG_Square_Ex.m` and `TRG_Honeycomb_Ex.m`

Complete two functions zipped together with this document. They are desinged to perform the TRG on a square lattice and on a honeycomb lattice, respectively. **Complete the parts which are enclosed by the comments `TODO - Exercise (a).`** Once you complete the function, you can follow the first demonstration below.

There are several remarks.

- **Leg order convention**: The function `TRG_Square_Ex.m` receives the input ket tensors `A` and `B` whose legs are organized as in the previous tutorial T12: left-up-(physical)-down-right. (For double tensors, or equivalently transfer operators, the physical leg is absent.) However, inside the function, the legs are permuted to be in counter-clockwise order, so that the legs with the same indices are contracted together. This leg order convention largely simplifies code writing. On the other hand, the function `TRG_Honeycomb_Ex.m` receives the input of local ket tensors whose legs are already organized in the counter-clockwise order.
- **Norm of tensors**: At each RG step, the number of sites in the lattice decreases by 2 in the square lattice case and by 3 in the honeycomb lattice case. Rephrasing, the number of physical sites represented by a single coarse-grained tensor increases by 2 and by 3, respectively, after each RG step. Without proper rescaling, the norm of the tensor may increase as a exponential of an exponential (!) of the RG step iteration. So the total norm often numerically diverges after a few steps. To deal with this numerically, the functions compute the squared norm per lattice site, i.e., $(\langle\Psi|\Psi\rangle)^{1/N}$ where $|\Psi\rangle$ is the total state and N is the number of lattice sites.
- **Last contraction step**: The RG steps are taken until we have a PEPS which is small enough to contract exactly. For the square lattice case, it is 2 by 2; for the honeycomb lattice case, it is of 2 sites.
- There is a new function `PEPS/contPlaq.m` which contracts tensors arranged around a plaquette. At each RG step of the TRG, the tensors are decomposed via the SVD. Then the decomposed tensors, placed around a plaquette, are contracted. Thus using this function would simplify code writing.

Correlation function 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 $(i,j+1)$ for the RVB state on a square lattice. (The honeycomb lattice code, `TRG_Honeycomb_Ex.m`, will be used in the next

demonstration below.) The RVB state is considered in the last tutorial T12. While in T12 the small system (a strip of a few hundred sites) with open boundary condition is considered, here we consider very large systems with periodic boundary condition.

To obtain the PEPS representation of the RVB state, we adopt the lines used in the demonstration of T12.

```
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]);

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 (out)
        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

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

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

Compute the spin-spin correlation function.

```
Nkeep = 50;
log2L = 10; % lattice size = 2^log2L by 2^log2L

[Nsq,SzSzval] = TRG_Square_Ex (A,A,log2L,Nkeep,Sz,Sz);
```

```
TERG on a square lattice: size = 2^10 x 2^10, Nkeep = 50
21-06-29 13:15:14 | Start
21-06-29 13:15:14 | #01/18
21-06-29 13:15:14 | #02/18
21-06-29 13:15:17 | #03/18
21-06-29 13:15:23 | #04/18
21-06-29 13:15:54 | #05/18
21-06-29 13:16:27 | #06/18
21-06-29 13:17:00 | #07/18
21-06-29 13:17:33 | #08/18
```

```
21-06-29 13:18:05 | #09/18
21-06-29 13:18:37 | #10/18
21-06-29 13:19:09 | #11/18
21-06-29 13:19:42 | #12/18
21-06-29 13:20:15 | #13/18
21-06-29 13:20:47 | #14/18
21-06-29 13:21:20 | #15/18
21-06-29 13:21:52 | #16/18
21-06-29 13:22:25 | #17/18
21-06-29 13:22:57 | #18/18
21-06-29 13:22:58 | Squared norm per site = 2.16603, Correlation func. = -0.112885
Elapsed time: 463.6s, CPU time: 3345s, Avg # of cores: 7.214
```

Exercise (b): Complete `SimpleUp_Honeycomb_Ex.m`

There is a function `SimpleUp_Honeycomb_Ex.m` zipped together with this document. The function is designed to find the ground state, as a PEPS, of an infinite system on a honeycomb lattice whose Hamiltonian consists of nearest-neighbor interaction terms. Keep in mind that the simple update method is the generalization of the infinite time-evolving block decimation (iTEBD) method, a 1D method covered in the previous tutorial T08, to two dimensions.

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

The ground state of the Heisenberg model on a honeycomb lattice

Let's compute the ground state of the Heisenberg model on a honeycomb lattice, by using the simple update method.

```
clear

% % system parameters
z = 3; % coordination number = number of nearest neighbors
[S,I] = getLocalSpace('Spin',1/2); % local operators
% Heisenberg interaction as two-site gate S*S
HSS = contract(S,3,2,permute(conj(S),[3 2 1]),3,2);

% % simple update parameters
Nkeep = 8;
beta_init = 1e-1; % initial imaginary time step size
beta_fin = 1e-4; % final imaginary time step size
Nstep = 1e4; % number of imaginary time steps
betas = beta_init*((beta_fin/beta_init).^linspace(0,1,Nstep));
% discrete imaginary time steps; decays slowly but exponentially
```

Here we use very small maximum bond dimension, `Nkeep = 8`, contrary to previously demonstrated methods. It is because the computational time complexity of the simple update method, $O(D^{z+1}d^2 + D^3d^6)$, increases quickly as the bond dimension D increases. Here z is coordination number and d is local space dimension.

The simple update method is the generalization of the iTEBD for two dimensions. It considers an ansatz wavefunction consists of Γ tensors (having physical legs) and Λ tensors (connecting Γ tensors, carrying the information of entanglement at the bonds). We initialize these tensors with random numbers.

```
% initialize Lambda and Gamma tensors
Lambda = cell(1,z); % for the bonds in three different directions
Lambda(:) = {rand(Nkeep,1)};
GA = rand([Nkeep*[1 1 1],size(I,2)]); % Gamma for sublattice A sites
GB = GA; % Gamma for sublattice B sites
```

We use the leg order convention for the Γ tensors such that the physical leg is placed at the last; see the documentation of `SimpleUp_Honeycomb_Ex.m` and `TRG_Honeycomb_Ex.m`.

Run the simple update method.

```
[Lambda,GA,GB,Es] = SimpleUp_Honeycomb_Ex (Lambda,GA,GB,HSS,Nkeep,betas);
```

```
21-06-29 13:22:58 | Start
21-06-29 13:23:14 | #3000/10000, Measured energy = -0.355693
21-06-29 13:23:25 | #6000/10000, Measured energy = -0.359339
21-06-29 13:23:36 | #9000/10000, Measured energy = -0.359798
21-06-29 13:23:39 | #10000/10000, Measured energy = -0.359831
Elapsed time: 41.23s, CPU time: 48.97s, Avg # of cores: 1.188
```

Note that the 4th output `Es` is the energy across a bond, measured just before applying imaginary time evolution step onto the bond. The "environment" tensors, which are not directly associated with the bond of interest, are **not** in any canonical form. Thus this on-the-fly measurement of the energy can be inaccurate.

To have a better estimate of the ground-state energy, we use the TRG. For this, we split the singular value tensors, `Lambda`, into two by taking their square roots, and then absorb the square roots to the `Gamma` tensors. Then the ground state is expressed by two rank-4 local tensors `TA` and `TB`.

```
TA = GA; TB = GB;
for it3 = (1:z)
    TA = contract(TA,z+1,it3,diag(sqrt(Lambda{it3})),2,2,[(1:it3-1),z+1,(it3:z)]);
    TB = contract(TB,z+1,it3,diag(sqrt(Lambda{it3})),2,1,[(1:it3-1),z+1,(it3:z)]);
end
```

Then run the TRG.

```
E_GS = zeros(3,1); % for three different bond directions
rgstep = 11; % large enough number, to estimate the value in thermodynamic limit
Nkeep2 = 50; % larger bond dimension than in the simple update
[~,E_GS(1)] = TRG_Honeycomb_Ex (TA,TB, ...
    rgstep,Nkeep2,S,permute(conj(S),[3 2 1]));
```

```
TERG on a honeycomb lattice: rgstep = 11, Nkeep = 50
21-06-29 13:26:30 | Start
21-06-29 13:28:02 | #01/11
21-06-29 13:28:19 | #02/11
21-06-29 13:28:37 | #03/11
21-06-29 13:28:53 | #04/11
21-06-29 13:29:10 | #05/11
21-06-29 13:29:27 | #06/11
21-06-29 13:29:43 | #07/11
21-06-29 13:30:00 | #08/11
21-06-29 13:30:16 | #09/11
21-06-29 13:30:32 | #10/11
21-06-29 13:30:48 | #11/11
```

```
21-06-29 13:30:48 | Squared norm per site = 1.41877, Correlation func. = -0.365723
Elapsed time: 257.9s, CPU time: 1968s, Avg # of cores: 7.634
```

Note that the last input to `TRG_Honeycomb_Ex` is the Hermitian conjugate of spin operator.

```
[~,E_GS(2)] = ...
    TRG_Honeycomb_Ex (permute(TA,[2 3 1 4]),permute(TB,[2 3 1 4]), ...
    rgstep,Nkeep2,S,permute(conj(S),[3 2 1]));
```

```
TERG on a honeycomb lattice: rgstep = 11, Nkeep = 50
21-06-29 13:30:48 | Start
21-06-29 13:32:24 | #01/11
21-06-29 13:32:42 | #02/11
21-06-29 13:32:59 | #03/11
21-06-29 13:33:16 | #04/11
21-06-29 13:33:33 | #05/11
21-06-29 13:33:50 | #06/11
21-06-29 13:34:06 | #07/11
21-06-29 13:34:23 | #08/11
21-06-29 13:34:39 | #09/11
21-06-29 13:34:56 | #10/11
21-06-29 13:35:12 | #11/11
21-06-29 13:35:12 | Squared norm per site = 1.41877, Correlation func. = -0.365858
Elapsed time: 264s, CPU time: 2007s, Avg # of cores: 7.602
```

```
[~,E_GS(3)] = ...
    TRG_Honeycomb_Ex (permute(TA,[3 1 2 4]),permute(TB,[3 1 2 4]), ...
    rgstep,Nkeep2,S,permute(conj(S),[3 2 1]));
```

```
TERG on a honeycomb lattice: rgstep = 11, Nkeep = 50
21-06-29 13:35:12 | Start
21-06-29 13:36:45 | #01/11
21-06-29 13:37:01 | #02/11
21-06-29 13:37:17 | #03/11
21-06-29 13:37:33 | #04/11
21-06-29 13:37:50 | #05/11
21-06-29 13:38:07 | #06/11
21-06-29 13:38:24 | #07/11
21-06-29 13:38:41 | #08/11
21-06-29 13:38:58 | #09/11
21-06-29 13:39:15 | #10/11
21-06-29 13:39:33 | #11/11
21-06-29 13:39:33 | Squared norm per site = 1.41877, Correlation func. = -0.365993
Elapsed time: 260.8s, CPU time: 1987s, Avg # of cores: 7.617
```

To obtain the energy across different bond directions, we have rotated the tensors `TA` and `TB` accordingly.

```
disp(E_GS)
```

```
   -0.3657
   -0.3659
   -0.3660
```

These energy values are close to each other, but not exactly the same. It is similar as that in the iTEBD calculation the even- and odd-bond energies are different and oscillating with iterations. So we average them.

```
disp(mean(E_GS))
```

```
   -0.3659
```

This is the ground-state energy **per bond**. On the other hand, Jiang2008 provides the value **per site**. To make comparison, we multiply 3/2 to the value per bond.

```
disp(mean(E_GS)*(3/2))
```

```
    -0.5488
```

The error against the published result -0.5506 (for the bond dimension $D = 8$, in Jiang2008) is smaller than 0.4%.