[Solution] DMRG: Single-site update for ground state search

Author: Seung-Sup Lee

Solution to Exercise (a): Initialize MPS with the iterative diagonalization result

First, we repeat the part in which parameters and operators are defined.

```
clear

% system parameter
J = -1; % coupling strength
L = 40; % number of sites in a chain

% DMRG parameter
Nkeep = 30; % bond dimension
Nsweep = 4; % number of pairs of left+right sweeps

% Local operators
[S,I] = getLocalSpace('Spin',1/2);

% % MPO formulation of Hamiltonian
% Hamiltonian tensor for each chain site
Hloc = cell(4,4);
Hloc(:) = {zeros(size(I))};
Hloc{1,1} = I;
Hloc{2,1} = S(:,:,1);
Hloc{3,1} = S(:,:,3);
Hloc{4,2} = J*S(:,:,1)';
Hloc{4,3} = J*S(:,:,3)';
Hloc{end,end} = I;
Hloc = cell2mat(reshape(Hloc,[1 1 size(Hloc,1) size(Hloc,2)]));

% full chain
Hs = cell(1,L);
Hs(:) = {Hloc};
Hs{1} = Hs{1}(:,:,end,:); % choose the last components of the left leg
Hs{end} = Hs{end}(:,:,:,1); % choose the first components of the right leg
```

Now we perform iterative diagonalization with the MPO Hamiltonian. At each iteration, the Hamiltonian is obtained by contracting the MPO tensors with the MPS tensors (bras and kets). The contraction result is rank-3, having a leg pointing right. This leg accounts for the interaction terms involving operators at later sites, not included in the Hilbert space up to the current iteration. To obtain the Hamiltonian that only involves the sites included so far, we project the right leg onto its first index. By doing this, the Hamiltonian becomes rank-2 (as the projected leg becomes a dummy leg with singleton dimension) and can be diagonalized.

```
Minit = cell(1,L);
```

```
% tensors for the vaccum (i.e., dummy leg)
Hprev = 1; % initialize Hamiltonian with 1, as we will use MPO
Aprev = 1; % identity tensor for the dummy leg

for itN = (1:L)
    % add new site
    Anow = getIdentity(Aprev,2,I,2,[1 3 2]);
    Hnow = updateLeft(Hprev,3,Anow,Hs{itN},4,Anow);

    Hmat = Hnow(:,:,1);
    [V,D] = eig((Hmat+Hmat')/2);
    [D,ids] = sort(diag(D),'ascend');
    if itN < L
        Ntr = min([numel(D);Nkeep]);
    else
        Ntr = 1;
    end
    V = V(:,ids(1:Ntr));

    Anow = contract(Anow,3,2,V,2,1,[1 3 2]);

    Minit{itN} = Anow;

    Hprev = contract(Hnow,3,2,V,2,1);
    Hprev = contract(V',2,2,Hprev,3,1,[1 3 2]);
    Aprev = Anow;
end
```

`M` represents the MPS for the ground state out of the iterative diagonalization.

Solution to Exercise (b): Complete the single-site DMRG function

Check out the funciton `DMRG_GS_1site.m` under the `DMRG` sub-directory. Compare with your implementation of `DMRG_GS_1site_Ex.m`!

```
[M0,E0,Eiter] = DMRG_GS_1site(Minit,Hs,Nkeep,Nsweep);
```

```
22-09-23 21:05:13 | Single-site DMRG: ground state search
22-09-23 21:05:13 | # of sites = 40, Nkeep = 30, # of sweeps = 4 x 2
22-09-23 21:05:14 | Sweep #1/8 (right -> left) : Energy = -12.55385
22-09-23 21:05:14 | Sweep #2/8 (left -> right) : Energy = -12.5539
22-09-23 21:05:14 | Sweep #3/8 (right -> left) : Energy = -12.5539
22-09-23 21:05:14 | Sweep #4/8 (left -> right) : Energy = -12.5539
22-09-23 21:05:14 | Sweep #5/8 (right -> left) : Energy = -12.5539
22-09-23 21:05:14 | Sweep #6/8 (left -> right) : Energy = -12.5539
22-09-23 21:05:15 | Sweep #7/8 (right -> left) : Energy = -12.5539
22-09-23 21:05:15 | Sweep #8/8 (left -> right) : Energy = -12.5539
Elapsed time: 1.47s, CPU time: 15.9s, Avg # of cores: 10.82
```

```
E0_exact = 0.5 - (1/2/sin(pi/2/(L+1))); % exact value
disptime(['Exact GS energy = ',sprintf('%.5g',E0_exact),', DMRG = ', ...
    sprintf('%.5g',E0),', error = ',sprintf('%.5g',E0-E0_exact)]);
```

```
22-09-23 21:05:15 | Exact GS energy = -12.554, DMRG = -12.554, error = 9.6503e-07
```