[Solutions] MPO representation of time evolution operators

Author: Seung-Sup Lee

Solution to Exercise (a): MPO for the first-order Trotterization

We first construct a two-site time evolution gate and decompose it into two one-site tensors.

```
clear

J = -1; % coupling strength
dt = 0.05; % step size
N = 6; % take a small value for verification

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

% Sx*Sx + Sy*Sy interaction
Hxy = J*contract(S(:,:,[1 3]),3,3, ...
    permute(conj(S(:,:,[1 3])),[2 1 3]),3,3);

ds = size(Hxy);
Hxy_mat = reshape(permute(Hxy,[1 3 2 4]), ...
    [prod(ds([1 3])), prod(ds([2 4]))]);
[V,D] = eig((Hxy_mat+Hxy_mat')/2);
[D,ids] = sort(diag(D),'ascend');
V = V(:,ids);

% time evolution over dt
Udt = permute(reshape(V*diag(exp((-1i*dt)*D))*V', ...
    ds([1 3 2 4])),[1 3 2 4]);

% decompose Udt
[Ldt,Sdt,Rdt] = svdTr(Udt,4,[1 2],[],[]);
Rdt = contract(diag(Sdt),2,2,Rdt,3,1,[2 3 1]);
```

Place the one-site tensors in a 2-by-L cell array. Its first (second) row is for the tensors from the two-site gates from $\hat{H}_{\text{even}}$ ($\hat{H}_{\text{odd}}$).

```
Os = cell(2,N);
Os(1,1) = {I};
Os(1,2:2:N-1) = {Ldt};
Os(1,3:2:N) = {Rdt};
if isempty(Os{1,end})
    Os{1,end} = I;
end
Os(2,1:2:N-1) = {Ldt};
Os(2,2:2:N) = {Rdt};
if isempty(Os{2,end})
    Os{2,end} = I;
```

```
end
```

Contract the operators on the same column via their vertical (i.e., physical) legs, to construct an MPO.

```
MPO = cell(1,N);
for itN = (1:N)
    MPO{itN} = contract(Os{2,itN},3,2,Os{1,itN},3,1);
    if mod(itN,2) == 1
        MPO{itN} = permute(MPO{itN},[1 3 4 2]);
    else
        MPO{itN} = permute(MPO{itN},[1 3 2 4]);
    end
end
```

We verify the result by explicitly constructing $\widehat{H}_{\text{odd}}$ and $\widehat{H}_{\text{even}}$ in their large matrix form. We also bring the MPO into the matrix form.

```
Hodd = 0;
Heven = 0;
U_MPO = 1;
Aprev = 1;

for itN = (1:N)
    Anow = getIdentity(Aprev,2,I,2,[1 3 2]);

    Hodd = updateLeft(Hodd,2,Anow,[],[],Anow);
    Heven = updateLeft(Heven,2,Anow,[],[],Anow);

    % S*S interaction
    if itN > 1
        HSS = updateLeft(Sprev,3,Anow, ...
            permute(conj(S(:,:,[1 3])),[2 1 3]),3,Anow);

        if mod(itN,2) == 0
            Hodd = Hodd + J*HSS;
        else
            Heven = Heven + J*HSS;
        end
    end

    U_MPO = updateLeft(U_MPO,3,Anow,MPO{itN},4,Anow);

    Sprev = updateLeft([],[],Anow,S(:,:,[1 3]),3,Anow);
    Aprev = Anow;
end
```

Compute $\exp(-\mathrm{i}\widehat{H}_{\text{odd}}\Delta t)\exp(-\mathrm{i}\widehat{H}_{\text{even}}\Delta t)$ (= U_exp) and compare it with the MPO result.

```
[Vodd,Dodd] = eig(Hodd);
```

```
Uodd = Vodd*diag(exp((-1i*dt)*diag(Dodd)))*Vodd';
[Veven,Deven] = eig(Heven);
Ueven = Veven*diag(exp((-1i*dt)*diag(Deven)))*Veven';
U_exp = Uodd*Ueven;

disp(size(U_MPO));
```

```
    64    64
```

```
disp(size(U_exp));
```

```
    64    64
```

```
disp(max(abs(U_MPO(:)-U_exp(:))));
```

```
   1.7764e-15
```

They agree up to double precision.

Solution to Exercise (b): MPO for the first-order Taylor expansion

We first define the bulk tensor of the MPO representation of $\hat{H}_{\mathrm{XY}}$. Here we multiply $-\mathrm{i}\Delta t$ to the terms that include the coupling strength J.

```
clear

J = -1; % coupling strength
dt = 0.05; % step size
N = 6; % take a small value for verification

[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} = (-1i*dt*J)*S(:,:,1)';
Hloc{4,3} = (-1i*dt*J)*S(:,:,3)';
Hloc{end,end} = I;
```

Note that we have not converted `Hloc` into a rank-4 tensor yet, for future convenience.

Following Sec. 5.2 of Schollwoeck2011 [U. Schollwöck, Ann. Phys. **326**, 96 (2011)], we define the bulk tensor of the MPO representation of $\hat{I} - \mathrm{i}\hat{H}_{\mathrm{XY}}\Delta t$.

```
Ubulk = cell(1+size(Hloc));
Ubulk(:) = {zeros(size(I))};
Ubulk{1,1} = I;
```

```
Ubulk(2:end,2:end) = Hloc;
```

The tensor at the left boundary can be defined by making a "row vector" made of `Ubulk(1,1)` and `Ubulk(end,2:end)`. On the other hand, the tensor at the right boundary is from a "column vector" made of `Ubulk(1,1)` and `Ubulk(2:end,2)`.

```
Ulb = [Ubulk(1,1),Ubulk(end,2:end)];
Urb = [Ubulk(1,1);Ubulk(2:end,2)];

MPO = cell(1,N);
MPO(2:end-1) = {cell2mat(reshape(Ubulk,[1 1 size(Ubulk)]))};
MPO{1} = cell2mat(reshape(Ulb,[1 1 size(Ulb)]));
MPO{end} = cell2mat(reshape(Urb,[1 1 size(Urb)]));
```

We verify this construction by comparing with the explicit construction.

```
H = 0;
U_MPO = 1;
Aprev = 1;

for itN = (1:N)
    Anow = getIdentity(Aprev,2,I,2,[1 3 2]);

    H = updateLeft(H,2,Anow,[],[],Anow);

    % S*S interaction
    if itN > 1
        HSS = updateLeft(Sprev,3,Anow, ...
            permute(conj(S(:,:,[1 3])),[2 1 3]),3,Anow);
        H = H + J*HSS;
    end

    U_MPO = updateLeft(U_MPO,3,Anow,MPO{itN},4,Anow);

    Sprev = updateLeft([],[],Anow,S(:,:,[1 3]),3,Anow);
    Aprev = Anow;
end

U_exp = eye(size(Aprev,2)) - (1i*dt)*H;

disp(size(U_MPO));
```

```
    64    64
```

```
disp(size(U_exp));
```

```
    64    64
```

```
disp(max(abs(U_MPO(:)-U_exp(:))));
```

```
   3.4694e-18
```

They agree up to double precision.