[Solution] Diagonalize many-body Hamiltonians

Author: Seung-Sup Lee

Exercise (a): Spin-1/2 Heisenberg triangle (pen-and-paper)

We can rephrase the Hamiltonian of the system in terms of the squared spin operators,

$$\hat{H} = J(\hat{\vec{S}}_1 \cdot \hat{\vec{S}}_2 + \hat{\vec{S}}_1 \cdot \hat{\vec{S}}_3 + \hat{\vec{S}}_2 \cdot \hat{\vec{S}}_3) = (J/2)\left[\hat{\vec{S}}_{\rm tot}^2 - \sum_{\ell=1}^{3} \hat{\vec{S}}_\ell^2\right],$$

where $\vec{S}_{\rm tot} = \sum_{\ell=1}^{3} \vec{S}_\ell$. Since all the spins are spin-1/2's, the action of each $\hat{\vec{S}}_\ell^2$ always gives a prefactor $(1/2)(1/2+1) = 3/4$.

The total spin $S_{\rm tot}$ can be identified by the relation for adding two SU(2) spins,

$$S_1 \otimes S_2 = |S_1 - S_2| \oplus |S_1 - S_2| + 1 \oplus \cdots \oplus S_1 + S_2.$$

Hence we get

$$1/2 \otimes 1/2 \otimes 1/2 = (0 \oplus 1) \otimes 1/2 = 1/2 \oplus 1/2 \oplus 3/2.$$

There are two sectors of $S_{\rm tot} = 1/2$, each of which has multiplet dimension two. Such multiple occurence of symmetry secotrs with the same symmetry labels is called *outer multiplicity*. And there is a sector of $S_{\rm tot} = 3/2$ having four states. The action of $\hat{\vec{S}}_{\rm tot}^2$ yields a prefactor $3/4$ for the $S_{\rm tot} = 1/2$ sectors and $15/4$ for the $S_{\rm tot} = 3/2$ sector. Therefore, the Hamiltonian has four-fold degenerate eigenvalues $-3/4$ and another four-fold degenerate eigenvalues $3/4$.

New function: `Tensor/updateLeft.m`

Here we introduce a function, which is useful for solving the coding exercises below. This function performs a single iteration within the iterative "zipper" contraction, for computing matrix elements, overlaps, and expectation values with respect to matrix product states (MPSs). Please refer to its documentation for its usage.

Exercise (b): Spin-1/2 Heisenberg triangle (coding)

```
clear
J = 1; % interaction strength
N = 3; % number of spins
```

We define the operators (spin operator `S` and identity `I`) acting on each spin site. And we define a cell array `Ss`, where `Ss{n}` contains the n-th spin operator in the many-body basis spanned by the second leg of the identity tensor defined for a given iteration.

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

```
Ss = cell(1,N);
```

To make the code compact, we can think of the Hamiltonian and the identity tensor for the vacuum, corresponding to the left dummy leg of the first tensor of MPS. Since there is nothing and the vacuum space has only a single state (i.e., vacuum state), we can set the Hamiltonian as 0 and the identity as 1.

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

Now we are ready to construct the Hamiltonian iteratively.

```
for itN = (1:N)
```

Each iteration starts with defining the rank-3 identity (i.e., isometry) tensor `Anow` for the current iteration. The first (i.e., left) leg of `Anow` spans the Hilbert spaces from the previous iteration, covering from the first to the (`itN`−1)-th spin; the second (i.e., right) leg spans the space covering from the first to the `itN`-th spin. The third (i.e., bottom) leg spans the physical space for the `itN`-th spin.

```
    Anow = getIdentity(Aprev,2,I,2,[1 3 2]);
```

Contract the Hamiltonian up to the last iteration with the ket tensor `Anow` and its Hermitian conjugate.

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

    for itN2 = (1:itN-1)
```

Compute the spin-spin interaction term $\hat{\vec{S}}_{\text{itN2}} \cdot \hat{\vec{S}}_{\text{itN}}$, multiply interaction strength prefactor, and add to the Hamiltonian. Don't forget to take the Hermitian conjugation, `permute(conj(S),[2 1 3])`, for only one of the two spins involved. Otherwise, the third legs of the spin operators do not have consistent directions, and even worse, the contraction result does not represent $\hat{\vec{S}}_{\text{itN2}} \cdot \hat{\vec{S}}_{\text{itN}}$ properly.

```
        Hsp = updateLeft(Ss{itN2},3,Anow, ...
            permute(conj(S),[2 1 3]),3,Anow);
        H = H + J*Hsp;
    end
```

Update the spin operators to be represented in the new basis associated with the second leg of `Anow`.

```
    for itN2 = (1:itN)
        if itN2 < itN
            Ss{itN2} = updateLeft(Ss{itN2},3,Anow,[],[],Anow);
        else
            Ss{itN2} = updateLeft([],[],Anow,S,3,Anow);
        end
    end
```

For the next iteration, replace `Aprev` with `Anow`.

```
    Aprev = Anow;
end
```

Diagonalize the Hamiltonian.

```
Es = sort(eig((H+H')/2),'ascend');
```

Here we **Hermitianize the Hamiltonian** by taking `(H+H')/2`. Of course, Hamiltonians should be Hermitian, in principle. But in practice, the numerical representation of the Hamiltonian is susceptible to noise, so that it can be slightly non-Hermitian. It will lead to complex-valued eigenvalues (which should be always real for physical problems) and eigenvectors (which should be real-valued for the real-valued Hamiltonians in most of our examples). They are not only wrong, but also a possible source of unnecessary computational overhead, since the algebra of complex numbers takes more computational cost than that of real numbers. Therefore, it is a good practice to Hermitianize the Hamiltonian before diagonalization.

```
disp(Es);
```

```
   -0.7500
   -0.7500
   -0.7500
   -0.7500
    0.7500
    0.7500
    0.7500
    0.7500
```

We find that the Hamiltonian has four-fold degenerate eigenvalues $-3/4$ and another four-fold degenerate eigenvalues $3/4$, which is consistent with the result of Exercise (a).

Exercise (c): Non-interacting tight-binding chain

```
clear
N = 11; % number of spins
t = exp(1i*(1:N-1)); % hopping amplitudes

[F,Z,I] = getLocalSpace('Fermion');

H = 0; % initialize Hamiltonian
Aprev = 1; % identity for the vacuum

for itN = (1:N)
    % rank-3 identity tensor for the current iteration
    Anow = getIdentity(Aprev,2,I,2,[1 3 2]);

    % contract the Hamiltonian up to the last iteration with
    % ket and bra tensors
    H = updateLeft(H,2,Anow,[],[],Anow);
```

So far it's similar with the above example on spin. Now it becomes different; we need to construct the hopping terms, instead of spin-spin interaction. Here we first contract $Z_{[\mathrm{itN}]}F_{[\mathrm{itN}]}$, then take the Hermitian conjugate

to get $F^\dagger_{[\mathrm{itN}]}Z_{[\mathrm{itN}]}$. ($F_{[\mathrm{itN}]}$ and $Z_{[\mathrm{itN}]}$ are the tensors for, respectively, the particle annihilation opeator and the fermionic sign operator, acting at site `itN`.) Then we obtain the matrix elements of $-t_{\mathrm{itN}-1}F^\dagger_{[\mathrm{itN}]}Z_{[\mathrm{itN}]}F_{[\mathrm{itN}-1]}$, which represents $-t_{\mathrm{itN}-1}f^\dagger_{\mathrm{itN}}f_{\mathrm{itN}-1}$. The rest of the hopping term, $-t^*_{\mathrm{itN}-1}f^\dagger_{\mathrm{itN}-1}f_{\mathrm{itN}}$, can be obtained as the Hermitian conjugate.

```
    if itN > 1
        ZF = contract(Z,2,2,F,3,1);
        Hhop = (-t(itN-1))*updateLeft(Fprev,3,Anow, ...
            permute(conj(ZF),[2 1 3]),3,Anow);
        % hopping from the last site to the current site
        H = H + Hhop + Hhop';
    end

    % update operator for the next iteration
    Fprev = updateLeft([],[],Anow,F,3,Anow);

    Aprev = Anow; % to be used for the next iteration
end
Es = sort(eig((H+H')/2),'ascend');
disp(Es(1:7).');
```

```
   -6.5958   -6.5958   -6.0781   -6.0781   -6.0781   -6.0781   -5.5958
```

We see that there are two-fold degenerate ground states with energy -6.5958 and four-fold lowest excited states with energy -6.0781. The ground-state energy and degeneracy can be also cross-checked by using `Util/nonIntTB.m`:

```
[E_G, d_G, e_1p] = nonIntTB(t);
E_G % ground-state energy
```

```
E_G = -6.5958
```

```
E_G - Es(1) % agree up to double precision
```

```
ans = -7.9936e-15
```

```
d_G % ground-state degeneracy
```

```
d_G = 2
```

In such non-interacting system, the lowest excited states are given, on top of each ground state, either by filling the single-particle level of the lowest positive enrgy (as a one-particle excitation) or by emptying out the level of the highest negative energy (as a one-hole excitation). So the lowest excited states have four-fold degeneracy, i.e., 2 (ground-state degeneracy) times 2 (particle or hole excitation).

We also find the consistency between two calculations regarding the lowest excited states:

```
E_G + min(e_1p(e_1p>0)) % one-particle excitation
```

```
ans = -6.0781
```

```
E_G + min(e_1p(e_1p>0)) - Es(3) % agree up to double precision
```

```
ans = 3.2863e-14
```

```
E_G - max(e_1p(e_1p<0)) % one-hole excitation
```

```
ans = -6.0781
```

```
E_G - max(e_1p(e_1p<0)) - Es(3) % agree up to double precision
```

```
ans = 3.2863e-14
```