Set 4

Xiongxiao Wang, Pulkit Kukreja

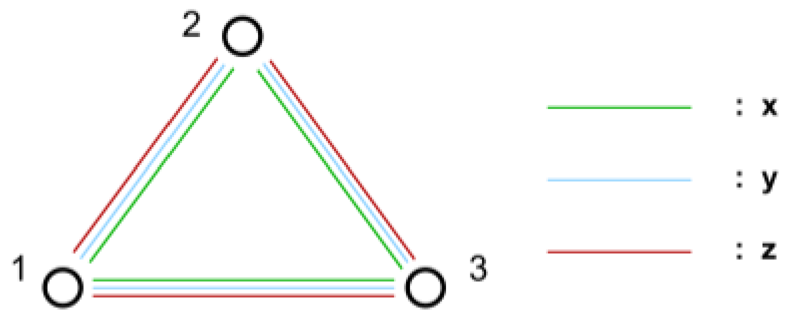
Exercise 1 : Spin-models on a three-sites cluster

Consider the following (general) Hamiltonian for a spin-model on a three-site cluster:

$$H = -\sum_{ij\alpha} J_{ij}^{\alpha} S_i^{\alpha} S_j^{\alpha}$$

with i,j =1,2,3 (i<j in $\sum_{ij}$) and $\alpha = x, y, z$

To visualize the model, a colour code for the x,y and z components of the spincouplings turns out to be useful, see the figure.



a) Rewrite the Hamiltonian using the operators

$$S_i^{\pm} = S_i^x \pm iS_i^y, and S_i^z$$

Hamiltonian can be rewritten as the form below:

$$H = -\sum_{ij\alpha} J_{ij}^{\alpha} S_i^{\alpha} S_j^{\alpha} = -\sum_{ij}(J_{ij}^x S_i^x S_j^x + J_{ij}^y S_i^y S_j^y + J_{ij}^z S_i^z S_j^z)$$

Using :$S_i^{\pm} = S_i^x \pm iS_i^y$

We get $S_i^x = \frac{S_i^+ + S_i^-}{2}$, $S_i^y = \frac{S_i^+ - S_i^-}{2i}$

Replace the terms S_i^x, S_i^y in Hailtonian by the equations above, we get:

$$H = -\sum_{ij}[\frac{J_{ij}^x}{4}(S_i^+ + S_i^-)(S_j^+ + S_j^-) - \frac{J_{ij}^y}{4}(S_i^+ - S_i^-)(S_j^+ - S_j^-) + J_{ij}^z S_i^z S_j^z]$$

Now set up (by hand!) the 8*8 Hamilton matrices $\overline{H}$ for the following three special cases:

The basis of states are | ↓↓↓> | ↓↓↑> | ↓↑↓> | ↓↑↑> | ↑↓↓> | ↑↓↑> | ↑↑↓> | ↑↑↑>

b) the Ising model, i.e. $J_{ij}^{\alpha} = J\delta_{\alpha z}$

$$H_{Ising} = -\sum_{ij\alpha} J_{ij}^{\alpha} S_i^{\alpha} S_j^{\alpha} = -J\sum_{ij} S_i^z S_j^z = -J(S_1^z S_2^z + S_2^z S_3^z + S_3^z S_1^z)$$

$$\overline{H}_{Ising} = -J\begin{bmatrix} 3/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3/2 \end{bmatrix}$$

c) the isotropic Heisenberg model, i.e. $J_{ij}^{\alpha} = J$

$H_{Heisenberg} = -J\sum_{ij\alpha} S_i^{\alpha} S_j^{\alpha}$

$= -J\sum_{ij}[\frac{1}{4}(S_i^+ + S_i^-)(S_j^+ + S_j^-) - \frac{1}{4}(S_i^+ - S_i^-)(S_j^+ - S_j^-) + S_i^z S_j^z]$

$= -J\sum_{ij}[\frac{1}{2}(S_i^+ S_j^- + S_i^- S_j^+) + S_i^z S_j^z]$

$= -J[\frac{1}{2}(S_1^+ S_2^- + S_1^- S_2^+ + S_2^+ S_3^- + S_2^- S_3^+ + S_3^+ S_1^- + S_3^- S_1^+) + S_1^z S_2^z + S_2^z S_3^z + S_3^z S_1^z]$

The basis of states are | ↓↓↓> | ↓↓↑> | ↓↑↓> | ↓↑↑> | ↑↓↓> | ↑↓↑> | ↑↑↓> | ↑↑↑>

$$\overline{H}_{Heisenberg} = -J\begin{bmatrix} 3/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1/4 & 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 1/2 & -1/4 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1/4 & 0 & 1/2 & 1/2 & 0 \\ 0 & 1/2 & 1/2 & 0 & -1/4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & -1/4 & 1/2 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 & -1/4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3/4 \end{bmatrix}$$

d) a model with $J_{12}^x = J_{23}^y = J_{31}^z = J$ and all other $J_{ij}^{\alpha} = 0$

The basis of states are | ↓↓↓> | ↓↓↑> | ↓↑↓> | ↓↑↑> | ↑↓↓> | ↑↓↑> | ↑↑↓> | ↑↑↑>

$H_{model} = -J(S_1^x S_2^x + S_2^y S_3^y + S_3^z S_1^z)$

$= -J[\frac{1}{4}(S_1^+ + S_1^-)(S_2^+ + S_2^-) - \frac{1}{4}(S_2^+ - S_2^-)(S_3^+ - S_3^-) + S_3^z S_1^z]$
$= -J[\frac{1}{4}(S_1^+ S_2^+ + S_1^- S_2^+ + S_1^+ S_2^- + S_1^- S_2^-) - \frac{1}{4}(S_2^+ S_3^+ - S_2^- S_3^+ - S_2^+ S_3^- + S_2^- S_3^-) + S_3^z S_1^z]$

$$\overline{H}_{model} = -J \begin{bmatrix} 1/4 & 0 & 0 & 1/4 & 0 & 0 & -1/4 & 0 \\ 0 & -1/4 & 1/4 & 0 & 0 & 0 & 0 & -1/4 \\ 0 & 1/4 & 1/4 & 0 & 1/4 & 0 & 0 & 0 \\ 1/4 & 0 & 0 & -1/4 & 0 & 1/4 & 0 & 0 \\ 0 & 0 & 1/4 & 0 & -1/4 & 0 & 0 & 1/4 \\ 0 & 0 & 0 & 1/4 & 0 & 1/4 & 1/4 & 0 \\ -1/4 & 0 & 0 & 0 & 0 & 1/4 & -1/4 & 0 \\ 0 & -1/4 & 0 & 0 & 1/4 & 0 & 0 & 1/4 \end{bmatrix}$$

Exercise 2 : Hamilton matrices for spin models

Consider a spin model of the form

$$H = -\sum_{ij\alpha} J_{ij}^{\alpha} S_i^{\alpha} S_j^{\alpha}$$

with i,j = 1,...,N and $\alpha = x, y, z$

a)Write a program which sets up the Hamilton matrix for a model with artibtrary N and an arbitrary list of couplings J_{ij}^{α}. As discussed in the lecture, the x-,y- and z-links should be treated separatedly. For each link of the form $-J_{ij}^{\alpha} S_i^{\alpha} S_j^{\alpha}$, the action of $S_i^{\alpha} S_j^{\alpha}$ on the basis state $|n>$ gives another basis state $|l>$ (times a prefactor). With the equation for l given in the lecture, the Hamiltonian matrix can be set up very efficiently.

Assuming the size of spin model is N, and the number of links is L (number of terms in the Hamiltonian of the form $J_{pq}^{\alpha} S_p^{\alpha} S_q^{\alpha}$)

The idea is:

1.Loop over the basis states from n=0 to 2^N-1

2.Calculate the binary representation of n : $[n]_1 0 \rightarrow [z_N ... z_1]_2$

3.Loop over the links from j = 1 to L

4.if $\alpha = x$ or y, calculate $l = n + (1 - 2z_p)2^{p-1} + (1 - 2z_q)2^{q-1}$

5.add to the matrix element H_{ln} of the Hamilton matrix:

for $\alpha = x$: $-\frac{1}{4} J_{pq}^{x}$

for $\alpha = y$: $(2z_p - 1)(2z_q - 1)\frac{1}{4} J_{pq}^{y}$

for $\alpha = z$: $\overline{H}_{nn} = \frac{1}{4} J_{pq}^{z}(2z_p - 1)(2z_q - 1)$

In [1]:

```
import numpy as np

N = 3

J_x = np.random.randint(2, size=(2**N, 2**N))
J_y = np.random.randint(2, size=(2**N, 2**N))
J_z = np.random.randint(2, size=(2**N, 2**N))
```

In [2]:

```
def Hmatrix(N,J_x,J_y,J_z):
    n_states = 2**N
    matrix = np.zeros((n_states,n_states))
    for n in range(n_states):
        z = np.zeros(N,dtype = int)
        nz = n
        for z_i in range(N):
            if nz//2 >= 0:
                z[z_i] = nz%2
                nz = nz//2

        for p in range(1,N+1):
            for q in range(p+1,N+1):
                s = (2*z[p-1]-1)*(2*z[q-1]-1)
                x_link =  0.25*  J_x[p-1,q-1]
                y_link = -0.25*s*J_y[p-1,q-1]
                z_link =  0.25*s*J_z[p-1,q-1]

                l =n + (1-2*z[p-1])*2**(p-1) + (1-2*z[q-1])*2**(q-1)
                matrix[l,n]-= x_link
                matrix[l,n]-= y_link
                matrix[n,n]-= z_link
    return(matrix)
Hmatrix(N,J_x,J_y,J_z)
```

Out[2]:

```
array([[-0.25,  0.  ,  0.  ,  0.  ,  0.  ,  0.  , -0.25,  0.  ],
       [ 0.  ,  0.25,  0.  ,  0.  ,  0.  ,  0.  ,  0.  , -0.25],
       [ 0.  ,  0.  , -0.25,  0.  , -0.25,  0.  ,  0.  ,  0.  ],
       [ 0.  ,  0.  ,  0.  ,  0.25,  0.  , -0.25,  0.  ,  0.  ],
       [ 0.  ,  0.  , -0.25,  0.  ,  0.25,  0.  ,  0.  ,  0.  ],
       [ 0.  ,  0.  ,  0.  , -0.25,  0.  , -0.25,  0.  ,  0.  ],
       [-0.25,  0.  ,  0.  ,  0.  ,  0.  ,  0.  ,  0.25,  0.  ],
       [ 0.  , -0.25,  0.  ,  0.  ,  0.  ,  0.  ,  0.  , -0.25]])
```

b) Set up the Hamilton matrices for the three models discussed in exercise 1 (the three-site clusters) and calculate the eigenenergies for each model.

In [3]:

```
from numpy import linalg as LA
#Ising model, set N=3, J = -1

N =3

J_x_Ising = np.zeros((2**N,2**N))
J_y_Ising = np.zeros((2**N,2**N))
J_z_Ising = -np.ones((2**N,2**N))

H_Ising = Hmatrix(N,J_x_Ising,J_y_Ising,J_z_Ising)
w_Ising,v_Ising = LA.eig(H_Ising)
print(H_Ising)
print('eigenvalues are',w_Ising)
```

```
[[ 0.75  0.    0.    0.    0.    0.    0.    0.  ]
 [ 0.   -0.25  0.    0.    0.    0.    0.    0.  ]
 [ 0.    0.   -0.25  0.    0.    0.    0.    0.  ]
 [ 0.    0.    0.   -0.25  0.    0.    0.    0.  ]
 [ 0.    0.    0.    0.   -0.25  0.    0.    0.  ]
 [ 0.    0.    0.    0.    0.   -0.25  0.    0.  ]
 [ 0.    0.    0.    0.    0.    0.   -0.25  0.  ]
 [ 0.    0.    0.    0.    0.    0.    0.    0.75]]
eigenvalues are [ 0.75 -0.25 -0.25 -0.25 -0.25 -0.25 -0.25  0.75]
```

$$\overline{H}_{Ising} = -J\begin{bmatrix} 3/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3/2 \end{bmatrix}$$

Compared the result from exercise 1, the result is the same

In [4]:

```
#Heisenberg model, set N=3, J=-1

N=3

J_x_hei = -np.ones((2**N,2**N))
J_y_hei = -np.ones((2**N,2**N))
J_z_hei = -np.ones((2**N,2**N))


H_hei = Hmatrix(N,J_x_hei,J_y_hei,J_z_hei)
w_hei,v_hei = LA.eig(H_hei)
print(H_hei)
print('eigenvalues are',w_hei)
```

```
[[ 0.75  0.    0.    0.    0.    0.    0.    0.  ]
 [ 0.   -0.25  0.5   0.    0.5   0.    0.    0.  ]
 [ 0.    0.5  -0.25  0.    0.5   0.    0.    0.  ]
 [ 0.    0.    0.   -0.25  0.    0.5   0.5   0.  ]
 [ 0.    0.5   0.5   0.   -0.25  0.    0.    0.  ]
 [ 0.    0.    0.    0.5   0.   -0.25  0.5   0.  ]
 [ 0.    0.    0.    0.5   0.    0.5  -0.25  0.  ]
 [ 0.    0.    0.    0.    0.    0.    0.    0.75]]
eigenvalues are [-0.75  0.75 -0.75  0.75 -0.75 -0.75  0.75  0.75]
```

$$\overline{H}_{Heisenberg} = -J\begin{bmatrix} 3/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1/4 & 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 1/2 & -1/4 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1/4 & 0 & 1/2 & 1/2 & 0 \\ 0 & 1/2 & 1/2 & 0 & -1/4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & -1/4 & 1/2 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 & -1/4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3/4 \end{bmatrix}$$

Compared the result from exercise 1, the result is the same

In [5]:

```
#a model with  Jx12=Jy23=Jz31=J , set N = 3, J=-1

N=3

J_x_model = np.zeros((2**N,2**N))
J_y_model = np.zeros((2**N,2**N))
J_z_model = np.zeros((2**N,2**N))
J_x_model[0,1] = -1
J_y_model[1,2] = -1
J_z_model[0,2] = -1

H_model = Hmatrix(N,J_x_model,J_y_model,J_z_model)
w_model,v_model = LA.eig(H_model)
print(H_model)
print('eigenvalues are',w_model)
```

```
[[ 0.25  0.    0.    0.25  0.    0.   -0.25  0.  ]
 [ 0.   -0.25  0.25  0.    0.    0.    0.   -0.25]
 [ 0.    0.25  0.25  0.    0.25  0.    0.    0.  ]
 [ 0.25  0.    0.   -0.25  0.    0.25  0.    0.  ]
 [ 0.    0.    0.25  0.   -0.25  0.    0.    0.25]
 [ 0.    0.    0.    0.25  0.    0.25  0.25  0.  ]
 [-0.25  0.    0.    0.    0.    0.25 -0.25  0.  ]
 [ 0.   -0.25  0.    0.    0.25  0.    0.    0.25]]
eigenvalues are [ 0.4330127 -0.4330127 -0.4330127  0.4330127 -0.4330127  0.4330127
 -0.4330127  0.4330127]
```

$$\overline{H}_{model} = -J\begin{bmatrix} 1/4 & 0 & 0 & 1/4 & 0 & 0 & -1/4 & 0 \\ 0 & -1/4 & 1/4 & 0 & 0 & 0 & 0 & -1/4 \\ 0 & 1/4 & 1/4 & 0 & 1/4 & 0 & 0 & 0 \\ 1/4 & 0 & 0 & -1/4 & 0 & 1/4 & 0 & 0 \\ 0 & 0 & 1/4 & 0 & -1/4 & 0 & 0 & 1/4 \\ 0 & 0 & 0 & 1/4 & 0 & 1/4 & 1/4 & 0 \\ -1/4 & 0 & 0 & 0 & 0 & 1/4 & -1/4 & 0 \\ 0 & -1/4 & 0 & 0 & 1/4 & 0 & 0 & 1/4 \end{bmatrix}$$

Compared the result from exercise 1, the result is the same