Group 7

Weekly presentation

Aaina Bansal
Bingcheng He
Jiecheng Yang

TALENT Course @ Xinxiang, Henan

Theory is when one knows everything but nothing works.

Practice is when everything works but nobody knows why. In our group, ...

https://github.com/abansal92/pairingmodel

July 20, 2018

Status

Problem 1a:

Theorem

$$definition : \hat{S}_z = \frac{1}{2}\sum_{p\sigma}\sigma\ a^\dagger_{p\sigma}a_{p\sigma},\ \ \hat{H} = \sum_{p\sigma}(p-1)\ a^\dagger_{p\sigma}a_{p\sigma} - g\sum pq\hat{P}^\dagger_p\hat{P}^-_q$$

$$[\hat{H}_0, \hat{S}_z] = \frac{1}{2}\sum_{p\sigma}\sum_{p'\sigma'}(p-1)\sigma'[a^\dagger_{p\sigma}a_{p\sigma}, a^\dagger_{p'\sigma'}a_{p'\sigma'}] \tag{1}$$

which

$$[a^\dagger_{p\sigma}a_{p\sigma}, a^\dagger_{p'\sigma'}a_{p'\sigma'}] = [a^\dagger_{p\sigma}a_{p\sigma}, a^\dagger_{p'\sigma'}]a_{p'\sigma'} + a^\dagger_{p'\sigma'}[a^\dagger_{p\sigma}a_{p\sigma}, a_{p'\sigma'}] \tag{2}$$

substituting those two commutators into Eq(2),

$$[a^\dagger_{p\sigma}a_{p\sigma}, a^\dagger_{p'\sigma'}a_{p'\sigma'}] = a^\dagger_{p\sigma}a_{p'\sigma'}\delta_{pp'}\delta_{\sigma\sigma'} - a^\dagger_{p\sigma}a_{p'\sigma'}\delta_{pp'}\delta_{\sigma\sigma'} \tag{3}$$

substituting into original equation,

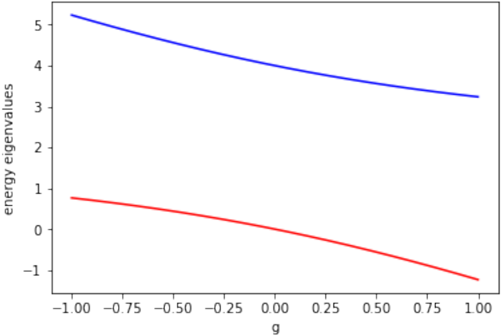
$$\begin{aligned}[\hat{H}_0, \hat{S}_z] &= \frac{1}{2}\sum_{p\sigma}\sum_{p'\sigma'}(p-1)\sigma'(a^\dagger_{p\sigma}a_{p'\sigma'}\delta_{pp'}\delta_{\sigma\sigma'} - a^\dagger_{p\sigma}a_{p'\sigma'}\delta_{pp'}\delta_{\sigma\sigma'}) \\ &= \frac{1}{2}\sum_{p\sigma}(p-1)\sigma(a^\dagger_{p\sigma}a_{p\sigma} - a^\dagger_{p\sigma}a_{p\sigma}) = 0\end{aligned} \tag{4}$$

Status

Problem 1b:

$$H = \begin{bmatrix} -g & -g \\ -g & 2d - g \end{bmatrix}$$

for g $\in$ [-1,1] and d = 2:

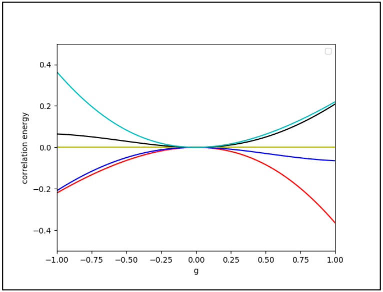


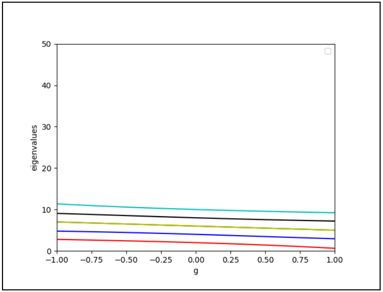
Status

Problem 1c:

$$H = \begin{bmatrix} 2d-g & -g/2 & -g/2 & -g/2 & -g/2 & 0 \\ -g/2 & 4d-g & -g/2 & -g/2 & 0 & -g/2 \\ -g/2 & -g/2 & 6d-g & 0 & -g/2 & -g/2 \\ -g/2 & -g/2 & 0 & 6d-g & -g/2 & -g/2 \\ -g/2 & 0 & -g/2 & -g/2 & 8d-g & -g/2 \\ 0 & -g/2 & -g/2 & -g/2 & -g/2 & 10-g/2 \end{bmatrix}$$

for g $\in$ [-1,1] and d = 1,2 ... 5: (6 eigenvalues)





Status

Problem 1d:

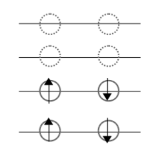
- Consider the states and quantum numbers
- Generate all the possible configurations either with normal counting or bits
- Select the cases satisfying the pairing model

```python
def paired_states(bra):
    # returns whether all the particles are paired or not in a given state
    x = []
    for i in np.arange(0,len(bra),2):
        x.append((bra[i]%2 != 0 and bra[i+1] == bra[i] + 1))
    res = x[0]
    for i in range(len(x)):
        res = res and x[i]
    return res

def ph_config(bra,ket):
    #returns number of ph excitations of a given state
    return len(bra) - len(set(bra) & set(ket))
```

- Assign the matrix elements with the above configuration
- Build the matrix and calculate the eigenvalues

Coding



$|11110000>$

1: unoccupied;
0: occupied
counting from right side
For example: 4 particles in 8 sp states

- Create $|11111111>$
- Assign 0 into any four bits, such as $|11010100>$
- Apply the pairing model and make the selection (6 sets)
- Generate the matrix according to the configration and solve the eigenvalue problem

```
1 1 0 0.5 -0.5
2 1 0 0.5 0.5
3 2 0 0.5 -0.5
4 2 0 0.5 0.5
5 3 0 0.5 -0.5
6 3 0 0.5 0.5
7 4 0 0.5 -0.5
8 4 0 0.5 0.5
1 x |11110000>
1 x |11001100>
1 x |111100>
1 x |11000011>
1 x |110011>
1 x |1111>
[[1.5, -0.25, -0.25, -0.25, -0.25, 0], [-0.25, 3.5, -0.25, -0.25, 0, -0.25],
 [-0.25, -0.25, 5.5, 0, -0.25, -0.25], [-0.25, -0.25, 0, 5.5, -0.25, -0.25],
 [-0.25, 0, -0.25, -0.25, 7.5, -0.25], [0, -0.25, -0.25, -0.25, -0.25, 9.5]]
[1.41677428 3.47067322 5.5        5.5        7.54943676 9.56311574]
```