Quantum Computing Lectures for Nano and Quantum Workshop

Morten Hjorth-Jensen[1,2]

[1]Department of Physics and Center for Computing in Science Education, University of Oslo, Norway
[2]Department of Physics and Astronomy and Facility for Rare Isotope Beams, Michigan State University, East Lansing, Michigan, USA

Cali, Colombia, December 4-8, 2023

Solving quantum mechanical problems

1. Simple Hamiltonian, the Lipkin model
2. Introducing the Variational Quantum Eigensolver (VQE)
3. Additional material with technicalities

Simple Hamiltonian, the Lipkin model

We will study a schematic model (the Lipkin model, see Nuclear Physics **62** (1965) 188), for the interaction among 2 and more fermions that can occupy two different energy levels.

For four fermions, the case we consider first here, each levels has degeneration $d = 4$, leading to different total spin values. The two levels have quantum numbers $\sigma = \pm 1$, with the upper level having $2\sigma = +1$ and energy $\varepsilon_1 = \varepsilon/2$. The lower level has $2\sigma = -1$ and energy $\varepsilon_2 = -\varepsilon/2$. That is, the lowest single-particle level has negative spin projection (or spin down), while the upper level has spin up. In addition, the substates of each level are characterized by the quantum numbers $p = 1, 2, 3, 4$.

Single-particle states

We define the single-particle states (for the four fermion case which we will work on here)

$$|u_{\sigma=-1,p}\rangle = a^{\dagger}_{-p}|0\rangle \qquad |u_{\sigma=1,p}\rangle = a^{\dagger}_{+p}|0\rangle.$$

The single-particle states span an orthonormal basis.

The Hamiltonian

The Hamiltonian of the system is given by

$$\begin{aligned}
\hat{H} = &\quad \hat{H}_0 + \hat{H}_1 + \hat{H}_2 \\
\hat{H}_0 = &\quad \tfrac{1}{2}\varepsilon \textstyle\sum_{\sigma,p} \sigma a^\dagger_{\sigma,p} a_{\sigma,p} \\
\hat{H}_1 = &\quad \tfrac{1}{2} V \textstyle\sum_{\sigma,p,p'} a^\dagger_{\sigma,p} a^\dagger_{\sigma,p'} a_{-\sigma,p'} a_{-\sigma,p} \\
\hat{H}_2 = &\quad \tfrac{1}{2} W \textstyle\sum_{\sigma,p,p'} a^\dagger_{\sigma,p} a^\dagger_{-\sigma,p'} a_{\sigma,p'} a_{-\sigma,p}
\end{aligned}$$

where V and W are constants. The operator H_1 can move pairs of fermions while H_2 is a spin-exchange term. The latter moves a pair of fermions from a state $(p\sigma, p' - \sigma)$ to a state $(p - \sigma, p'\sigma)$.

Rewrite in terms of quasispin operators

We are going to rewrite the above Hamiltonian in terms of so-called quasispin operators

$$\begin{aligned}
\hat{J}_+ = &\quad \textstyle\sum_p a^\dagger_{p+} a_{p-} \\
\hat{J}_- = &\quad \textstyle\sum_p a^\dagger_{p-} a_{p+} \\
\hat{J}_z = &\quad \tfrac{1}{2} \textstyle\sum_{p\sigma} \sigma a^\dagger_{p\sigma} a_{p\sigma} \\
\hat{J}^2 = &\quad J_+ J_- + J_z^2 - J_z
\end{aligned}$$

We can in turn express $\hat{H}$ in terms of the above quasispin operators and the number operator

$$\hat{N} = \sum_{p\sigma} a^\dagger_{p\sigma} a_{p\sigma}.$$

Rewriting the Hamiltonian

We can rewrite the Hamiltonian in terms of the above quasi-spin operators and the number operator We have

$$H_0 = \varepsilon J_z. \tag{1}$$

Moving over to H_1 and using the anti-commutation relations (20) through (22) we obtain

$$H_1 = \frac{1}{2} V \left(J_+^2 + J_-^2 \right). \tag{2}$$

Finally, we rewrite the last term

$$H_2 = \frac{1}{2} W \left(-N + J_+ J_- + J_- J_+ \right). \tag{3}$$

We have thus expressed the Hamiltonian in term of the quasi-spin operators.

Variational Quantum Eigensolver

One initial algorithm to estimate the eigenenergies of a quantum Hamiltonian was quantum phase estimation. In it, one encodes the eigenenergies, one binary bit at a time (up to n bits), into the complex phases of the quantum states of the Hilbert space for n qubits. It does this by applying powers of controlled unitary evolution operators to a quantum state that can be expanded in terms of the Hamiltonian's eigenvectors of interest. The eigenenergies are encoded into the complex phases in such a way that taking the inverse quantum Fourier transformation (see material on Quantum Fourier Transforms) of the states into which the eigen-energies are encoded results in a measurement probability distribution that has peaks around the bit strings that represent a binary fraction which corresponds to the eigen-energies of the quantum state acted upon by the controlled unitary operators. While quantum phase estimation (QPE) is provably efficient, non-hybrid, and non-variational, the number of qubits and length of circuits required is too great for our NISQ era quantum computers. Thus, QPE is only efficiently applicable to large, fault-tolerant quantum computers that likely won't exist in the near, but the far future.

Therefore, a different algorithm for finding the eigen-energies of a quantum Hamiltonian was put forth in 2014 called the variational quantum eigensolver, commonly referred to as VQE. The algorithm is hybrid, meaning that it requires the use of both a quantum computer and a classical computer. It is also variational, meaning that it relies, ultimately, on solving an optimization problem by varying parameters and thus is not as deterministic as QPE. The variational quantum eigensolver is based on the variational principle: The expectation value of a Hamiltonian H in a state $|\psi(\theta)\rangle$ parameterized by a set of angles θ, is always greater than or equal to the minimum eigen-energy E_0. To see this, let $|n\rangle$ be the eigenstates of H, that is

$$H|n\rangle = E_n|n\rangle. \tag{4}$$

We can then expand our state $|\psi(\theta)\rangle$ in terms of said eigenstates

$$|\psi(\theta)\rangle = \sum_n c_n |n\rangle,$$

and plug this into the expectation value to yield

$$\langle\psi(\theta)|H|\psi(\theta)\rangle = \sum_{nm} c_m^* c_n \langle m|H|n\rangle = \sum_{nm} c_m^* c_n E_n \langle m|n\rangle = \sum_{nm} \delta_{nm} c_m^* c_n E_n = \sum_n |c_n|^2 E_n \geq E_0 \sum_n |c_n|^2 = E_0,$$

which implies that we can minimize over the set of angles θ and arrive at the ground state energy E_0

$$\min_{\theta} \langle\psi(\theta)|H|\psi(\theta)\rangle = E_0.$$

Using this fact, the VQE algorithm can be broken down into the following steps

1. Prepare the variational state $|\psi(\theta)\rangle$ on a quantum computer.
2. Measure this circuit in various bases and send these measurements to a classical computer
3. The classical computer post-processes the measurement data to compute the expectation value $\langle\psi(\theta)|H|\psi(\theta)\rangle$
4. The classical computer varies the parameters θ according to a classical minimization algorithm and sends them back to the quantum computer which runs step 1 again.

This loop continues until the classical optimization algorithm terminates which results in a set of angles $\theta_{\min}$ that characterize the ground state $|\phi(\theta_{\min})\rangle$ and an estimate for the ground state energy $\langle\psi(\theta_{\min})|H|\psi(\theta_{\min})\rangle$.

Expectation values

To execute the second step of VQE, we need to understand how expectation values of operators can be estimated via quantum computers by post-processing measurements of quantum circuits in different basis. To rotate bases, one uses the basis rotator R_σ which is defined for each Pauli gate σ to be

$$R_\sigma = H, \text{if } \sigma = X, \tag{5}$$

and

$$HS^\dagger, \text{if } \sigma = Y, \tag{6}$$

and

$$I, \text{if } \sigma = Z. \tag{7}$$

We can show that these rotations allow us to measure the eigenvalues of the Pauli operators. The eigenvectors of the Pauli X gate are

$$|\pm\rangle = \frac{|0\rangle \pm |1\rangle}{\sqrt{2}},$$

with eigenvalues ± 1. Acting on the eigenstates with the rotation in eq. () gives

$$H|+\rangle = +1|0\rangle,$$

and

$$H|-\rangle = -1|1\rangle.$$

Any single-qubit state can be written as a linear combination of these eigenvectors,

$$|\psi\rangle = \alpha|+\rangle + \beta|-\rangle.$$

We then have the following expectation value for the Pauli X operator

$$X = \psi X|\psi\rangle = |\alpha|^2 - |\beta|^2.$$

However, we can only measure the qubits in the computational basis. Applying the rotation in eq. () to our state gives

$$H|\psi\rangle = \alpha|0\rangle - \beta|1\rangle.$$

This tells us that we are able to estimate $|\alpha|^2$ and $|\beta|^2$ (and hence the expectation value of the Pauli X operator) by using the rotation in eq. () and measure the resulting state in the computational basis. We can show this for the Pauli Z and Pauli Y similarly.

Note the following identity of the basis rotator

$$R_\sigma^\dagger Z R_\sigma = \sigma,$$

which follows from the fact that $HZH = X$ and $SXS^\dagger = Y$. With this, we see that the expectation value of an arbitrary Pauli-gate σ in the state $|\psi\rangle$ can be expressed as a linear combination of probabilities

$$\begin{aligned}
E_\psi(\sigma) &= \langle\psi|\sigma|\psi\rangle \\
&= \langle\psi|R_\sigma^\dagger Z R_\sigma|\psi\rangle = \langle\phi|Z|\phi\rangle \\
&= \langle\phi| \left(\sum_{x\in\{0,1\}} (-1)^x |x\rangle\langle x| \right) |\phi\rangle \\
&= \sum_{x\in\{0,1\}} (-1)^x |\langle x|\phi\rangle|^2 \\
&= \sum_{x\in\{0,1\}} (-1)^x P(|\phi\rangle \to |x\rangle), \qquad (8)
\end{aligned}$$

where $|\phi\rangle = |R_\sigma\phi\rangle$ and $P(|\phi\rangle \to |x\rangle$ is the probability that the state $|\phi\rangle$ collapses to the state $|x\rangle$ when measured. This can be extended to any arbitrary Pauli string: consider the string of Pauli operators $P = \bigotimes_{p\in Q} \sigma_p$ which acts non-trivially on the set of qubits Q which is a subset of the total set of n qubits in the system. Then

$$
\begin{aligned}
E_\psi\left(P\right) &= \langle\psi|\left(\bigotimes_{p\in Q}\sigma_p\right)|\psi\rangle \\
&= \langle\psi|\left(\bigotimes_{p\in Q}\sigma_p\right)\left(\bigotimes_{q\notin Q}I_q\right)|\psi\rangle \\
&= \langle\psi|\left(\bigotimes_{p\in Q}R^\dagger_{\sigma_p}Z_pR_{\sigma_p}\right)\left(\bigotimes_{q\notin Q}I_q\right)|\psi\rangle \\
&= \langle\psi|\left(\bigotimes_{p\in Q}R^\dagger_{\sigma_p}\right)\left(\bigotimes_{p\in Q}Z_p\right)\left(\bigotimes_{q\notin Q}I_q\right)\left(\bigotimes_{p\in Q}R_{\sigma_p}\right)|\psi\rangle \\
&= \langle\phi|\left(\bigotimes_{p\in Q}Z_p\right)\left(\bigotimes_{q\notin Q}I_q\right)|\phi\rangle \\
&= \langle\phi|\left(\bigotimes_{p\in Q}\sum_{x_p\in\{0_p,1_p\}}(-1)^{x_p}|x_p\rangle\langle x_p|\right)\left(\bigotimes_{q\notin Q}\sum_{y_q\in\{0_q,1_q\}}|y_q\rangle\langle y_q|\right)|\phi\rangle \\
&= \langle\phi|\left(\sum_{x\in\{0,1\}^n}(-1)^{\sum_{p\in Q}x_p}|x\rangle\langle x|\right)|\phi\rangle \\
&= \sum_{x\in\{0,1\}^n}(-1)^{\sum_{p\in Q}x_p}|\langle x||\phi\rangle|^2 \\
&= \sum_{x\in\{0,1\}^n}(-1)^{\sum_{p\in Q}x_p}P(|\phi\rangle\rightarrow|x\rangle), \qquad (9)
\end{aligned}
$$

where $|\phi\rangle = |\bigotimes_{p\in Q}R_{\sigma_p}\psi\rangle$. Finally, because the expectation value is linear

$$
E_\psi\left(\sum_m\lambda_mP_m\right) = \sum_m\lambda_mE_\psi(P_m), \qquad (10)
$$

one can estimate any observable that can be written as a linear combination of Pauli-string terms.

Measurement

To estimate the probability $P(|\phi\rangle\rightarrow|x\rangle)$ from the previous section, one prepares the state $|\phi\rangle$ on a quantum computer and measures it, and then repeats this process (prepare and measure) several times. The probability $P(|\phi\rangle\rightarrow|x\rangle)$ is estimated to be the number of times that one measures the bit-string x divided by the total number of measurements that one makes; that is

$$P(|\phi\rangle \to |x\rangle) \approx \sum_{m=1}^{M} \frac{x_m}{M}, \tag{11}$$

where $x_m = 1$ if the result of measurement is x and 0 if the result of measurement is not x.

By the law of large numbers the approximation approaches equality as M goes to infinity

$$P(|\phi\rangle \to |x\rangle) = \lim_{M\to\infty} \sum_{m=1}^{M} \frac{x_m}{M}. \tag{12}$$

As we obviously do not have infinite time nor infinite quantum computers (which could be run in parallel), we must truncate our number of measurement M to a finite, but sufficiently large number. More precisely, for precision ϵ, each expectation estimation subroutine within VQE requires $\mathcal{O}(1/\epsilon^2)$ samples from circuits with depth $\mathcal{O}(1)$.

Quantum computing and solving the eigenvalue problem for the Lipkin model

We turn now to a simpler variant of the Lipkin model without the W-term and a total spin of $J = 1$ only as maximum value of the spin. This corresponds to a system with $N = 2$ particles (fermions in our case). Our Hamiltonian is given by the quasispin operators (see below)

$$\hat{H} = \epsilon \hat{J}_z - \frac{1}{2} V(\hat{J}_+ \hat{J}_+ + \hat{J}_- \hat{J}_-).$$

As discussed previously, the quasispin operators act like lowering and raising angular momentum operators.

With these properties we can calculate the Hamiltonian matrix for the Lipkin model by computing the various matrix elements

$$\langle JJ_z|H|JJ'_z\rangle, \tag{13}$$

where the non-zero elements are given by

$$\begin{aligned} \langle JJ_z|H|JJ'_z\rangle &= \epsilon J_z \\ \langle JJ_z|H|JJ'_z \pm 2\rangle &= \langle JJ_z \pm 2|H|JJ'_z\rangle \\ &= -\frac{1}{2} VC, \end{aligned}$$

where C is the Clebsch-Gordan coefficients (from the raising and lowering operators) one gets when $J_\pm^2$ operates on the state $|JJ_z\rangle$. Using the above definitions we can calculate the exact solution to the Lipkin model.

With the V-interaction terms, we obtain the following Hamiltonian matrix

$$\begin{pmatrix} -\epsilon & 0 & -V \\ 0 & 0 & 0 \\ -V & 0 & \epsilon \end{pmatrix} \tag{14}$$

The following **python** code sets up the above matrix and finds the pertinent eigenvalues.

```
import numpy as np
import qiskit
from qiskit.visualization import circuit_drawer
from qiskit.quantum_info import Statevector
from matplotlib.pyplot import figure
from qiskit import QuantumRegister, QuantumCircuit, ClassicalRegister, Aer, assemble
from qiskit.providers.aer.noise import NoiseModel
import pylatexenc
from qiskit.algorithms import VQE
from qiskit.utils import QuantumInstance
from qiskit.opflow import X, Z, I, Y
from qiskit.circuit import Parameter
from qiskit.algorithms.optimizers import ADAM
from qiskit.opflow import AerPauliExpectation
from qiskit import IBMQ
import cmath
import pandas as pd
from scipy.sparse import diags
import numpy.linalg as LA
import matplotlib.pyplot as plt
from IPython.display import Image
import warnings
warnings.filterwarnings('ignore')
pi=np.pi


#function that sorts eigenvalues with its eigenvectors in accending order
def eigen(A):
    eigenValues, eigenVectors = LA.eig(A)
    idx = np.argsort(eigenValues)
    eigenValues = eigenValues[idx]
    eigenVectors = eigenVectors[:,idx]
    return (eigenValues, eigenVectors)

#one body expectation value
def one_body(E,N):
  k = N/2
  m = np.arange(-k,k+1,1) # Since the collective space is Omega+1
  return E*np.diag(m) #return a matrix where its diagonal elemens are epsilon*K_0

#two body expectation value
def two_body(V,N):
  k = N/2
  m = np.arange(-k,k+1,1)
  left =np.zeros(len(m)-2,dtype=complex)
  right = np.zeros(len(m)-2,dtype=complex)
  diag = np.zeros(len(m),dtype=complex)
  for i in range(len(left)):
    CG = cmath.sqrt(k*(k+1)-(m[i]+2)*(m[i]+1))*cmath.sqrt(k*(k+1)-m[i]*(m[i]+1)) #calculate Clebs
    left[i] = CG
```

```
    right[i] = CG
  k = [left,diag,right]
  offset = [-2,0,2]
  return -0.5*V*diags(k,offset).toarray() #return a matrix where its off digonal elements are (1/2

#full expectation value
def quasi_spin(E,V,N):
  ob = one_body(E,N)
  tb = two_body(V,N)
  H = ob+tb
  e,v = eigen(H) # find the eigenvalues of the Hamiltonian
  return e,H

#converts chi to V
def Vp(E,omega,chi):
  return (chi*E)/(omega-1)



#parameters
E = 1
chi = np.arange(0,2.1,0.1)
omega = 2

EV0 = []
EV1 = []
EV2 = []
Ham = []
for i in chi:
  v = Vp(E,omega,i)
  EigenV,H = quasi_spin(E,v,omega) #return eigenvalues and Hamiltonian
  Ham.append(H)
  EV0.append(EigenV[0])
  EV1.append(EigenV[1])
  EV2.append(EigenV[2])


matrix = pd.DataFrame(Ham[5].real)
print('Hamiltonian matrix')
matrix.head()


plt.plot(chi,EV0)
plt.plot(chi,EV1)
plt.plot(chi,EV2)
plt.xlabel('$\chi$')
plt.ylabel('Energy')
plt.title('$\Omega=2$ exact Lipkin Model')
```

Implementing the VQE method, one qubit system

We start with a simple 2×2 Hamiltonian matrix expressed in terms of Pauli X and Z matrices, as discussed in the project text.

We define a symmetric matrix $H \in \mathbb{R}^{2\times 2}$

$$H = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix},$$

We let $H = H_0 + H_I$, where

$$H_0 = \begin{bmatrix} E_1 & 0 \\ 0 & E_2 \end{bmatrix},$$

is a diagonal matrix. Similarly,

$$H_I = \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix},$$

where V_{ij} represent various interaction matrix elements. We can view H_0 as the non-interacting solution

$$H_0|0\rangle = E_1|0\rangle, \tag{15}$$

and

$$H_0|1\rangle = E_2|1\rangle, \tag{16}$$

where we have defined the orthogonal computational one-qubit basis states $|0\rangle$ and $|1\rangle$.

We rewrite H (and H_0 and H_I) via Pauli matrices

$$H_0 = \mathcal{E}I + \Omega\sigma_z, \quad \mathcal{E} = \frac{E_1 + E_2}{2}, \; \Omega = \frac{E_1 - E_2}{2},$$

and

$$H_I = c\boldsymbol{I} + \omega_z\sigma_z + \omega_x\sigma_x,$$

with $c = (V_{11} + V_{22})/2$, $\omega_z = (V_{11} - V_{22})/2$ and $\omega_x = V_{12} = V_{21}$. We let our Hamiltonian depend linearly on a strength parameter λ

$$H = H_0 + \lambda H_{\mathrm{I}},$$

with $\lambda \in [0, 1]$, where the limits $\lambda = 0$ and $\lambda = 1$ represent the non-interacting (or unperturbed) and fully interacting system, respectively. The model is an eigenvalue problem with only two available states.

Here we set the parameters $E_1 = 0$, $E_2 = 4$, $V_{11} = -V_{22} = 3$ and $V_{12} = V_{21} = 0.2$.

The non-interacting solutions represent our computational basis. Pertinent to our choice of parameters, is that at $\lambda \geq 2/3$, the lowest eigenstate is dominated by $|1\rangle$ while the upper is $|0\rangle$. At $\lambda = 1$ the $|0\rangle$ mixing of the lowest eigenvalue is 1% while for $\lambda \leq 2/3$ we have a $|0\rangle$ component of more than 90%. The character of the eigenvectors has therefore been interchanged when passing $z = 2/3$. The value of the parameter V_{12} represents the strength of the coupling between the two states.

Setting up the matrix

```
from  matplotlib import pyplot as plt
import numpy as np
dim = 2
Hamiltonian = np.zeros((dim,dim))
```

```
e0 = 0.0
e1 = 4.0
Xnondiag = 0.20
Xdiag = 3.0
Eigenvalue = np.zeros(dim)
# setting up the Hamiltonian
Hamiltonian[0,0] = Xdiag+e0
Hamiltonian[0,1] = Xnondiag
Hamiltonian[1,0] = Hamiltonian[0,1]
Hamiltonian[1,1] = e1-Xdiag
# diagonalize and obtain eigenvalues, not necessarily sorted
EigValues, EigVectors = np.linalg.eig(Hamiltonian)
permute = EigValues.argsort()
EigValues = EigValues[permute]
# print only the lowest eigenvalue
print(EigValues[0])
```

Now rewrite it in terms of the identity matrix and the Pauli matrix X and Z

```
# Now rewrite it in terms of the identity matrix and the Pauli matrix X and Z
X = np.array([[0,1],[1,0]])
Y = np.array([[0,-1j],[1j,0]])
Z = np.array([[1,0],[0,-1]])
# identity matrix
I = np.array([[1,0],[0,1]])

epsilon = (e0+e1)*0.5; omega = (e0-e1)*0.5
c = 0.0; omega_z=Xdiag; omega_x = Xnondiag
Hamiltonian = (epsilon+c)*I+(omega_z+omega)*Z+omega_x*X
EigValues, EigVectors = np.linalg.eig(Hamiltonian)
permute = EigValues.argsort()
EigValues = EigValues[permute]
# print only the lowest eigenvalue
print(EigValues[0])
```

Implementing the VQE

For a one-qubit system we can reach every point on the Bloch sphere (as discussed earlier) with a rotation about the x-axis and the y-axis.

We can express this mathematically through the following operations (see whiteboard for the drawing), giving us a new state $|\psi\rangle$

$$|\psi\rangle = R_y(\phi)R_x(\theta)|0\rangle.$$

We can produce multiple ansatzes for the new state in terms of the angles θ and ϕ. With these ansatzes we can in turn calculate the expectation value of the above Hamiltonian, now rewritten in terms of various Pauli matrices (and thereby gates), that is compute

$$\langle\psi|(c+\mathcal{E})\boldsymbol{I}+(\Omega+\omega_z)\boldsymbol{\sigma}_z+\omega_x\boldsymbol{\sigma}_x|\psi\rangle.$$

We can now set up a series of ansatzes for $|\psi\rangle$ as function of the angles θ and ϕ and find thereafter the variational minimum using for example a gradient descent method.

To do so, we need to remind ourselves about the mathematical expressions for the rotational matrices/operators.

$$R_x(\theta) = \cos\frac{\theta}{2}\boldsymbol{I} - \imath\sin\frac{\theta}{2}\boldsymbol{\sigma}_x,$$

and

$$R_y(\phi) = \cos\frac{\phi}{2}\boldsymbol{I} - \imath\sin\frac{\phi}{2}\boldsymbol{\sigma}_y.$$

```
# define the rotation matrices
# Define angles theta and phi
theta = 0.5*np.pi; phi = 0.2*np.pi
Rx = np.cos(theta*0.5)*I-1j*np.sin(theta*0.5)*X
Ry = np.cos(phi*0.5)*I-1j*np.sin(phi*0.5)*Y
#define basis states
basis0 = np.array([1,0])
basis1 = np.array([0,1])

NewBasis = Ry @ Rx @ basis0
print(NewBasis)
# Compute the expectation value
#Note hermitian conjugation
Energy = NewBasis.conj().T @ Hamiltonian @ NewBasis
print(Energy)
```

Not an impressive results. We set up now a loop over many angles θ and ϕ and compute the energies

```
# define a number of angles
n = 20
angle = np.arange(0,180,10)
n = np.size(angle)
ExpectationValues = np.zeros((n,n))
for i in range (n):
    theta = np.pi*angle[i]/180.0
    Rx = np.cos(theta*0.5)*I-1j*np.sin(theta*0.5)*X
    for j in range (n):
        phi = np.pi*angle[j]/180.0
        Ry = np.cos(phi*0.5)*I-1j*np.sin(phi*0.5)*Y
        NewBasis = Ry @ Rx @ basis0
        Energy = NewBasis.conj().T @ Hamiltonian @ NewBasis
        Edifference=abs(np.real(EigValues[0]-Energy))
        ExpectationValues[i,j]=Edifference

print(np.min(ExpectationValues))
```

Clearly, this is not the very best way of proceeding. Rather, here we would compute the gradient and thereby find the minimum as function of the angles θ and ϕ. Furthermore, in sertting up the angles, a better practice is to select random values for these.

For the lectures of April 17-21, we will add code example using gradient descent for the one- and two-qubit case. We will follow https://journals.aps.org/pra/abstract/10.1103/PhysRevA.99.032331 as a guideline to calculate gradients of the Hamiltonian.

A smarter way of doing this

The above approach means that we are setting up several matrix-matrix amd matrix-vector multiplications. Although straight forward it is not the most efficient way of doing this, in particular in case the matrices become large (and sparse). But there are some more important issues.

In a physical realization of these systems we cannot just multiply the state with the Hamiltonian. When performing a measurement we can only measure in one particular direction. For the computational basis states which we have, $|0\rangle$ and $|1\rangle$, we have to measure along the bases of the Pauli matrices and reconstruct the eigenvalues from these measurements.

From our earlier discussions we know that the Pauli Z matrix has the above basis states as eigen states through

$$\boldsymbol{\sigma}_z|0\rangle = \boldsymbol{Z}|0\rangle = +1|0\rangle,$$

and

$$\boldsymbol{\sigma}_z|1\rangle = \boldsymbol{Z}|1\rangle = -1|1\rangle,$$

with eigenvalue -1.

For the Pauli X matrix on the other hand we have

$$\boldsymbol{\sigma}_x|0\rangle = \boldsymbol{X}|0\rangle = +1|1\rangle,$$

and

$$\boldsymbol{\sigma}_x|1\rangle = \boldsymbol{X}|1\rangle = -1|0\rangle,$$

with eigenvalues 1 in both cases. The latter two equations tell us that the computational basis we have chosen, and in which we will prepare our states, is not an eigenbasis of the σ_x matrix.

We will thus try to rewrite the Pauli X matrix in terms of a Pauli Z matrix. Fortunately this can be done using the Hadamard matrix twice, that is

$$\boldsymbol{X} = \boldsymbol{\sigma}_x = \boldsymbol{HZH}.$$

The Pauli Y matrix can be written as

$$\boldsymbol{Y} = \boldsymbol{\sigma}_y = \boldsymbol{HS}^\dagger\boldsymbol{ZHS},$$

where S is the phase matrix

$$S = \begin{bmatrix} 1 & 0 \\ 0 & \imath \end{bmatrix}.$$

From here and on we will denote the Pauli matrices by X, Y and Z and we can write the expectation value of the Hamiltonian as

$$\langle\psi|(c+\mathcal{E})\boldsymbol{I} + (\Omega+\omega_z)\boldsymbol{Z} + \omega_x\boldsymbol{HZH}|\psi\rangle,$$

which we can rewrite as

$$(c+\mathcal{E})\langle\psi|\boldsymbol{I}|\psi\rangle + (\Omega+\omega_z)\langle\psi|\boldsymbol{Z}|\psi\rangle + \omega_x\langle\psi\boldsymbol{H}|\boldsymbol{Z}|\boldsymbol{H}\psi\rangle.$$

The first and second term are to easy to perform a measurement on since we we just need to compute $\langle\psi|\boldsymbol{I}|\psi\rangle$ and $\langle\psi|\boldsymbol{Z}|\psi\rangle$. For the final term we need just to add the action of the Hadamard matrix and we are done.

Additional material on the Lipkin model

Properties of the Lipkin model

We have the following quasispin operators

$$J_{\pm} = \sum_{p} a^{\dagger}_{p\pm} a_{p\mp}, \tag{17}$$

$$J_z = \frac{1}{2}\sum_{p,\sigma} \sigma a^{\dagger}_{p\sigma} a_{p\sigma}, \tag{18}$$

$$J^2 = J_+ J_- + J_z^2 - J_z, \tag{19}$$

and we want to compute the commutators

$$[J_z, J_{\pm}], \quad [J_+, J_-], \quad [J^2, J_{\pm}] \quad \text{og} \quad [J^2, J_z].$$

Let us start with the first one and inserting for J_z and $J_{\pm}$ given by the equations (18) and (17), respectively, we obtain

$$\begin{aligned}
[J_z, J_{\pm}] &= J_z J_{\pm} - J_{\pm} J_z \\
&= \left(\frac{1}{2}\sum_{p,\sigma} \sigma a^{\dagger}_{p\sigma} a_{p\sigma}\right)\left(\sum_{p'} a^{\dagger}_{p'\pm} a_{p'\mp}\right) - \left(\sum_{p'} a^{\dagger}_{p'\pm} a_{p'\mp}\right)\left(\frac{1}{2}\sum_{p,\sigma} \sigma a^{\dagger}_{p\sigma} a_{p\sigma}\right) \\
&= \frac{1}{2}\sum_{p,p',\sigma} \sigma \left(a^{\dagger}_{p\sigma} a_{p\sigma} a^{\dagger}_{p'\pm} a_{p'\mp} - a^{\dagger}_{p'\pm} a_{p'\mp} a^{\dagger}_{p\sigma} a_{p\sigma}\right).
\end{aligned}$$

Using the commutation relations for the creation and annihilation operators

$$\{a_l, a_k\} = 0, \tag{20}$$

$$\{a^{\dagger}_l, a^{\dagger}_k\} = 0, \tag{21}$$

$$\{a^{\dagger}_l, a_k\} = \delta_{lk}, \tag{22}$$

in order to move the operators in the right product to be in the same order as those in the lefthand product

$$\begin{aligned}
[J_z, J_{\pm}] &= \frac{1}{2}\sum_{p,p',\sigma} \sigma \left(a^{\dagger}_{p\sigma} a_{p\sigma} a^{\dagger}_{p'\pm} a_{p'\mp} - a^{\dagger}_{p'\pm}\left(\delta_{p'p}\delta_{\mp\sigma} - a^{\dagger}_{p\sigma} a_{p'\mp}\right) a_{p\sigma}\right) \\
&= \frac{1}{2}\sum_{p,p',\sigma} \sigma \left(a^{\dagger}_{p\sigma} a_{p\sigma} a^{\dagger}_{p'\pm} a_{p'\mp} - a^{\dagger}_{p'\pm}\delta_{p'p}\delta_{\mp\sigma} a_{p\sigma} + a^{\dagger}_{p'\pm} a^{\dagger}_{p\sigma} a_{p'\mp} a_{p\sigma}\right),
\end{aligned}$$

which results in

$$
\begin{aligned}
[J_z, J_\pm] &= \frac{1}{2}\sum_{p,p',\sigma} \sigma \left(a^\dagger_{p\sigma} a_{p\sigma} a^\dagger_{p'\pm} a_{p'\mp} - a^\dagger_{p'\pm}\delta_{pp'}\delta_{\mp\sigma} a_{p\sigma} + a^\dagger_{p\sigma} a^\dagger_{p'\pm} a_{p\sigma} a_{p'\mp} \right) \\
&= \frac{1}{2}\sum_{p,p',\sigma} \sigma \left(a^\dagger_{p\sigma} a_{p\sigma} a^\dagger_{p'\pm} a_{p'\mp} - a^\dagger_{p'\pm}\delta_{pp'}\delta_{\mp\sigma} a_{p\sigma} + a^\dagger_{p\sigma} \left(\delta_{pp'}\delta_{\pm\sigma} - a_{p\sigma} a^\dagger_{p'\pm} \right) a_{p'\mp} \right) \\
&= \frac{1}{2}\sum_{p,p',\sigma} \sigma \left(a^\dagger_{p\sigma}\delta_{pp'}\delta_{\pm\sigma} a_{p'\mp} - a^\dagger_{p'\pm}\delta_{pp'}\delta_{\mp\sigma} a_{p\sigma} \right).
\end{aligned}
$$

The last equality leads to

$$
\begin{aligned}
[J_z, J_\pm] &= \frac{1}{2}\sum_p \left((\pm 1) a^\dagger_{p\pm} a_{p\mp} - (\mp 1) a^\dagger_{p\pm} a_{p\mp} \right) = \pm\frac{1}{2}\sum_p \left(a^\dagger_{p\pm} a_{p\mp} + (\pm 1) a^\dagger_{p\pm} a_{p\mp} \right) \\
&= \pm \sum_p a^\dagger_{p\pm} a_{p\mp} = \pm J_\pm,
\end{aligned}
$$

where the last results follows from comparing with Eq. (17).

We can then continue with the next commutation relation, using Eq. (17),

$$
\begin{aligned}
[J_+, J_-] &= J_+ J_- - J_- J_+ \\
&= \left(\sum_p a^\dagger_{p'+} a_{p-} \right) \left(\sum_{p'} a^\dagger_{p'-} a_{p'+} \right) - \left(\sum_{p'} a^\dagger_{p'-} a_{p'+} \right) \left(\sum_p a^\dagger_{p+} a_{p-} \right) \\
&= \sum_{p,p'} \left(a^\dagger_{p'+} a_{p-} a^\dagger_{p'-} a_{p'+} - a^\dagger_{p'-} a_{p'+} a^\dagger_{p+} a_{p-} \right) \\
&= \sum_{p,p'} \left(a^\dagger_{p'+} a_{p-} a^\dagger_{p'-} a_{p'+} - a^\dagger_{p'-} \left(\delta_{++}\delta_{pp'} - a^\dagger_{p+} a_{p'+} \right) a_{p-} \right) \\
&= \sum_{p,p'} \left(a^\dagger_{p'+} a_{p-} a^\dagger_{p'-} a_{p'+} - a^\dagger_{p'-}\delta_{pp'} a_{p-} + a^\dagger_{p'-} a^\dagger_{p+} a_{p'+} a_{p-} \right) \\
&= \sum_{p,p'} \left(a^\dagger_{p'+} a_{p-} a^\dagger_{p'-} a_{p'+} - a^\dagger_{p'-}\delta_{pp'} a_{p-} + a^\dagger_{p+} a^\dagger_{p'-} a_{p-} a_{p'+} \right) \\
&= \sum_{p,p'} \left(a^\dagger_{p'+} a_{p-} a^\dagger_{p'-} a_{p'+} - a^\dagger_{p'-}\delta_{pp'} a_{p-} + a^\dagger_{p+} \left(\delta_{--}\delta_{pp'} - a_{p-} a^\dagger_{p'-} \right) a_{p'+} \right) \\
&= \sum_{p,p'} \left(a^\dagger_{p+}\delta_{pp'} a_{p'+} - a^\dagger_{p'-}\delta_{pp'} a_{p-} \right),
\end{aligned}
$$

which results in

$$
[J_+, J_-] = \sum_p \left(a^\dagger_{p+} a_{p+} - a^\dagger_{p-} a_{p-} \right) = 2J_z,
$$

It is straightforward to show that

$$[J^2, J_\pm] = [J_+J_- + J_z^2 - J_z, J_\pm] = [J_+J_-, J_\pm] + [J_z^2, J_\pm] - [J_z, J_\pm].$$

Using the relations

$$[AB, C] = A[B, C] + [A, C]B, \tag{23}$$
$$[A, BC] = [A, B]C + B[A, C], \tag{24}$$

we obtain

$$[J^2, J_\pm] = J_+[J_-, J_\pm] + [J_+, J_\pm]J_- + J_z[J_z, J_\pm] + [J_z, J_\pm]J_z - [J_z, J_\pm].$$

Finally, from the above it follows that

$$\begin{aligned}
[J^2, J_+] &= -2J_+J_z + J_z[J_z, J_+] + [J_z, J_+]J_z - [J_z, J_+] \\
&= -2J_+J_z + J_zJ_+ + J_+J_z - J_+ \\
&= -2J_+J_z + J_+ + J_+J_z + J_+J_z - J_+ = 0,
\end{aligned}$$

and

$$\begin{aligned}
[J^2, J_-] &= 2J_zJ_- + J_z[J_z, J_-] + [J_z, J_-]J_z - [J_z, J_-] \\
&= 2J_zJ_- - J_zJ_- - J_-J_z + J_- \\
&= J_zJ_- - (J_zJ_- + J_-) + J_- = 0.
\end{aligned}$$

Our last commutator is given by

$$\begin{aligned}
[J^2, J_z] &= [J_+J_- + J_z^2 - J_z, J_z] \\
&= [J_+J_-, J_z] + [J_z^2, J_z] - [J_z, J_z] \\
&= J_+[J_-, J_z] + [J_+, J_z]J_- \\
&= J_+J_- - J_+J_- = 0
\end{aligned}$$

Summing up we have

$$[J_z, J_\pm] = \pm J_\pm, \tag{25}$$
$$[J_+, J_-] = 2J_z, \tag{26}$$
$$[J^2, J_\pm] = 0, \tag{27}$$
$$[J^2, J_z] = 0, \tag{28}$$

which are the standard commutation relations for angular (or orbital) momentum $L_\pm$, L_z og L^2.

We can rewrite the Hamiltonian in terms of the above quasi-spin operators and the number operator

$$N = \sum_{p,\sigma} a^\dagger_{p\sigma} a_{p\sigma}. \tag{29}$$

Going through each term of the Hamiltonian and using the expressions for the quasi-spin operators we obtain

$$H_0 = \varepsilon J_z. \tag{30}$$

Moving over to H_1 and using the anti-commutation relations (20) through (22) we obtain

$$\begin{aligned} H_1 &= \frac{1}{2}V \sum_{p,p',\sigma} a^\dagger_{p\sigma} a^\dagger_{p'\sigma} a_{p'-\sigma} a_{p-\sigma} \\ &= \frac{1}{2}V \sum_{p,p',\sigma} -a^\dagger_{p\sigma} a^\dagger_{p'\sigma} a_{p-\sigma} a_{p'-\sigma} \\ &= \frac{1}{2}V \sum_{p,p',\sigma} -a^\dagger_{p\sigma} \left(\delta_{pp'}\delta_{\sigma-\sigma} - a_{p-\sigma} a^\dagger_{p'\sigma}\right) a_{p'-\sigma} \\ &= \frac{1}{2}V \sum_{p,p',\sigma} a^\dagger_{p\sigma} a_{p-\sigma} a^\dagger_{p'\sigma} a_{p'-\sigma} \end{aligned}$$

Rewriting the sum over σ we arrive at

$$\begin{aligned} H_1 &= \frac{1}{2}V \sum_{p,p'} a^\dagger_{p+} a_{p-} a^\dagger_{p'+} a_{p'-} + a^\dagger_{p-} a_{p+} a^\dagger_{p'-} a_{p'+} \\ &= \frac{1}{2}V \left[\sum_p \left(a^\dagger_{p+} a_{p-}\right) \sum_{p'} \left(a^\dagger_{p'+} a_{p'-}\right) + \sum_p \left(a^\dagger_{p-} a_{p+}\right) \sum_{p'} \left(a^\dagger_{p'-} a_{p'+}\right) \right] \\ &= \frac{1}{2}V \left[J_+ J_+ + J_- J_-\right] = \frac{1}{2}V \left[J_+^2 + J_-^2\right], \end{aligned}$$

which leads to

$$H_1 = \frac{1}{2}V \left(J_+^2 + J_-^2\right). \tag{31}$$

Finally, we rewrite the last term

$$\begin{aligned}
H_2 &= \frac{1}{2}W \sum_{p,p',\sigma} a^\dagger_{p\sigma} a^\dagger_{p'-\sigma} a_{p'\sigma} a_{p-\sigma} \\
&= \frac{1}{2}W \sum_{p,p',\sigma} -a^\dagger_{p\sigma} a^\dagger_{p'-\sigma} a_{p-\sigma} a_{p'\sigma} \\
&= \frac{1}{2}W \sum_{p,p',\sigma} -a^\dagger_{p\sigma} \left(\delta_{pp'}\delta_{-\sigma-\sigma} - a_{p-\sigma} a^\dagger_{p'-\sigma} \right) a_{p'\sigma} \\
&= \frac{1}{2}W \sum_{p,p',\sigma} -a^\dagger_{p\sigma} \delta_{pp'} a_{p'\sigma} + a^\dagger_{p\sigma} a_{p-\sigma} a^\dagger_{p'-\sigma} a_{p'\sigma} \\
&= \frac{1}{2}W \left(-\sum_{p,\sigma} a^\dagger_{p\sigma} a_{p\sigma} + \sum_{p,p',\sigma} a^\dagger_{p\sigma} a_{p-\sigma} a^\dagger_{p'-\sigma} a_{p'\sigma} \right)
\end{aligned}$$

Using the expression for the number operator we obtain

$$\begin{aligned}
\sum_{p,p',\sigma} a^\dagger_{p\sigma} a_{p-\sigma} a^\dagger_{p'-\sigma} a_{p'\sigma} &= \sum_{p,p'} a^\dagger_{p+} a_{p-} a^\dagger_{p'-} a_{p'+} + a^\dagger_{p-} a_{p+} a^\dagger_{p'+} a_{p'-} \\
&= \sum_{p} \left(a^\dagger_{p+} a_{p-} \right) \sum_{p'} \left(a^\dagger_{p'-} a_{p'+} \right) + \sum_{p} \left(a^\dagger_{p-} a_{p+} \right) \sum_{p'} \left(a^\dagger_{p'+} a_{p'-} \right) \\
&= J_+ J_- + J_- J_+,
\end{aligned}$$

resulting in

$$H_2 = \frac{1}{2}W \left(-N + J_+ J_- + J_- J_+ \right). \tag{32}$$

We have thus expressed the Hamiltonian in term of the quasi-spin operators.

Commutation relations for the Hamiltonian

The above expressions can in turn be used to show that the Hamiltonian commutes with the various quasi-spin operators. This leads to quantum numbers which are conserved. Let us first show that $[H, J^2] = 0$, which means that J is a so-called *good* quantum number and that the total spin is a conserved quantum number.

We have

$$\begin{aligned}
[H, J^2] &= [H_0 + H_1 + H_2, J^2] \\
&= [H_0, J^2] + [H_1, J^2] + [H_2, J^2] \\
&= \varepsilon[J_z, J^2] + \frac{1}{2}V[J_+^2 + J_-^2, J^2] + \frac{1}{2}W[-N + J_+ J_- + J_- J_+, J^2].
\end{aligned}$$

We have previously shown that

$$[H, J^2] = \frac{1}{2}V\left([J_+^2, J^2] + [J_-^2, J^2]\right) + \frac{1}{2}W\left(-[N, J^2] + [J_+J_-, J^2] + [J_-J_+, J^2]\right)$$

Using that $[J_\pm, J^2] = 0$, it follows that $[J_\pm^2, J^2] = 0$. We can then see that $[J_+J_-, J^2] = 0$ and $[J_-J_+, J^2] = 0$ which leads to

$$\begin{aligned}
[H, J^2] &= -\frac{1}{2}W[N, J^2] \\
&= \frac{1}{2}W\left(-[N, J_+J_-] - [N, J_z^2] + [N, J_z]\right) \\
&= \frac{1}{2}W\left(-[N, J_+]J_- - J_+[N, J_-] - [N, J_z]J_z - J_z[N, J_z] + [N, J_z]\right).
\end{aligned}$$

Combining with the number operator we have

$$\begin{aligned}
[N, J_\pm] &= NJ_\pm - J_\pm N \\
&= \left(\sum_{p,\sigma} a_{p\sigma}^\dagger a_{p\sigma}\right)\left(\sum_{p'} a_{p'\pm}^\dagger a_{p'\mp}\right) - \left(\sum_{p'} a_{p'\pm}^\dagger a_{p'\mp}\right)\left(\sum_{p,\sigma} a_{p\sigma}^\dagger a_{p\sigma}\right) \\
&= \sum_{p,p',\sigma} a_{p\sigma}^\dagger a_{p\sigma} a_{p'\pm}^\dagger a_{p'\mp} - a_{p'\pm}^\dagger a_{p'\mp} a_{p\sigma}^\dagger a_{p\sigma} \\
&= \sum_{p,p',\sigma} a_{p\sigma}^\dagger a_{p\sigma} a_{p'\pm}^\dagger a_{p'\mp} - a_{p'\pm}^\dagger \left(\delta_{\mp\sigma}\delta_{pp'} - a_{p\sigma}^\dagger a_{p'\mp}\right) a_{p\sigma} \\
&= \sum_{p,p',\sigma} a_{p\sigma}^\dagger a_{p\sigma} a_{p'\pm}^\dagger a_{p'\mp} - a_{p'\pm}^\dagger \delta_{\mp\sigma}\delta_{pp'} a_{p\sigma} + a_{p'\pm}^\dagger a_{p\sigma}^\dagger a_{p'\mp} a_{p\sigma} \\
&= \sum_{p,p',\sigma} a_{p\sigma}^\dagger a_{p\sigma} a_{p'\pm}^\dagger a_{p'\mp} + a_{p\sigma}^\dagger a_{p'\pm}^\dagger a_{p\sigma} a_{p'\mp} - \sum_p a_{p\pm}^\dagger a_{p\mp} \\
&= \sum_{p,p',\sigma} a_{p\sigma}^\dagger a_{p\sigma} a_{p'\pm}^\dagger a_{p'\mp} + a_{p\sigma}^\dagger \left(\delta_{pp'}\delta_{\pm\sigma} - a_{p\sigma} a_{p'\pm}^\dagger\right) a_{p'\mp} - \sum_p a_{p\pm}^\dagger a_{p\mp} \\
&= \sum_p a_{p\pm}^\dagger a_{p\mp} - \sum_p a_{p\pm}^\dagger a_{p\mp} = 0.
\end{aligned}$$

We obtain then

$$\begin{aligned}
[N, J_z] &= NJ_z - J_z N \\
&= \left(\sum_{p,\sigma} a_{p\sigma}^\dagger a_{p\sigma}\right)\left(\frac{1}{2}\sum_{p',\sigma} \sigma a_{p'\sigma}^\dagger a_{p'\sigma}\right) - \left(\frac{1}{2}\sum_{p',\sigma} \sigma a_{p'\sigma}^\dagger a_{p'\sigma}\right)\left(\sum_{p,\sigma} a_{p\sigma}^\dagger a_{p\sigma}\right) \\
&= \sum_{p,p',\sigma} \sigma a_{p\sigma}^\dagger a_{p\sigma} a_{p'\sigma}^\dagger a_{p'\sigma} - \sigma a_{p'\sigma}^\dagger a_{p'\sigma} a_{p\sigma}^\dagger a_{p\sigma} = 0,
\end{aligned}$$

which leads to

$$[H, J^2] = 0, \tag{33}$$

and J is a good quantum number.

Constructing the Hamiltonian matrix for $J = 2$. We start with the state (unique) where all spins point down

$$|2, -2\rangle = a^\dagger_{1-} a^\dagger_{2-} a^\dagger_{3-} a^\dagger_{4-} |0\rangle \tag{34}$$

which is a state with $J_z = -2$ and $J = 2$. (we label the states as $|J, J_z\rangle$). For $J = 2$ we have the spin projections $J_z = -2, -1, 0, 1, 2$. We can use the lowering and raising operators for spin in order to define the other states

$$J_+|J, J_z\rangle = \sqrt{J(J+1) - J_z(J_z+1)}|J, J_z + 1\rangle, \tag{35}$$

$$J_-|J, J_z\rangle = \sqrt{J(J+1) - J_z(J_z-1)}|J, J_z - 1\rangle. \tag{36}$$

We can then construct all other states with $J = 2$ using the raising operator J_+ on $|2, -2\rangle$

$$J_+|2, -2\rangle = \sqrt{2(2+1) - (-2)(-2+1)}|2, -2+1\rangle = \sqrt{6-2}|2, -1\rangle = 2|2, -1\rangle,$$

which gives

$$\begin{aligned} |2, -1\rangle &= \frac{1}{2} J_+ |2, -2\rangle \\ &= \frac{1}{2} \sum_p a^\dagger_{p+} a_{p-} a^\dagger_{1-} a^\dagger_{2-} a^\dagger_{3-} a^\dagger_{4-} |0\rangle \\ &= \frac{1}{2} \left(a^\dagger_{1+} a^\dagger_{2-} a^\dagger_{3-} a^\dagger_{4-} + a^\dagger_{1-} a^\dagger_{2+} a^\dagger_{3-} a^\dagger_{4-} + a^\dagger_{1-} a^\dagger_{2-} a^\dagger_{3+} a^\dagger_{4-} + a^\dagger_{1-} a^\dagger_{2-} a^\dagger_{3-} a^\dagger_{4+} \right) |0\rangle. \end{aligned} \tag{37}$$

We can construct all the other states in the same way. That is

$$J_+|2, -1\rangle = \sqrt{2(2+1) - (-1)(-1+1)}|2, -1+1\rangle = \sqrt{6}|2, 0\rangle,$$

which results in

$$\begin{aligned} |2, 0\rangle = \frac{1}{\sqrt{6}} \Big(& a^\dagger_{1+} a^\dagger_{2+} a^\dagger_{3-} a^\dagger_{4-} + a^\dagger_{1+} a^\dagger_{2-} a^\dagger_{3+} a^\dagger_{4-} + a^\dagger_{1+} a^\dagger_{2-} a^\dagger_{3-} a^\dagger_{4+} + a^\dagger_{1-} a^\dagger_{2+} a^\dagger_{3+} a^\dagger_{4-} + \\ & a^\dagger_{1-} a^\dagger_{2+} a^\dagger_{3-} a^\dagger_{4+} + a^\dagger_{1-} a^\dagger_{2-} a^\dagger_{3+} a^\dagger_{4+} \Big) |0\rangle \end{aligned} \tag{38}$$

The two remaining states are

$$|2, 1\rangle = \frac{1}{2} \left(a^\dagger_{1+} a^\dagger_{2+} a^\dagger_{3+} a^\dagger_{4-} + a^\dagger_{1+} a^\dagger_{2+} a^\dagger_{3-} a^\dagger_{4+} + a^\dagger_{1+} a^\dagger_{2-} a^\dagger_{3+} a^\dagger_{4+} + a^\dagger_{1-} a^\dagger_{2+} a^\dagger_{3+} a^\dagger_{4+} \right). \tag{39}$$

and

$$|2, 2\rangle = a^\dagger_{1+} a^\dagger_{2+} a^\dagger_{3+} a^\dagger_{4+} |0\rangle. \tag{40}$$

These five states can in turn be used as computational basis states in order to define the Hamiltonian matrix to be diagonalized. The matrix elements are

given by $\langle J, J_z|H|J^{'},J_z^{'}\rangle$. The Hamiltonian is hermitian and we obtain after all this labor of ours

$$H_{J=2} = \begin{bmatrix} -2\varepsilon & 0 & \sqrt{6}V & 0 & 0 \\ 0 & -\varepsilon + 3W & 0 & 3V & 0 \\ \sqrt{6}V & 0 & 4W & 0 & \sqrt{6}V \\ 0 & 3V & 0 & \varepsilon + 3W & 0 \\ 0 & 0 & \sqrt{6}V & 0 & 2\varepsilon \end{bmatrix} \tag{41}$$

We can now select a set of parameters and diagonalize the above matrix. We select $\epsilon = 2$, $V = -1/3$, $W = -1/4$ and our matrix becoes

$$H^{(1)}_{J=2} = \begin{bmatrix} -4 & 0 & -\sqrt{6}/3 & 0 & 0 \\ 0 & -2 - 3/4 & 0 & -1 & 0 \\ -\sqrt{6}/3 & 0 & -1 & 0 & -\sqrt{6}/3 \\ 0 & -1 & 0 & 2 + -3/4 & 0 \\ 0 & 0 & -\sqrt{6}/3 & 0 & 4 \end{bmatrix},$$

which gives the eigenvalue

$$D = \begin{bmatrix} -4.21288 & 0 & 0 & 0 & 0 \\ 0 & -2.98607 & 0 & 0 & 0 \\ 0 & 0 & -0.91914 & 0 & 0 \\ 0 & 0 & 0 & 1.48607 & 0 \\ 0 & 0 & 0 & 0 & 4.13201 \end{bmatrix}.$$

The lowest state has an admixture of basis states given by

$$|\psi_0\rangle = 0.96735|2, -2\rangle + 0.25221|2, 0\rangle + 0.02507|2, 2\rangle,$$

with energy $E_0 = -4.21288$.

We can now change the parameters to $\varepsilon = 2$, $V = -4/3$, $W = -1$. Our matrix reads then

$$H^{(2)}_{J=2} = \begin{bmatrix} -4 & 0 & -4\sqrt{6}/3 & 0 & 0 \\ 0 & -5 & 0 & -4 & 0 \\ -4\sqrt{6}/3 & 0 & -4 & 0 & -4\sqrt{6}/3 \\ 0 & -4 & 0 & -1 & 0 \\ 0 & 0 & -4\sqrt{6}/3 & 0 & 4 \end{bmatrix},$$

with the following eigenvalues

$$D = \begin{bmatrix} -7.75122 & 0 & 0 & 0 & 0 \\ 0 & -7.47214 & 0 & 0 & 0 \\ 0 & 0 & -1.55581 & 0 & 0 \\ 0 & 0 & 0 & 1.47214 & 0 \\ 0 & 0 & 0 & 0 & 5.30704 \end{bmatrix}.$$

The new ground state (lowest state) has the following admixture of computational basis states

$$|\psi_0\rangle = 0.64268|2,-2\rangle + 0.73816|2,0\rangle + 0.20515|2,2\rangle,$$

with energy $E_0 = -7.75122$.

For the first set of parameters, the likelihood for observing the system in the computational basis state $|2,-2\rangle$ is rather large. This is expected since the interaction matrix elements are smaller than the single-particle energies. For the second case, with larger matrix elements, we see a much stronger mixing of the other states, again as expected due to the ratio of the interaction matrix elements and the single-particle energies.

Quantum Circuit, rewriting the Lipkin model in terms of Pauli matrices

To solve the Lipkin model on a quantum computer we have to solve Schrodinger's equation. To achieve this, we will use the Variational Quantum Eigensolver (VQE) discussed above

Before we proceed however, we need to rewrite the quasispin operators in terms of Pauli spin matrices/operators.

We take the liberty here of reminding you of some of the derivations done previously. We defined the number operator as

$$N = \sum_{n\sigma} a^\dagger_{n\sigma} a_{n\sigma},$$

which commutes with the Lipkin Hamiltonian. This can be seen by examining the Lipkin model Hamiltonian and noticing that the one-body part simply counts particles while the two-body term moves particles in pairs. Thus, the Hamiltonian conserves particle number. To find more symmetries we rewrote the Lipkin Hamiltonian in terms of $SU(2)$ quasispin operators

$$H = \epsilon J_z + \frac{1}{2} V (J_+^2 + J_-^2), \tag{42}$$

via the mappings

$$J_z = \sum_n j_z^{(n)},$$

and

$$J_\pm = \sum_n j_\pm^{(n)},$$

where we have the onebody operators

$$j_z^{(n)} = \frac{1}{2} \sum_\sigma \sigma a^\dagger_{n\sigma} a_{n\sigma},$$

and

$$j_{\pm}^{(n)} = a_{n\pm}^{\dagger} a_{n\mp}.$$

These operators obey the $SU(2)$ commutation relations

$$[J_+, J_-] = 2J_z,$$

and

$$[J_z, J_\pm] = \pm J_\pm.$$

Here the ladder operators are defined as $J_\pm = J_x \pm iJ_y$. With this rewriting, we can see that the total spin operator J^2, which is defined as

$$J^2 = J_x^2 + J_y^2 + J_z^2 = \frac{1}{2}\{J_+, J_-\} + J_z^2,$$

commutes with the Hamiltonian since the Hamiltonian. We note also that the rotation operator

$$R = e^{i\phi J_z},$$

commutes with the Hamiltonian, which can be explained as follows. Writing J_z as

$$J_z = \frac{1}{2}(N_+ - N_-),$$

where $N_\pm = \sum_{n\pm} a_{n\pm}^{\dagger} a_{n\pm}$, allows us to see that it measures half the difference between the number of particles in the upper and lower levels. Thus, the possible eigenvalues r of the signature operator are

$$r = +1, j_z = 2n \tag{43}$$

$$r = +i, j_z = 2n + \frac{1}{2} \tag{44}$$

$$r = -1, j_z = 2n + 1 \tag{45}$$

$$r = -i, j_z = 2n + \frac{3}{2} \tag{46}$$

$$\tag{47}$$

for $n \in \mathbb{Z}$. Note that r is real or imaginary if the number of particles N is even or odd, respectively. Since, as discussed above, the Lipkin Hamiltonian conserves N, r cannot jump between being real and imaginary. Additionally, because particles must be moved in pairs, and J_z measures half the difference between particles in the upper and lower levels, j_z can only change by as

$$j_z \to \frac{1}{2}[(N_+ \pm 2n) - (N_- \mp 2n)]$$

or $j_z \to J_z \pm 2n$.

To solve the Lipkin model with a quantum computer, the first step is to map the system to a set of qubits. We will restrict ourselves here to the half-filled case where the number of particles N equals the degeneracy of the states Ω.

One could assign each possible state (n, σ) a qubit such that the qubit being in the state $|1\rangle$ or $|0\rangle$ would imply that the state (n, σ) is occupied or unoccupied, respectively. This mapping scheme (which we will call occupation mapping) requires 2Ω qubits.

The Hamiltonian takes the form

$$H = \epsilon J_z + \frac{1}{2} V (J_+^2 + J_-^2). \tag{48}$$

Plugging the mapping from the total J operators to the individual one-body j operators yields

$$H = \epsilon \sum_n j_z^{(n)} + \frac{1}{2} V \left[\left(\sum_n j_+^{(n)} \right)^2 + \left(\sum_n j_-^{(n)} \right)^2 \right] \tag{49}$$

$$= \epsilon \sum_n j_z^{(n)} + \frac{1}{2} V \sum_{n,m} \left(j_+^{(n)} j_+^{(m)} + j_-^{(n)} j_-^{(m)} \right) \tag{50}$$

$$= \epsilon \sum_n j_z^{(n)} + 2V \sum_{n<m} \left(j_x^{(n)} j_x^{(m)} - j_y^{(n)} j_y^{(m)} \right), \tag{51}$$

where we have used the definitions

$$j_\pm^{(n)} = j_x^{(n)} \pm i j_y^{(n)}.$$

To convert to Pauli matrices, we make the transformations

$$j_x^{(n)} \to X_n/2,$$

and

$$j_y^{(n)} \to Y_n/2,$$

and finally

$$j_z^{(n)} \to Z_n/2,$$

which preserve the above $SU(2)$ commutation relations. The factor of $1/2$ is due to the eigenvalues of the Pauli matrices being ± 1 while we are dealing with spin $1/2$ particles.

This transforms our Hamiltonian into

$$H = \frac{1}{2} \epsilon \sum_{k=1}^{n} Z_k + \frac{1}{2} V \sum_{n \neq j=1}^{N} (X_k X_j - Y_k Y_j).$$

With this form, we can clearly see that the first (one-body) term in the Hamiltonian returns the energy $-\epsilon/2$ or $+\epsilon/2$ if the qubit representing the particle of a doublet is in the ground ($|1\rangle$) or excited ($|0\rangle$) state, respectively.

The action of the second (two-body) term in the Hamiltonian can be determined by noting that

$$\frac{1}{2}(XX - YY)|00\rangle = |11\rangle, \tag{52}$$

$$\frac{1}{2}(XX - YY)|01\rangle = 0, \tag{53}$$

$$\frac{1}{2}(XX - YY)|10\rangle = 0, \tag{54}$$

$$\frac{1}{2}(XX - YY)|11\rangle = |00\rangle. \tag{55}$$

That is, the two-body term moves a pair of particles between the ground states $|00\rangle$ and the excited states $|11\rangle$ of their respective doublets.

To construct an efficient ansatz, we must determine the subspace within which the Hamiltonian lives. To begin, note that particles are only ever moved between energy levels in pairs. Further, note that the Hamiltonian's coefficients (ϵ and V) are state independent (do not depend on the indices n or m) as the states labeled by these indices are degenerate and thus have the same energy level. Thus, the Hamiltonian treats all states with the same number of excited particles (Hamming weight of the state) as the same. Therefore, the following ansatz forms exactly cover the subspace within which the N-degenerate Hamiltonian explores:

$$|\psi_{\text{even}}\rangle = \sum_{k=0}^{\lfloor n/2 \rfloor} c_{2k}|D_{2k}^n\rangle, \tag{56}$$

$$|\psi_{\text{odd}}\rangle = \sum_{k=0}^{\lfloor n/2 \rfloor} c_{2k+1}|D_{2k+1}^n\rangle. \tag{57}$$

Here $|D_k^n\rangle$ represents a Dicke state which is defined as equal superposition of all n-qubit states with Hamming weight k. That is

$$|D_k^n\rangle = \frac{1}{\sqrt{\binom{n}{k}}} \sum_{x \in h_k^n} |x\rangle, \tag{58}$$

where $h_k^n = \{|x\rangle \mid \mathrm{l}(x) = n,\ \mathrm{wt}(x) = k\}$. There are two ways we can think of to prepare such ansatz: The first is to prepare them exactly as it is known how to deterministically prepare Dicke states with linear depth. The reference provides an algorithm for preparing a set of gates U_k^n that prepares a Dicke state from a product state of Hamming weight k; that is

$$U_k^n|1\rangle^{\otimes k}|0\rangle^{\otimes n-k} = |D_k^n\rangle. \tag{59}$$

It then describes how to one can create an arbitrary superposition of Dicke states, which we modify here to restrict ourselves to a Hamming weight of constant parity. The circuit to construct such a state (for the $k = 6$ case, as an example) is discussed next week.