Name: **Huan Q. Bui**
Course: **8.321 - Quantum Theory I**
Problem set: **#3**

1.

$$A = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix} \qquad B = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 0 & -1 \\ 1 & -1 & 2 \end{pmatrix}.$$

(a) To show that AB commute, we simply compute their commutator:

$$[A,B] = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}\begin{pmatrix} 2 & 1 & 1 \\ 1 & 0 & -1 \\ 1 & -1 & 2 \end{pmatrix} - \begin{pmatrix} 2 & 1 & 1 \\ 1 & 0 & -1 \\ 1 & -1 & 2 \end{pmatrix}\begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 3 & 0 & 3 \\ 0 & 0 & 0 \\ 3 & 0 & 3 \end{pmatrix} - \begin{pmatrix} 3 & 0 & 3 \\ 0 & 0 & 0 \\ 3 & 0 & 3 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

So, A and B commute.

(b) Notice that $\text{rank}(A) = 1$. So A must have eigenvalue of zero with multiplicity of two. The other eigenvalue is 2 by inspection, where the corresponding eigenvector is $(1,0,1)^\top$. The other two 0-eigenvectors must span the subspace orthogonal to $(1,0,1)^\top$. We may choose $(0,1,0)^\top$ and $(-1,0,1)^\top$.

To find the eigenvalues of B we may use the traditional method of characteristic polynomials.

$$0 = \det(B - \lambda\mathbb{I}) = -6 - \lambda + 4\lambda^2 - \lambda^3 \implies 0 = (\lambda-3)(\lambda-2)(\lambda+1).$$

The corresponding eigenvectors are

$$\begin{pmatrix} 2 & 1 & 1 \\ 1 & 0 & -1 \\ 1 & -1 & 2 \end{pmatrix}\vec{x}_1 = 3\vec{x}_1 \implies \vec{x}_1 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 1 & 1 \\ 1 & 0 & -1 \\ 1 & -1 & 2 \end{pmatrix}\vec{x}_2 = 2\vec{x}_2 \implies \vec{x}_2 = \begin{pmatrix} -1 \\ -1 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 1 & 1 \\ 1 & 0 & -1 \\ 1 & -1 & 2 \end{pmatrix}\vec{x}_3 = -1\vec{x}_3 \implies \vec{x}_3 = \begin{pmatrix} -1 \\ 2 \\ 1 \end{pmatrix}$$

(c) It is clear that $(1,0,1)^\top$ is a simultaneous eigenvector of A and B. Also notice that the eigenvectors $\vec{x}_2$ and $\vec{x}_3$ of B are orthogonal to each other and to $(1,0,1)^\top$. This means $\vec{x}_2$ and $\vec{x}_3$ span the subspace associated with the eigenvalue zero for A. Thus, $\vec{x}_2, \vec{x}_3$ are eigenvectors of A and it suffices to normalize $\vec{x}_1, \vec{x}_2, \vec{x}_3$ to form a unitary matrix:

$$\boxed{U = \begin{pmatrix} 1/\sqrt{2} & -1/\sqrt{3} & -1/\sqrt{6} \\ 0 & -1/\sqrt{3} & 2/\sqrt{6} \\ 1/\sqrt{2} & 1/\sqrt{3} & 1/\sqrt{6} \end{pmatrix}}$$

Simultaneous diagonalization of A and B:

$$U^\dagger A U = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$U^\dagger B U = \begin{pmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

as desired.

2. N spin-1/2 particles in

$$\mathcal{H} = \mathcal{H}_2^{(1)} \otimes \mathcal{H}_2^{(2)} \otimes \cdots \otimes \mathcal{H}_2^{(n)}.$$

where each $\mathcal{H}_2^{(i)}$ is two-dimensional.

(a) The dimension of $\mathcal{H}$ is 2^n.

(b) $S_z = S_z^{(1)} + S_z^{(2)} + \cdots + S_z^{(n)}$. There are $\binom{n}{i}$ product (eigen)states with i particles in $|\uparrow\rangle$ and $(n-i)$ particles in $|\downarrow\rangle$. For the product state with i particles in $|\uparrow\rangle$, the corresponding eigenvalue is

$$\lambda = \frac{\hbar}{2} i - \frac{\hbar}{2}(n-i) = \frac{\hbar}{2}\left(2i - n\right), \qquad i = 0, 1, 2, \ldots, n$$

So, the spectrum of S_z is

$$\sigma(S_z) = \left\{ \frac{n\hbar}{2}, \frac{(n-2)\hbar}{2}, \ldots, \frac{-(n-2)\hbar}{2}, \frac{-n\hbar}{2} \right\}$$

There are $n+1$ distinct eigenvalues. The multiplicity of each λ_i is $\binom{n}{i}$ where λ_i is the eigenvalue associated with the product state with i spins in $|\uparrow\rangle$.

As a sanity check, the sum of the multiplicities must be 2^n. This is the case here due to a well-known combinatorial relation:

$$\sum_{i=0}^{n} \binom{n}{i} = (1+1)^n = 2^n.$$

(c) $I = \mathbf{S}^{(1)} \cdot \mathbf{S}^{(2)} + \mathbf{S}^{(2)} \cdot \mathbf{S}^{(3)} + \cdots + \mathbf{S}^{(N-1)} \cdot \mathbf{S}^{(N)} + \mathbf{S}^{(N)} \cdot \mathbf{S}^{(1)}$. We claim that $[I, S_z] = 0$ and shall prove this by induction. Consider the base case where $N = 2$. We may prove it directly by calculating the Kronecker product of the Pauli matrices (working in the z-basis).

$$I = \mathbf{S}^{(1)} \cdot \mathbf{S}^{(2)} + \mathbf{S}^{(2)} \cdot \mathbf{S}^{(1)} = 2\left[S_x^{(1)} \otimes S_x^{(2)} + S_y^{(1)} \otimes S_y^{(2)} + S_z^{(1)} \otimes S_z^{(2)} \right]$$
$$S_z = S_z^{(1)} \otimes \mathbb{I} + \mathbb{I} \otimes S_z^{(2)}.$$

In Mathematica:

```
In[2]:= X = PauliMatrix[1];

In[3]:= Y = PauliMatrix[2];

In[4]:= Z = PauliMatrix[3];

In[6]:= Id = IdentityMatrix[2];

In[8]:= II =
2*(KroneckerProduct[X, X] + KroneckerProduct[Y, Y] +
KroneckerProduct[Z, Z]);

In[9]:= SZ = KroneckerProduct[Z, Id] + KroneckerProduct[Id, Z];

In[12]:= Commutator = II . SZ - SZ . II;

In[13]:= Commutator

Out[13]= {{0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}}
```

Thus

$$[I^{(2)}, Sz] = 0.$$

Now let us assume that $[I, S_z]$ is true up to N. We will show that $[I, S_z]$ also holds for $N+1$. This is a straightforward computation. Let us call $I = I_N + I'$ where

$$I_N = \mathbf{S}^{(1)} \cdot \mathbf{S}^{(2)} + \mathbf{S}^{(2)} \cdot \mathbf{S}^{(3)} + \cdots + \mathbf{S}^{(N-1)} \cdot \mathbf{S}^{(N)} + \mathbf{S}^{(N)} \cdot \mathbf{S}^{(1)}$$

and

$$I' = \mathbf{S}^{(N)} \cdot \mathbf{S}^{(N+1)} + \mathbf{S}^{(N+1)} \cdot \mathbf{S}^{(1)} - \mathbf{S}^{(N)} \cdot \mathbf{S}^{(1)}$$

Moreover, let us write

$$S_z = S_z^{(N+1)} + \sum_{i=1}^{N} S_z^{(i)}.$$

Since I_N commute with both S_z^{N+1} (by the fact that S_z^{N+1} does not act on the spins $i = 1, \ldots, N$) and $\sum_i^N S_z^{(i)}$ (by inductive hypothesis), we have

$$\begin{aligned}
[I, S_z] &= \left[I_N + I', S_z^{(N+1)} + \sum_{i=1}^{N} S_z^{(i)} \right] \\
&= \left[I', S_z^{(N+1)} \right] + \left[I', \sum_{i=1}^{N} S_z^{(i)} \right] \\
&= \left[\mathbf{S}^{(N)} \cdot \mathbf{S}^{(N+1)} + \mathbf{S}^{(N+1)} \cdot \mathbf{S}^{(1)}, S_z^{(N+1)} \right] + \left[\mathbf{S}^{(N)} \cdot \mathbf{S}^{(N+1)} + \mathbf{S}^{(N+1)} \cdot \mathbf{S}^{(1)}, \sum_{i=1}^{N} S_z^{(i)} \right]
\end{aligned}$$

where we have used the following facts:

$$\begin{aligned}
\left[\mathbf{S}^{(N)} \cdot \mathbf{S}^{(1)}, S_z^{(N+1)} \right] &= 0 \\
\left[\mathbf{S}^{(N)} \cdot \mathbf{S}^{(1)}, \sum_{i=1}^{N} S_z^{(i)} \right] &= \left[\mathbf{S}^{(N)} \cdot \mathbf{S}^{(1)}, S_z^{(1)} + S_z^{(N)} \right] = 0
\end{aligned}$$

where the second comes from the $N = 2$ result. We may simplify $[I, S_z]$ even further:

$$\begin{aligned}
[I, S_z] &= \left[\mathbf{S}^{(N)} \cdot \mathbf{S}^{(N+1)} + \mathbf{S}^{(N+1)} \cdot \mathbf{S}^{(1)}, S_z^{(N+1)} \right] + \left[\mathbf{S}^{(N)} \cdot \mathbf{S}^{(N+1)} + \mathbf{S}^{(N+1)} \cdot \mathbf{S}^{(1)}, S_z^{(1)} + S_z^{(N)} \right] \\
&= \left[\mathbf{S}^{(N)} \cdot \mathbf{S}^{(N+1)} + \mathbf{S}^{(N+1)} \cdot \mathbf{S}^{(1)}, S_z^{(1)} + S_z^{(N)} + S_z^{(N+1)} \right] \\
&= \left[\mathbf{S}^{(N)} \cdot \mathbf{S}^{(N+1)}, S_z^{(1)} + S_z^{(N)} + S_z^{(N+1)} \right] + \left[\mathbf{S}^{(N+1)} \cdot \mathbf{S}^{(1)}, S_z^{(1)} + S_z^{(N)} + S_z^{(N+1)} \right] \\
&= \left[\mathbf{S}^{(N)} \cdot \mathbf{S}^{(N+1)}, S_z^{(N)} + S_z^{(N+1)} \right] + \left[\mathbf{S}^{(N+1)} \cdot \mathbf{S}^{(1)}, S_z^{(1)} + S_z^{(N+1)} \right] \\
&= 0 + 0 \\
&= 0
\end{aligned}$$

in view of the $N = 2$ base case. We thus conclude that I and S_z are compatible observables.

(d) Spectrum and degeneracies of I for $N = 2, 3, 4$.

- $N = 2$ (dropping the factor of $\hbar^2/4$ for clarity):

$$\boxed{\sigma(I_2) = \{-6, \underbrace{2}_{\text{deg.}=3}\}}$$

- $N = 3$ (dropping the factor of $\hbar^2/4$ for clarity):

$$\boxed{\sigma(I_3) = \{\underbrace{-3}_{\text{deg.=4}},\underbrace{3}_{\text{deg.=4}}\}}$$

- $N = 4$ (dropping the factor of $\hbar^2/4$ for clarity):

$$\boxed{\sigma(I_4) = \{-8,\underbrace{-4}_{\text{deg.=3}},\underbrace{4}_{\text{deg.=5}},\underbrace{0}_{\text{deg.=7}}\}}$$

For this problem I have used a brute force via a simple routine in MATLAB which allows me compute the spectrum for large N's. Below is the code.

```
N    = 10;
state0 = zeros(2^N,1);
eigv = 0;

Sz = [1 0 ; 0 -1];
Sx = [0 1 ; 1 0];
Sy = [0 -complex(0,1); complex(0,1) 0];
Id = [1 0 ; 0 1];

% ZZ, YY, XX
cell_ZZ = cell(N,1);
cell_YY = cell(N,1);
cell_XX = cell(N,1);
termZ = zeros(2,2);
termY = zeros(2,2);
termX = zeros(2,2);
operatorsZ = cell(N,1);
operatorsY = cell(N,1);
operatorsX = cell(N,1);

for n = 0:N-2
operatorsZ = horzcat( horzcat( repmat({Id},1,n)  ,horzcat({Sz}, {Sz})), repmat({Id}, 1 , N-2-n));
operatorsY = horzcat( horzcat( repmat({Id},1,n)  ,horzcat({Sy}, {Sy})), repmat({Id}, 1 , N-2-n));
operatorsX = horzcat( horzcat( repmat({Id},1,n)  ,horzcat({Sx}, {Sx})), repmat({Id}, 1 , N-2-n));
termZ = operatorsZ{1};
termY = operatorsY{1};
termX = operatorsX{1};
for o = 2:N
termZ = kron(termZ, operatorsZ{o});
termY = kron(termY, operatorsY{o});
termX = kron(termX, operatorsX{o});
end
cell_ZZ{n+1} = termZ;
cell_YY{n+1} = termY;
cell_XX{n+1} = termX;
end


% deals with the periodic term
operatorsZ = horzcat(horzcat( {Sz}, repmat({Id}, 1, N-2) ), {Sz} );
operatorsY = horzcat(horzcat( {Sy}, repmat({Id}, 1, N-2) ), {Sy} );
operatorsX = horzcat(horzcat( {Sx}, repmat({Id}, 1, N-2) ), {Sy} );
termZ = operatorsZ{1};
termY = operatorsY{1};
termX = operatorsX{1};
for o = 2:N
termZ = kron(termZ, operatorsZ{o});
termY = kron(termY, operatorsY{o});
termX = kron(termX, operatorsX{o});
end
cell_ZZ{N} = termZ;
cell_YY{N} = termY;
cell_XX{N} = termX;

% generates Hamiltonian
Hamiltonian = zeros(2^N,2^N);
for i = 1:N
Hamiltonian = Hamiltonian + cell_ZZ{i} + cell_XX{i} + cell_YY{i};
end
```

```
[state0, eigv] = eig(Hamiltonian);

disp(transpose(diag(eigv)));
disp(state0);
```

(e) By running the MATLAB program for a range of N's, we find that

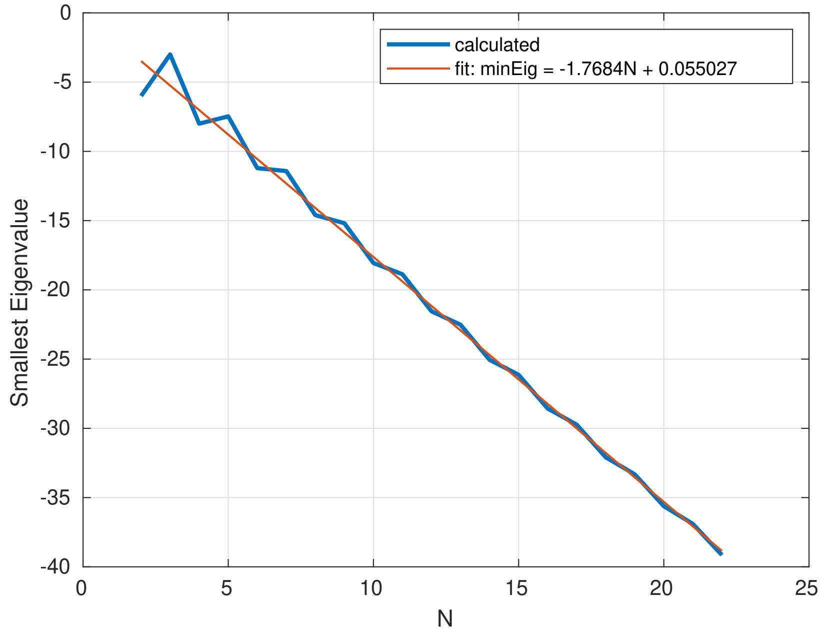
$$\boxed{\lambda_{\text{max}}^{(N)} = N} \qquad \text{restoring } \hbar^2/4 \implies \lambda_{\text{max}}^{(N)} = N\hbar^2/4$$

with degeneracy $N+1$. Since the degenaracy grows linearly in N, we have multiple choices for an eigenvector associated with each of these eigenvalues. However, using MATLAB we may find that the eigenvectors

$$\boxed{\left|\psi\right\rangle = (0,0,\dots,1)^\top \qquad \text{and} \qquad \left|\phi\right\rangle = (1,0,\dots,0)^\top}$$

are always associated with the largest positive eigenvalue.

(f) The largest N I could compute in a reasonable amount of time (**48 seconds**) is $\boxed{N = 22}$ We see that



$\lambda\text{min}^{(N)}$ also decreases without bounds and appears to scale linearly in N for small N, similar to $\lambda_{\text{max}}^{(N)}$. I did a linear fit to the data and found that

$$\lambda_{\text{min}}^{(N)} \approx \frac{\hbar^2}{4}\left(-1.7684N + 0.055027\right).$$

A few associated eigenvectors can also be found using the previous MATLAB routine, but I can't seem to find anything special about these eigenvectors in relation to N.

MATLAB code (optimized for the minimum eigenvalue problem):

```
clear
%%%%%%%%%%%%%%%%%%%%%
% clock starts
```

```
tic
% clock starts
%%%%%%%%%%%%%%%%%%%

N = 22;
parfor j=1:N-1
data(j) = MinEig(j+1);
end
plot(2:1:N, data, ’LineWidth’,2)
ylabel(’Smallest Eigenvalue’)
xlabel(’N’)
grid on

% linear fit
p = polyfit(2:1:N, data, 1);
fit = polyval(p,2:1:N);
hold on
plot(2:1:N,fit, ’LineWidth’, 1)
hold off

% display fit eqn in legend
a = p(1);
b = p(2);
legend(’calculated’, [’fit: minEig = ’ num2str(a) ’N + ’ num2str(b)])



%%%%%%%%%%%%%%%%%%%
% clock ends
Duration = seconds(round(toc));
Duration.Format = ’hh:mm:ss’;
disp([’Time taken : ’ char(Duration)]);
% disp([’Time in sec: ’ num2str(toc)]);
disp(’ ’)
% clock ends
%%%%%%%%%%%%%%%%%%%

% returns the smallest eigenvalue
function minEig = MinEig(N)

Sz = sparse([1 0 ; 0 -1]);
Sx = sparse([0 1 ; 1 0]);
Sy = sparse([0 -complex(0,1); complex(0,1) 0]);
Id = sparse([1 0 ; 0 1]);

% ZZ, YY, XX
cell_ZZ = cell(N,1);
cell_YY = cell(N,1);
cell_XX = cell(N,1);
termZ = zeros(2,2);
termY = zeros(2,2);
termX = zeros(2,2);
operatorsZ = cell(N,1);
operatorsY = cell(N,1);
operatorsX = cell(N,1);

parfor n = 0:N-2
operatorsZ = horzcat( horzcat( repmat({Id},1,n)  ,horzcat({Sz}, {Sz})), repmat({Id}, 1 , N-2-n));
operatorsY = horzcat( horzcat( repmat({Id},1,n)  ,horzcat({Sy}, {Sy})), repmat({Id}, 1 , N-2-n));
operatorsX = horzcat( horzcat( repmat({Id},1,n)  ,horzcat({Sx}, {Sx})), repmat({Id}, 1 , N-2-n));
termZ = operatorsZ{1};
termY = operatorsY{1};
termX = operatorsX{1};
for o = 2:N
termZ = sparse(kron(termZ, operatorsZ{o}));
termY = sparse(kron(termY, operatorsY{o}));
termX = sparse(kron(termX, operatorsX{o}));
end
cell_ZZ{n+1} = termZ;
cell_YY{n+1} = termY;
cell_XX{n+1} = termX;
end

% periodic term
operatorsZ = horzcat(horzcat( {Sz}, repmat({Id}, 1, N-2) ), {Sz} );
operatorsY = horzcat(horzcat( {Sy}, repmat({Id}, 1, N-2) ), {Sy} );
operatorsX = horzcat(horzcat( {Sx}, repmat({Id}, 1, N-2) ), {Sx} );
termZ = operatorsZ{1};
termY = operatorsY{1};
termX = operatorsX{1};
for o = 2:N
```

```
termZ = sparse(kron(termZ, operatorsZ{o}));
termY = sparse(kron(termY, operatorsY{o}));
termX = sparse(kron(termX, operatorsX{o}));
end
cell_ZZ{N} = termZ;
cell_YY{N} = termY;
cell_XX{N} = termX;

% generates Hamiltonian
Hamiltonian = sparse(2^N,2^N);
parfor i = 1:N
Hamiltonian = Hamiltonian + cell_ZZ{i} + cell_XX{i} + cell_YY{i};
end
% exact diagonalization
eigv = eigs(Hamiltonian,1,'smallestreal');

% returns smallest eigenvalue
minEig = eigv;

end
```

(g) We have

$$\mathcal{H} = bxS_z - a(1-x)I.$$

With $b = a\hbar$, we may set $a = 1$ and $\hbar = 1$, so that

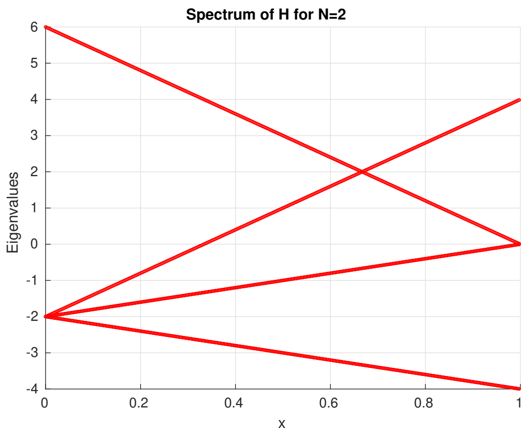
$$\mathcal{H} = \frac{x}{2}S_z - \frac{(1-x)}{4}I$$

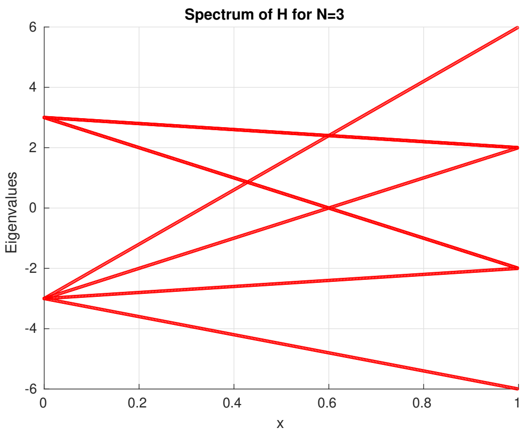
where we have nondimensionalized $S_z \propto \hbar/2$ and $I \propto \hbar^2/4$. We can also multiply $\mathcal{H}$ by 4 to remove all denominators:

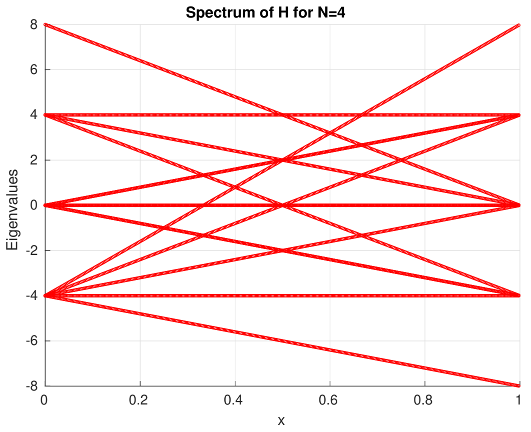
$$\mathcal{H} = 2xS_z - (1-x)I.$$

The figures generated using the MATLAB code below show good agreement with what we found in the previous parts (setting $x = 0$ and flipping the signs of the spectrum gives the results of Part (d)). When $x = 1$, the spectra of H are twice what we found in Part (b), which can be explained by the extra leading factor of 2 on S_z in the Hamiltonian, but are otherwise consistent. For example, at $x = 1$ and $N = 3$ we find $4 = 3 + 1$ distinct eigenvalues. To count degeneracies requires looking at the spectra numerically, since some of the splittings in the figures overlap. But in any case, the degeneracies are also consistent with what we found in Part (b).

```
clear crc
clear all

N = 4;
res = 400;
x = 0:1/res:1-1/res;
figure(1)
for j=0:1:res-1
eigv = Bfield(N,j/res);
hold on
X = (j/res)*ones(2^N,1);
plot(X,eigv,'Marker','o', 'MarkerSize', 2, 'Color', 'r', 'LineStyle', 'None')
hold off
end
hold off
grid on

xlabel('x')
ylabel('Eigenvalues')
title(['Spectrum of H for N=' num2str(N)])




function eigv = Bfield(N,x)
Sz = [1 0 ; 0 -1];
Sx = [0 1 ; 1 0];
Sy = [0 -complex(0,1); complex(0,1) 0];
```







```
Id = [1 0 ; 0 1];

% ZZ, YY, XX
cell_ZZ = cell(N,1);
cell_YY = cell(N,1);
cell_XX = cell(N,1);
termZ = zeros(2,2);
termY = zeros(2,2);
termX = zeros(2,2);
operatorsZ = cell(N,1);
operatorsY = cell(N,1);
operatorsX = cell(N,1);

for n = 0:N-2
operatorsZ = horzcat( horzcat( repmat({Id},1,n)  ,horzcat({Sz}, {Sz})), repmat({Id}, 1 , N-2-n));
operatorsY = horzcat( horzcat( repmat({Id},1,n)  ,horzcat({Sy}, {Sy})), repmat({Id}, 1 , N-2-n));
operatorsX = horzcat( horzcat( repmat({Id},1,n)  ,horzcat({Sx}, {Sx})), repmat({Id}, 1 , N-2-n));
termZ = operatorsZ{1};
termY = operatorsY{1};
termX = operatorsX{1};
for o = 2:N
termZ = kron(termZ, operatorsZ{o});
termY = kron(termY, operatorsY{o});
termX = kron(termX, operatorsX{o});
end
cell_ZZ{n+1} = termZ;
cell_YY{n+1} = termY;
cell_XX{n+1} = termX;
end


% deals with the periodic term
operatorsZ = horzcat(horzcat( {Sz}, repmat({Id}, 1, N-2) ), {Sz} );
operatorsY = horzcat(horzcat( {Sy}, repmat({Id}, 1, N-2) ), {Sy} );
operatorsX = horzcat(horzcat( {Sx}, repmat({Id}, 1, N-2) ), {Sx} );
termZ = operatorsZ{1};
```

```
termY = operatorsY{1};
termX = operatorsX{1};
for o = 2:N
termZ = kron(termZ, operatorsZ{o});
termY = kron(termY, operatorsY{o});
termX = kron(termX, operatorsX{o});
end
cell_ZZ{N} = termZ;
cell_YY{N} = termY;
cell_XX{N} = termX;


% generates Sz
% generate the fZ cell array of the Hamiltonian:
term = sparse(2,2);
cell_fZ = cell(N,1);
operators = cell(N,1);
for n = 0:N-1
operators = horzcat( horzcat( repmat({Id},1,n), {Sz}), repmat({Id}, 1 , N-1-n));
term = operators{1};
for o = 2:N
term = sparse(kron(term, operators{o}));
end
cell_fZ{n+1} = term;
end

% generates Hamiltonian
Hamiltonian = zeros(2^N,2^N);
for i = 1:N
Hamiltonian = Hamiltonian + 2*x*cell_fZ{i} - (1-x)*(cell_ZZ{i} + cell_XX{i} + cell_YY{i});
end

eigv = eig(Hamiltonian);

end
```

3. Qubits

(a)

(b)

(c)

(d)

(e)