A7_Q4

November 8, 2018

In [7]:
```
using LinearAlgebra

function HMat(J,N,B)
    Ham = zeros(Float32, 2^N, 2^N)
    for Ket = 0:(2^N - 1) # Loop over the kets
        Diagonal::Float32 = 0
        for SpinIndex = 0:N-2 # loop through the indices
            Spin1 = 2*((Ket>>SpinIndex) & 1)-1
            Spin2 = 2*((Ket>>(SpinIndex+1)) & 1) - 1
            Diagonal = Diagonal - 0.25*Spin1*Spin2
        end
        Ham[Ket+1,Ket+1] = J*Diagonal # Fill the diagonal component
        # Adding in the Bra component
        for SpinIndex = 0:N-1
            bit = 2^SpinIndex
            Bra = Ket  bit # Define our Bra for each Ket
            Ham[Ket+1,Bra+1] = -0.5*B # Fill the off-diagonal components
            #println(Ket, " ", Bra)
        end
    end
    return Ham
end

# (J=1,N=2,B=0.5)
Ham = HMat(1,2,0.5)
display(Ham)
print("Min Eigenvalue: ")
println(eigen(Ham).values[1])
print("Groundstate Eigenvector: ")
println(eigen(Ham).vectors[1:4])
```

```
4Œ4 Array{Float32,2}:
 -0.25  -0.25  -0.25   0.0
 -0.25   0.25   0.0   -0.25
 -0.25   0.0    0.25  -0.25
  0.0   -0.25  -0.25  -0.25
```

```
Min Eigenvalue: -0.5590167
Groundstate Eigenvector: Float32[-0.601501, -0.371748, -0.371748, -0.601501]
```

In [18]:
```julia
# Q4
Ham = HMat(1,6,0)
eigenvalues = eigen(Ham).values
m = 0
for i = 1:2^6
    m = m + eigenvalues[i]
end
print("Magnetization: ")
println(m/6)
```

```
Magnetization: 0.0
```

In [19]:
```julia
Ham = HMat(0,6,1)
eigenvalues = eigen(Ham).values
m = 0
for i = 1:2^6
    m = m + eigenvalues[i]
end
print("Magnetization: ")
println(m/6)
```

```
Magnetization: 2.5828679e-6
```

In [20]:
```julia
Ham = HMat(1,6,0.5)
eigenvalues = eigen(Ham).values
m = 0
for i = 1:2^6
    m = m + eigenvalues[i]
end
print("Magnetization: ")
println(m/6)
```

```
Magnetization: -1.1920929e-7
```