Academic Year 2022 – 2023
Research Unit: UCLA - MQST
Team(s): Stephen, Ryan, and Simon
Supervisor(s): Richard Ross

Lab 4: Two Qubit Deutsch-Josza and Grover's Algorithms

Ryan Dougherty

June 16, 2023

Abstract

In this lab we built upon the theory and experiments form the previous lab to implement the two qubit Deutsch-Josza and Grover's algorithms. We constructed all four possible oracle functions for the Deutsch-Josza algorithm and used them to determine whether the encoded function was constant or balanced. This was accomplished in a single query of the quantum oracle. Deutsch-Josza provides a significant speed up over its classical counterpart, which requires at least N/2 queries of an oracle to guarantee a solution. We then used Grover's algorithm to perform an unstructured search for a single element in a set of four elements. Due to the benefits of superposition, phase kickback, and amplitude amplification, we were able to solve this problem in $O(\sqrt{n})$ time, which provides a quadratic speed up over the classical algorithm. We achieved results within a reasonable error margin. However, we were unable to do readouts for our 1H spin states, which inhibits our ability to confidently verify that we achieved the correct result for our Grover's algorithm.

Contents

1 Introduction

1.1 The Deutsch-Josza Algorithm

The Deutsch-Josza is an example of quantum algorithm that provides a speed-up over its classical counterpart. It solves the following problem: Given a boolean oracle function f(x), determine whether the output of f(x) is constant or balanced. Classically this algorithm needs to query at least half of the possible inputs to determine whether the function is constant or balanced. This means that the classical algorithm would take at least $2^{n-1}+1$ queries to determine whether the function is constant or balanced - where n is the number of bits in the input. The quantum algorithm only requires one query to determine whether the function is constant or balanced, providing a significant speed up over the classical algorithm. This quantum advantage comes from the fact that we can place all our qubits in superposition, and then use a quantum representation of the oracle function to query all the possible inputs at once. [Tea22a, qiskit Deutsch-Josza] [DJ92, Deutsch-Josza Algorithm]

1.2 Grover's Algorithm

Grover's algorithm is a quantum algorithm that can be used to solve an unstructured search across a series of elements. In the classical case, this is done by searching through all elements in the set, and checking if the element is the one we are looking for. This means that the time complexity of this algorithm is $O(n)$ for a set of n elements. Grover's algorithm can be used to solve this problem in $O(\sqrt{n})$ time. We will see that this is accomplished by first putting our input in super position and then applying a quantum orcale that tags the element we are looking for. We then apply a series of reflections to our input known as a *diffuser*. This amplifies our tagged state, making it more likely to be measured. [Gro96, Lov Grover]

2 Methods and Procedure

2.0.1 Creating Equal Superposition

Quantum algorithms take advantage of the fact that quantum states can be in superposition of a given basis of states. This is a very powerful property of quantum states that allows us to perform computations that are not possible on classical computers. To create a superposition of the computational states $|0\rangle$ and $|1\rangle$, we apply a Hadamard gate to each qubit. Our process for this was briefly covered in the last lab, but we will go over it again here. We can apply a pseudo Hadamard gate to our qubits by applying the following sequence of pulses: [oP16, MIT pg. 16] [Jon10, Jones pg. 12]

$$I_y^1(\pi/2)I_y^2(\pi/2) \rightarrow I_x^1(\pi)I_x^2(\pi)$$

This pseudo Hadamard gate is off by a global phase, but is still equivalent to the Hadamard sequence computationally. It creates the following unitary operation:

$$\mathrm{H}^{\otimes 2} = \frac{1}{2}\begin{bmatrix} -1 & -1 & -1 & -1 \\ -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & 1 & 1 & -1 \end{bmatrix}$$

2.1 Two qubit Deutsch-Josza Algorithm

In this lab, we develop an algorithm of Deutsch-Josza that solves this for the case where $n = 2$. This has four possible oracle functions, which we will call $f_1(x)$, $f_2(x)$, $f_3(x)$, and $f_4(x)$. The oracle functions are defined by the following truth table: [Tea22a, Qiskit Deutch-Josza]

x	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
0	0	1	0	1
1	0	1	1	0

Where we can see our two functions $f_1(x)$ and $f_2(x)$ are constant, and $f_3(x)$ and $f_4(x)$ are balanced. To build a quantum algorithm, we first need to put our two qubits in superposition using Hadamard gates. We then apply the oracle function to our qubits. Lastly, we apply our hadamard gates again to take our qubits out of superposition and perform a measurement. Each of our oracle functions can be built with a set of pulses in our NMR machine. This is discussed in the following sections. A circuit diagram of the generalized Deutsch-Josza algorithm is shown in figure 1.

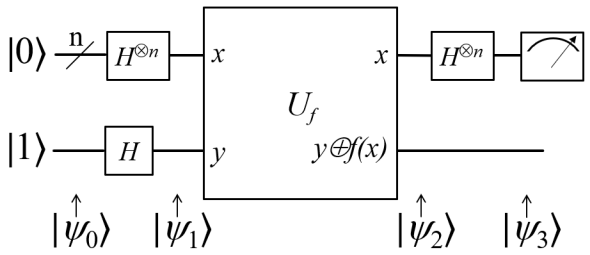
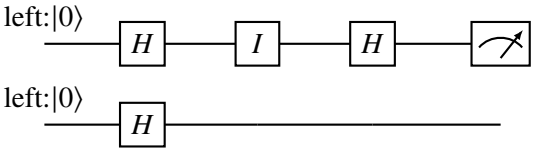


Figure 1 – Circuit diagram of the generalized Deutsch-Josza algorithm.

2.1.1 U1

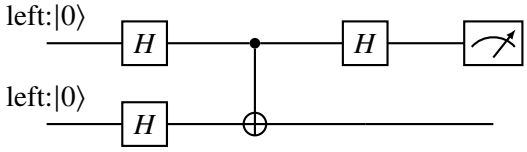
The first oracle function we will consider is for $f_1(x)$. This function is constant, and always returns 0. This can be simply created by doing nothing to our two qubits. The circuit diagram for this oracle function is shown below:



2.1.2 U2

The second oracle is for $f_2(x)$. This function is also constant, and always returns 1. This can be created by applying a NOT gate to our second qubit. This is accomplished by applying the following set of pulses:

$$I_x^2(2)$$

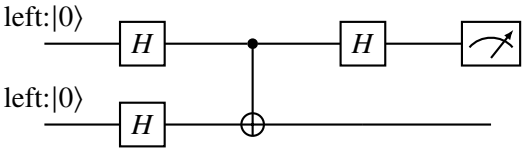


2.1.3 U3

The third oracle is for $f_3(x)$. This function is balanced, and returns 0 when $x = 0$, and returns 1 when $x = 1$. This can be created by applying a CNOT gate. U_3 is accomplished by applying the following set

of pulses:

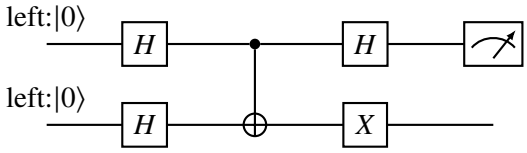
$$\text{CNOT}\{I_z^1(\pi/2) \rightarrow I_z^2(-\pi/2) \rightarrow I_x^2(\pi/2) \rightarrow 2I_z^1 I_z^2(\pi/2) \rightarrow I_y^2(-\pi/2)\}$$



2.1.4 U4

The fourth oracle is for $f_4(x)$. This function is also balanced, and returns 1 when $x = 0$, and returns 0 when $x = 1$. This can be created by applying a CNOT gate, followed by an X gate. U_4 is accomplished by applying the following set of pulses:

$$\text{CNOT}\{I_z^1(\pi/2) \rightarrow I_z^2(-\pi/2) \rightarrow I_x^2(\pi/2) \rightarrow 2I_z^1 I_z^2(\pi/2) \rightarrow I_y^2(-\pi/2)\} \rightarrow I_x^2(\pi)$$



2.1.5 Example of Deutsch-Josza Using U_1

Lastly, lets go through a concrete mathematical example of how to do the Deutsch-Josza algorithm using our oracle function U_1. We start with our two qubits in the state $|0\rangle \otimes |0\rangle$. We then apply our Hadamard gates to put our qubits in superposition. We then apply our oracle function U_1 to our qubits.

$$\begin{aligned} |\psi_1\rangle = U_1|\psi_0\rangle &= \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} \\ &= |00\rangle - |01\rangle + |10\rangle - |11\rangle \end{aligned}$$

Lastly, we apply our Hadamard gates again to take our qubits out of superposition. We then perform a measurement on our qubits. The result of this measurement will always be $|0\rangle \otimes |0\rangle$. This is because our oracle function U_1 is constant, and always returns 0. This applies $U_f|x\ y\rangle = |x\ \ f(x) \oplus y\rangle$ for a given quantum state.

$$\begin{aligned} |\psi_2\rangle &= \frac{1}{2}\{|0\ \ 0\rangle - |0\ \ f(0) \oplus 1\rangle + |1\ \ f(1)\rangle - |1\ \ f(1) \oplus 1\rangle\} \\ &= \frac{1}{2}\{(-1)^{f(0)}|0\rangle(|0\rangle - |1\rangle) + (-1)^{f(1)}|1\rangle(|0\rangle - |1\rangle)\} \\ &= \frac{(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} \end{aligned}$$

2.2 Two qubit Grover's Algorithm

For our implementation of Grover's algorithm, we use 2 qubits. We make it our goal to tag the state $|11\rangle$. We start with our two qubits in the state $|0\rangle \otimes |0\rangle$. We then apply our Hadamard gates to put our qubits in superposition. Next is the application of the oracle function O to our qubits. We then apply our diffuser

function to amplify the results of our oracle function. We then apply our Hadamard gates again to take our qubits out of superposition and measure.

Lastly, its worth mentioning that Grover's algorithm works probabilistically. It works based on the idea that the probability of measuring a state is proportional to the amplitude of out tagged state. When running Grovers for many qubits, we must run the algorithm multiple times to get a good estimate of the probability of measuring our tagged state. However, in our case with only 2 qubits that tags the state we can run the algorithm once and get a good estimate of the probability of measuring our tagged state. [Gro96, Lov. Grover] [Tea22b, Qiskit, Grovers Algorithm]

2.2.1 Oracle Function O

The oracle function O is a function that tags a given state we're searching for $|\omega\rangle$. To create a generalized oracle function, we need to create a unitary that gets applied a given input state $|x\rangle$ and returns a state in the form: [Tea22b, Qiskit Grover's Algorithm]

$$O|x\rangle = \begin{cases} |x\rangle & \text{if x } \neq \omega \\ -|x\rangle & \text{if x } = \omega \end{cases}$$

We can think about encoding this into a function f that takes an input x and returns either 0 or 1. If x is equal to ω, then $f(x) = 1$. If x is not equal to ω, then $f(x) = 0$. We can consider a unitary O that accomplishes this with the following unitary operation.

$$O|x\rangle = (-1)^{f(x)}|x\rangle$$

This technique is known as a *phase kickback* and is used to tag a desired state outputted from the function $f(x)$. We can visually represent what this transformation does by considering two orthogonal vectors $|\omega\rangle$ and $|s'\rangle$ that span a hilbert space. Where $|\omega\rangle$ are the states we are searching for and $|s'\rangle$ are the states we are not searching for. Our starting state can be thought of as the vector $|s\rangle$, which exists in some superposition of both. When we apply our oracle function O, we get a result that is reflected along the axes of the vector $|s'\rangle$. This is shown in the figure below:

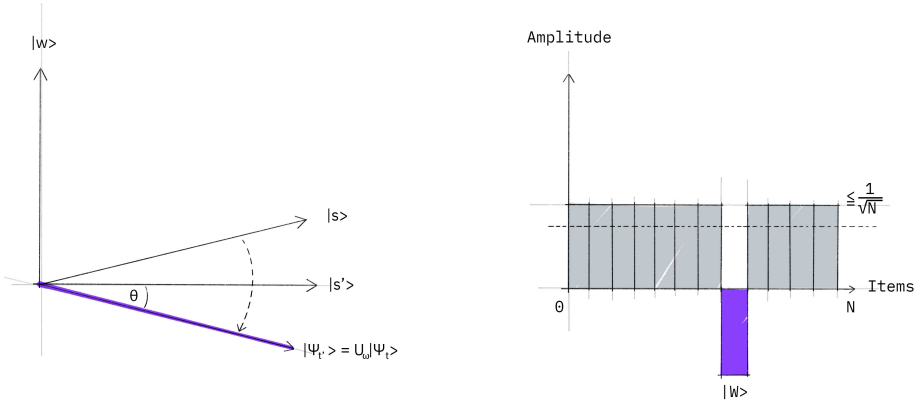


Figure 2 – Tagging our desired state with the oracle function O

2.2.2 Building the Oracle O for state $|11\rangle$

For our lab experiment, we chose to create a function $f(x)$ that tags the state $|11\rangle$. We're able to build this quite simply with the following unitary operation:

$$O = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

This is effectively just a CZ gate that flips the sign of the state $|11\rangle$. We can construct this on our NMR machine with the following set of pulses:

$$2I_z^1 I_z^2(\pi/2) \rightarrow I_y^1(-\pi/2)I_y^2(-\pi/2) \rightarrow I_x^1(\pi/2)I_x^2(\pi/2) \rightarrow I_y^1(\pi/2)I_y^2(\pi/2)$$

Where, as mentioned in our previous lab, $2I_z^1 I_z^2(\pi/2)$ is the free evolution delay of our qubit precession in the applied magnetic field. This sequence give us our oracle function O that tags the state $|11\rangle$.

2.2.3 Diffuser P

The diffuser function P is a unitary that is used to amplify the amplitude of our tagged state. The diffuser can be thought of as reflecting our tagged state along the axes of the starting state vector $|s\rangle$. Mathematically, this can be represented as the following transformation: [oP16, MIT, pg. 16]

$$P = 2|s\rangle\langle s| - I$$

We create this unitary on our NMR machine with the following sequence of pulses:

$$2I_z^1 I_z^2(\pi/2) \rightarrow I_y^1(-\pi/2)I_y^2(-\pi/2) \rightarrow I_x^1(-\pi/2)I_x^2(-\pi/2) \rightarrow I_y^1(\pi/2)I_y^2(\pi/2)(\pi)$$

This is equivalent to the following unitary:

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

Visually, we can see what this transformation does to our starting state vector $|s\rangle$ in the figure below:

2.2.4 Putting it all together

Finally, we can combine all these steps together to create an iteration of our Grover's algorithm. As mentioned earlier, we only need to run this algorithm once to get a good estimate of the probability of measuring our tagged state. We can run this algorithm by applying the following set of gates:

$$G = H^{\otimes 2} P H^{\otimes 2} O$$

This will successfully apply our oracle and apply the diffuser to amplify the amplitude of our tagged state. The only thing missing is to properly prepare our initial state in superposition. This can of course be accomplished by first applying a Hadamard gate to each qubit. This will give us the following gate sequence for the Grover's algorithm:

$$|\psi_k\rangle = G^k H^{\otimes 2}|00\rangle$$

Where k is the number of iterations we want to run our algorithm for and $|\psi_k\rangle$ is the state of our qubits after the k iterations.

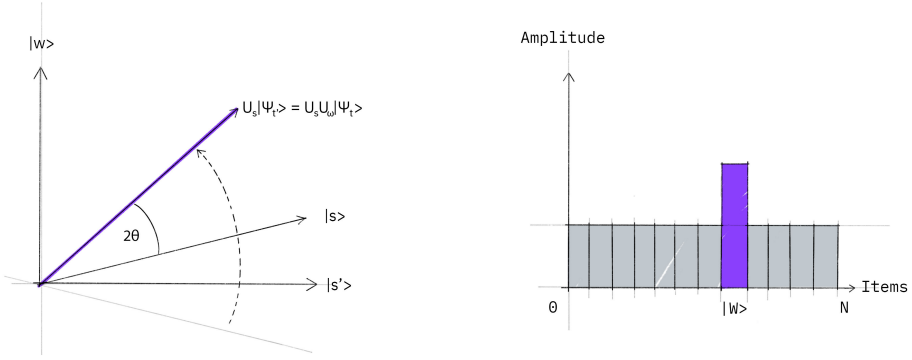


Figure 3 – Application of diffuser P to amplify the amplitude of our tagged state, making it more likely to be measured

3 Results and Discussion

3.1 Experiment 1: 2 Qubit Deutsch Josza Algorithm

Using the methods explained in section 2.1 we built and ran a 2 qubit Deutsch Josza algorithm on our NMR machine. We ran each oracle U_1, U_2, U_3, and U_4 each with the pseudo pure states $P0$, $P1$, and $P2$. We used the combined sum of each of these results to take an exhaustive average - the process of which is explained in detail in our previous lab report. With this exhaustive average, we can sum out the noise of our other spin states and get a more accurate result for the starting state $|00\rangle$.

3.1.1 Results for U_1

U_1 is a constant function that always returns $|0\rangle$. Therefore, we expect our algorithm to always return $|0\rangle$ since its balanced. Taking look at our exhaustive average stage in the lower graph, we see we have a large peak integral at the lowest energy peak. This corresponds to the state $|0\rangle$ on 13C, which implies that our algorithm worked as expected for this oracle.

We got a peak of 0.02 for the state $|1\rangle$, which is quite good since we expect all of these peaks to sum out to zero.

3.1.2 Results for U_2

U_2 is a constant function that always returns $|1\rangle$. Therefore, we expect our algorithm to always return $|0\rangle$ since it is balanced. Taking look at our exhaustive average stage in the lower graph, we see we again have a large peak integral at the lowest energy peak. This corresponds to the state $|0\rangle$ on 13C, which implies that our algorithm worked as expected for our constant oracle U_2.

Here, we got a peak of 0.0 for the state $|1\rangle$, which is exactly what we expect.

3.1.3 Results for U_3

U_3 is a balanced function that returns $|0\rangle$ if the input is $|0\rangle$ and $|1\rangle$ if the input is $|1\rangle$. Therefore, we expect our algorithm to always return $|1\rangle$ since it is balanced. Taking look at our exhaustive average stage in the lower graph, we see we have a large peak integral at the highest energy peak. This corresponds to the state $|1\rangle$ on 13C, which implies that our algorithm worked as expected for our balanced oracle U_3.

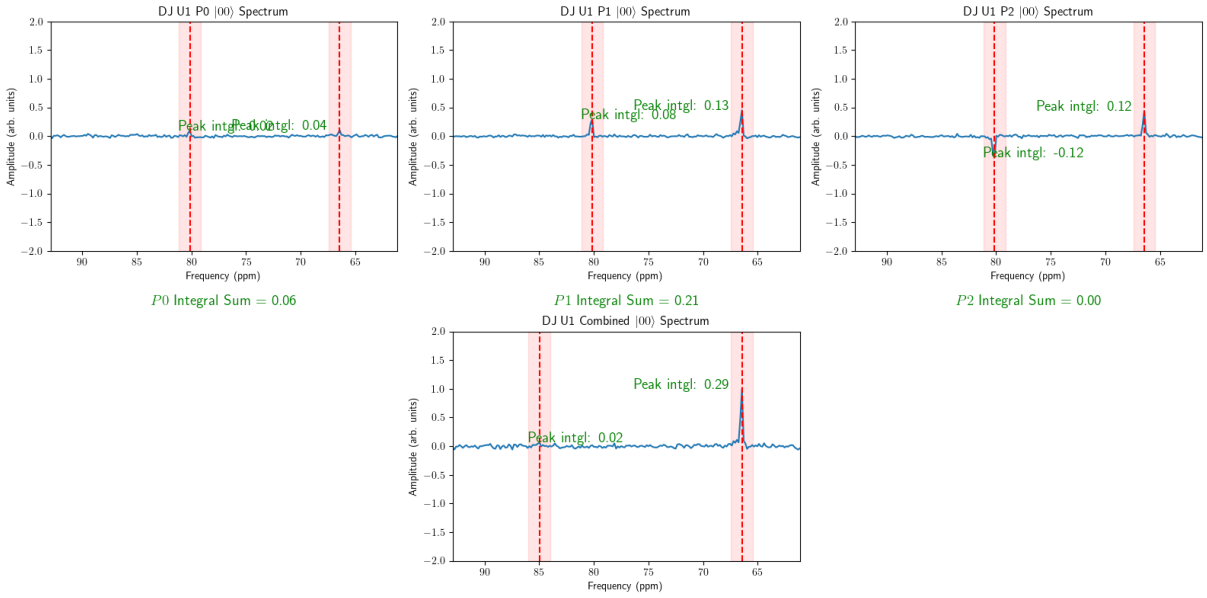


Figure 4 – Results for Deutsch Josza of U_1. We see a large peak integral at the lowest energy peak, which corresponds to the state $|0\rangle$ on 13C. We see a peak integral of 0.02 for the state $|1\rangle$ and a peak integral of 0.29 for the state $|0\rangle$

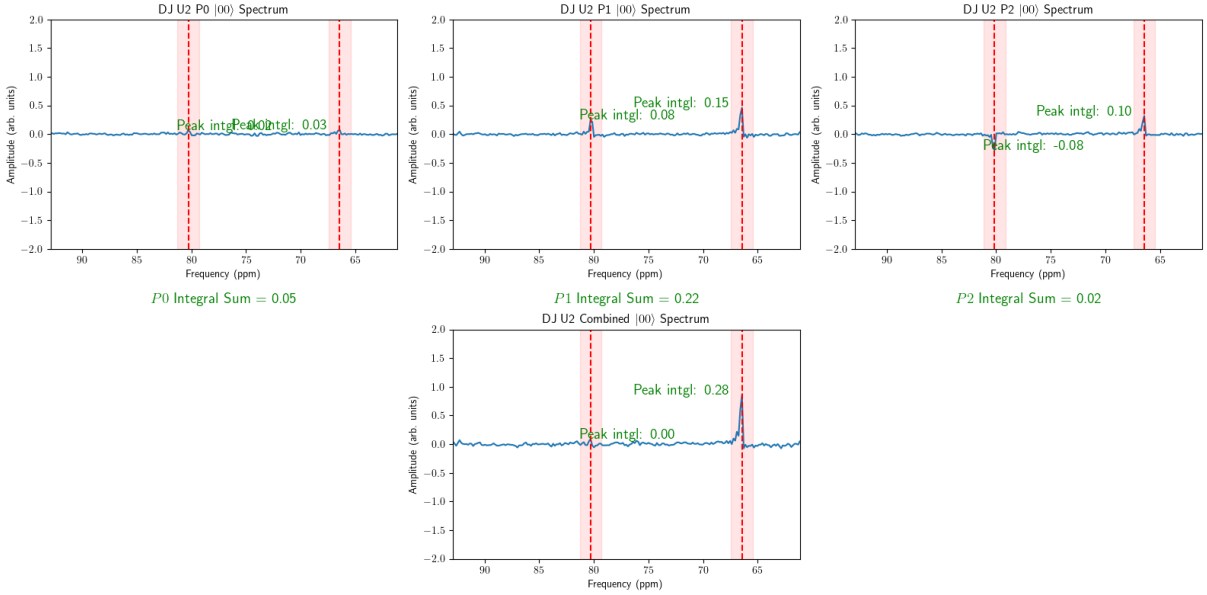


Figure 5 – Results for Deutsch Josza of U_2. We see a large peak integral at the lowest energy peak, which corresponds to the state $|0\rangle$ on 13C. We see a peak integral of 0.0 for the state $|1\rangle$ and a peak integral of 0.28 for the state $|0\rangle$

We see a peak of 18.26 for the state $|1\rangle$ and a peak of -5.07 for the state $|0\rangle$. We'd expect the state $|0\rangle$ to have a peak of 0.0, but we get a slightly negative value due to error. This is most likely due to slight experimental error or incorrect phase correction on our readout.

Lastly, its worth mentioning that we had to multiply $P2$ results for U_3 by 0.25 to get peaks that were consistent with our expectations. Why our $P2$ was so different from our $P0$ and $P1$ is unclear, but it is likely due to the fact we ran our experiments with different rx gain values.

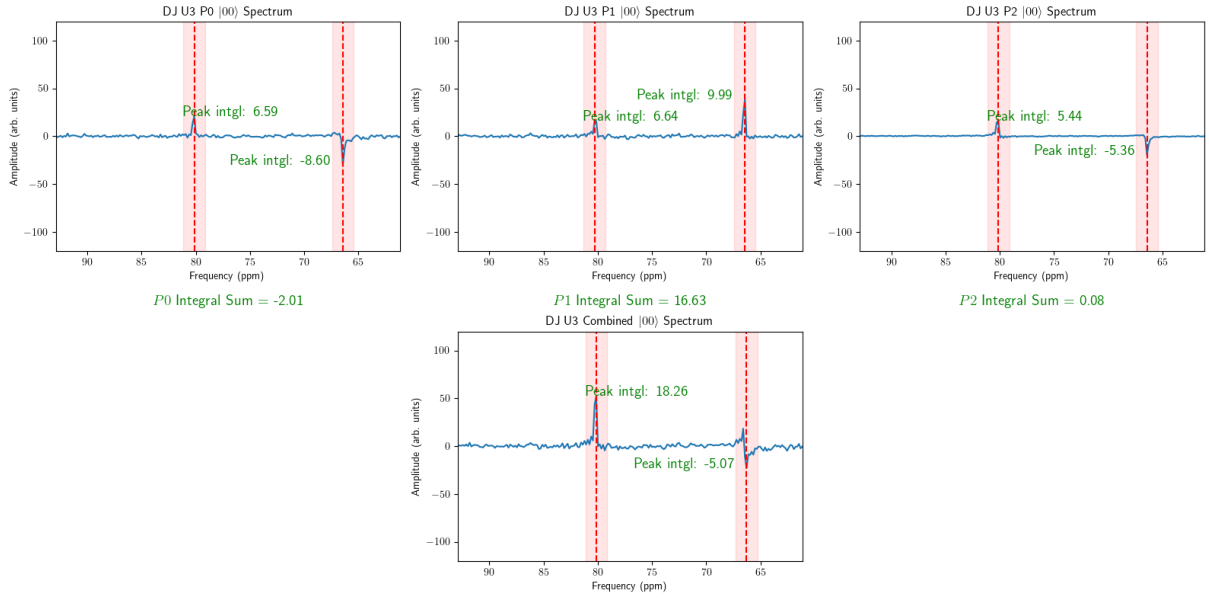


Figure 6 – Results for Deutsch Josza of U_3. We see a large peak integral at the highest energy peak, which corresponds to the state $|1\rangle$ on 13C. We see a peak integral of 18.26 for the state $|1\rangle$ and a peak integral of -5.07 for the state $|0\rangle$

3.1.4 Results for U_4

U_4 is our last balanced oracle. It returns $|0\rangle$ if the input is $|1\rangle$ and $|1\rangle$ if the input is $|0\rangle$. Therefore, we expect our algorithm to always return $|1\rangle$ since it is balanced. Taking look at our exhaustive average stage in the lower graph, we see we have a large peak integral at the highest energy peak. This corresponds to the state $|1\rangle$ on 13C, which implies that our algorithm worked as expected for our balanced oracle U_4.

We see a peak at 17.70 for the state $|1\rangle$ and a peak at -2.15 for the state $|0\rangle$. These results are quite good and consistent with what we'd expect. However, a peak of -2.15 is slightly high for what we'd expect for the state $|0\rangle$.

Just like our U_3 oracle, we had to correct our P_2 results by a factor of 0.25 in order to get consistent results. This again was most likely due to experimental error or incorrect phase correction on our readout.

3.2 Experiment 2: 2 Qubit Grovers Algorithm

We followed the methods discussed in section 2.2 to create a two qubit algorithm of Grover's that searches for the $|11\rangle$ state. We again used an exhaustive average to find our tagged output state with minimal noise from our other spin states.

Our results are shown in figure 8. We see a large negative peak of -4.27 at our $|1\rangle$ 13C state and a very small peak of -0.04 at our $|0\rangle$ state. This implies that either the $|11\rangle$ or $|01\rangle$ state was tagged. In order to assure we correctly tagged the $|11\rangle$ state, we would have had to run our experiment once more targeting 1H readout instead of 13C. However, due to the time constraints and limitations of our NMR system, we were unable to do this. However, given the accuracy of our results and pulse sequences, our group concluded that we correctly achieved tagging the $|11\rangle$ state.

Overall, our results were very good for our 13C readout. We see a large negative peak at our $|1\rangle$ state and a nearly zero peak at $|0\rangle$.

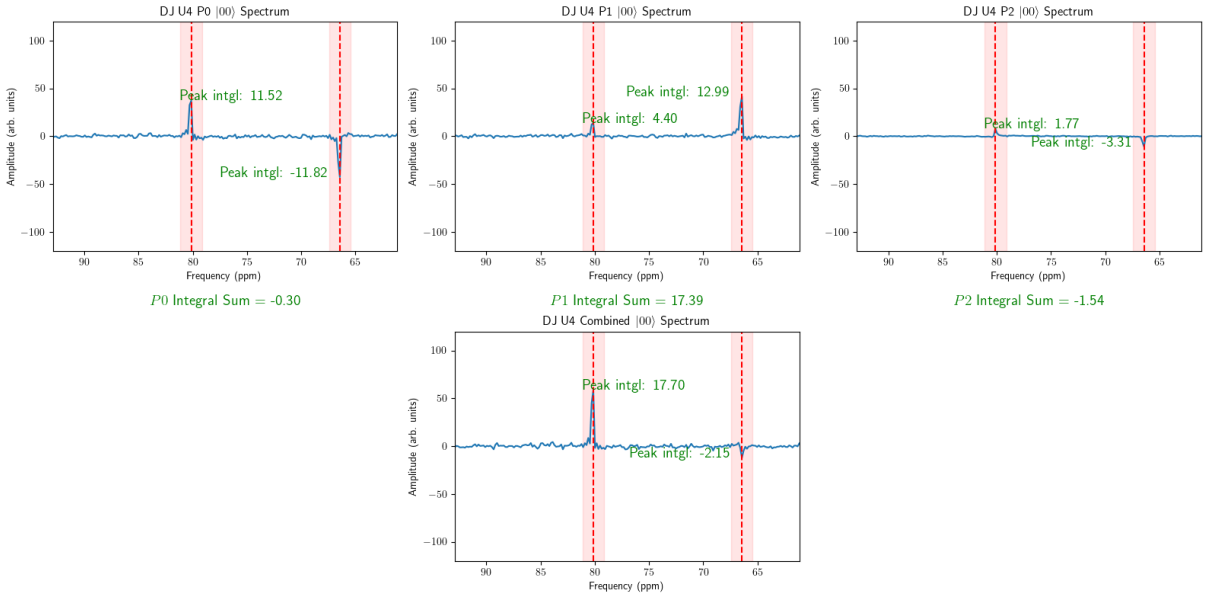


Figure 7 – Results for Deutsch Josza of U_4. We see a large peak integral at the highest energy peak, which corresponds to the state $|1\rangle$ on 13C. We see a peak integral of 17.70 for the state $|1\rangle$ and a peak integral of -2.15 for the state $|0\rangle$

3.2.1 Multiple Iterations

Lastly, we were instructed to run Grovers for multiple iterations to see how it compares to a single iteration result. Unfortunately, we did not find out about this additional requirement until after we'd finished our experiments. This was due to the fact that the code in our lab's Jupyter notebook did not compile and was not displayed correctly when first looked through it.

Considering the math behind Grovers, we'd expect to see the same results if we ran it 4, 7, and 10 times instead of a single iteration. This is due to how the diffuser works in Grovers. It will continually rotate the tagged state around the $|\omega\rangle$ and $|s'\rangle$ vector space explained in our theory section, and bring it back again to the $|s\rangle$ state. Each iteration rotates this state roughly 25 degrees in this vector space, which is why we'd expect to see the same results for 4, 7, and 10 iterations.

However, the more iterations we perform, the more pulses we perform in our system. We know this will cause more noise in our system, which will cause our results to be less accurate. Therefore, its beneficial to run Grovers for as few iterations as possible - even if several iteration counts produce the same results.

4 Conclusion

In conclusion, we were able to successfully implement Deutsch-Josza and Grover's Algorithm on a 2 qubit system. Our Deutsch-Josza had quite good results on our readout of 13C (our DJ output state). We saw excellent results for our U_1 and U_2 oracles. Our U_3 and U_4 oracles had not quite as good results and had slight dispersion curves. However, they still had large peaks in the correct states. Some of this was likely due to experimental error or incorrect phase correction on our readout. Its also worth mentioning that we had to correct our P_2 results by a factor of 0.25 in order to get consistent results. This was most likely due to differing values of our receiver gain.

For our implementation of Grover's Algorithm, we were able to successfully tag either the $|01\rangle$ or $|11\rangle$ state. Due to time constraints and limitations of our NMR system, we were unable to run our experiment again targeting 1H readout. Had we been able to accomplish this, then we would have been able to

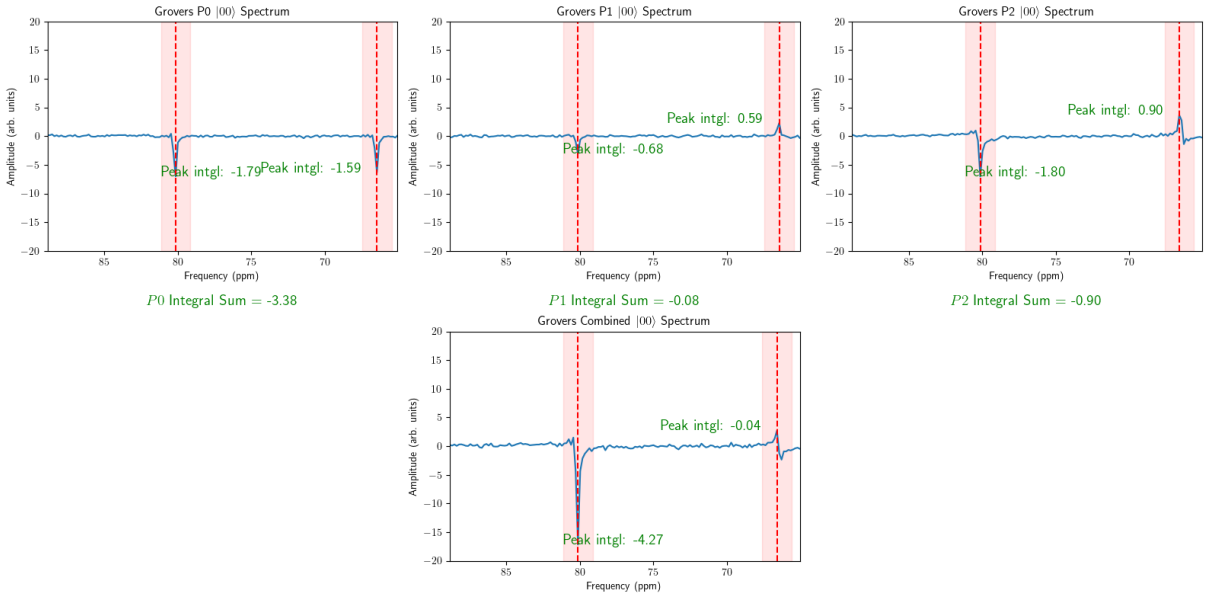


Figure 8 – Results for Grover's Algorithm. We see a large negative integral at the $|1\rangle$ state and a very small integral at the $|0\rangle$ state. This implies that either the $|11\rangle$ or $|01\rangle$ state was properly tagged.

confirm that we tagged the $|11\rangle$ state. However, given the accuracy of our results and pulse sequences, our group concluded that we correctly achieved tagging the $|11\rangle$ state.

Overall, we conclude that our results were quite good. After taking an exhaustive average over our results, we saw clear large amplitudes in the correctly expected states with relatively minimal error.

5 Diagonositics

Next we will discuss the diagnostics of our NMR system. We will discuss the results of our T1 and T2 experiments, as well as our 1D

References

[DJ92] David Deutsch and Richard Josza. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 439(1907):553–558, 1992.

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing - STOC '96*, 1996.

[Jon10] Jonathan A. Jones. Quantum computing with nmr. *Progress in Nuclear Magnetic Resonance Spectroscopy*, 59(2):91–120, Nov 2010.

[oP16] MIT Department of Physics. volume 180. 2016.

[Tea22a] Qiskit Team. Deutsch-jozsa algorithm, Nov 2022.

[Tea22b] Qiskit Team. Grover's algorithm, Nov 2022.