

Notes on quantum search

James D. Whitfield

August 30, 2009

Abstract

This set of notes is meant to supplement a set of introductory lectures given in the Aspuru-Guzik group in the spring of 2009. The intended audience was the new practitioner of quantum computing. The purpose of the lectures was to give sufficient background for understanding and evaluating results in quantum computing. This goal was accomplished by introducing the circuit model of quantum computing and two algorithms. This particular set of notes focuses on the Grover search algorithm.

1 Overview and introduction to the quantum search problem

Quantum physics requires us to think about processing and manipulating information different than classical physics[1, 2]. To quote R. Landauer: ‘Information is physical.’ The effect of using computational laws based in quantum physics can result dramatic differences from the classical counterpart and in this document you will get the flavor of a quantum algorithm that is quadratically faster than the classical counterparts, namely Grover’s search. The search problem is to find a marked state among $N = 2^n$ unsorted items. The canonical example is that of finding a particular name given a phone number and a phone book.

Classically, if we want to find a items from a set W it will take $O(N)$ time to search if W only has one element. To formalize these ideas suppose for database D we have function $f : D \rightarrow \{0, 1\}$ defined as

$$f(x) = \begin{cases} 1, & x \in W \\ 0 & x \notin W \end{cases} \quad (1)$$

In this document, we will only discuss the case where only one item is marked. To find marked item s , we will have to apply $f(x)$ to each $x \in 0, 1, \dots, N - 1$. On average this will take $N/2$ tries and the worst case it will take $N - 1$ tries. In other words, as the database increases in size the time required to search it will scale linearly with the size of the database. The advantage of the quantum algorithm is only in the case of unsorted databases. Databases can usually be sorted, structured, and ordered according to some sort of enumeration. If the classical algorithm is allowed to spend $O(N \log N)$ time to structure the database searches such as binary search can be performed in $O(\log(N))$ time and in some cases better [3].

To provide a common framework for discussing quantum algorithm, the appendix 6 will provide an introduction to the quantum circuit model. This model unifies thinking about quantum computing, but it is not the only way to think about quantum computing. We illustrate how superposition can be used for computational advantage using the simple Deutsch algorithm¹. If one has sufficient background in quantum computing this section can be skipped with little loss however it may be skimmed if the notation is unfamiliar.

The quantum search algorithm uses discrete steps to evolve the uniform superposition of all states to the desired state in $O(\sqrt{N})$ steps. Beginning the quantum search at the uniform superposition of all items already illustrates the radically different nature of the quantum search and the classical search. In the classical search, one must begin from a definite state and then proceed to search. The quantum algorithm proceeds by using the oracle to change the sign of marked items as explained in the next section §2. A second operator reflects each amplitude about the mean amplitude and is examined in §3. The combination of the two operations is called a Grover operation (§4) and we’ll see that this operator acts non-trivially in

¹The superposition are different than mixtures (mixed states) and come as result of the Schrödinger wave equation so one must be wary of attributing all computational advantage to superposition as classical objects like waves also exhibit superposition of states.

the two dimensional space which is spanned by the initial state and the answer state. As the composition of two reflections is a rotation, the Grover operator has a quaint geometrical interpretation which we will see in section §4.

2 Phase shift oracle

Using an oracle that computes function f of (1), we would like to create an operator that changes the sign of marked items. First we give the form of a unitary oracle, second we show how we can pick input states to create the desired operation. See §6 for any unfamiliar notation.

Now suppose we had an oracle U_o that operates as $U_o|x\rangle|y\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle$. The reason for the oracle to be implemented as a two qubit operation is that we wish it to be unitary. Each unitary operation is invertible so each input must map to a unique output (in other words: the operation must be injective or one-to-one). For functions that aren't injective using extra space to keep the input allows the operation to be reversible[4]. In our case, U_o calculates $f(x)$ corresponding to (1).

Note that if we let the second qubit, $|y\rangle$, be the state $|X-\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$ then oracle changes the sign of an input vector if it is a marked state. Observe:

$$U_o \left(\frac{|x\rangle|0\rangle - |x\rangle|1\rangle}{\sqrt{2}} \right) = \left(\frac{|x\rangle|f(x)\rangle - |x\rangle|1 \oplus f(x)\rangle}{\sqrt{2}} \right).$$

Temporarily ignoring normalization, either $f(x) = 0$ and the output state is $|x\rangle(|0\rangle - |1\rangle)$ or $f(x) = 1$ and output state is $|x\rangle(|0\rangle + |1\rangle)$. Therefore we write

$$U_o|x\rangle|X-\rangle = (-1)^{f(x)}|x\rangle|X-\rangle.$$

If we always consider the second qubit in state $|X-\rangle$, then the sign of answer state, $|w\rangle$, changes and the other $N - 1$ basis vectors are left unchanged. Thus we write:

$$U_o = \mathbf{1} - 2|w\rangle\langle w| \tag{2}$$

3 Reflection about the mean

The oracle followed by a second operator U_s completes a single Grover iteration.

$$U_s = 2|s\rangle\langle s| - \mathbf{1}. \tag{3}$$

The state $|s\rangle$ is selected as to have some overlap with all vectors including those of the answer space. We pick:

$$|s\rangle = H^{\otimes n}|0 \dots 0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle, \tag{4}$$

The operator U_s with s defined as in (4) is called an inversion about the mean. First we defined the mean of the amplitudes of $|v\rangle$ as:

$$\bar{v} = \sum_k \frac{\langle k|v\rangle}{N}.$$

Starting from $|s\rangle\langle s|$, inserting a resolution of the identity [5] and finally using definition (4) we have:

$$|s\rangle\langle s|v\rangle = \sum_{i=0}^{N-1} |s\rangle\langle s|i\rangle\langle i|v\rangle = \sum_i \left(\frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle \right) \left(\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \langle k| \right) |i\rangle\langle i|v\rangle \tag{5}$$

$$= \frac{1}{N} \sum_{ijk} |j\rangle\langle k|i\rangle\langle i|v\rangle = \frac{1}{N} \sum_{ijk} \delta_{ki} \langle i|v\rangle |j\rangle = \sum_j \frac{\sum_i \langle i|v\rangle}{N} |j\rangle \tag{6}$$

$$= \sum_j \bar{v} |j\rangle \tag{7}$$

Putting it all together, we see U_s performs the following operation: $U_s|v\rangle \rightarrow \sum_j (2\bar{v} - v_j)|j\rangle$ with $v_j \equiv \langle j|v\rangle$. Looking at figure 1, we see that this operation has the form of an inversion.

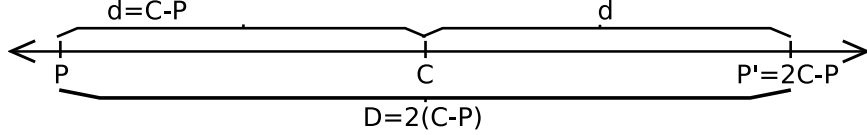


Figure 1: The reflection of point P about point C results in point P' . The distance from P to C is $d = C - P$ and the distance from P to P' is $D = 2d = 2(C - P)$. It follows that $P' = P + 2(C - P) = 2C - P$.

4 The Grover operator and its two-dimensional operation

The Grover operator is given by

$$G = U_s U_o.$$

The Grover search works by evolving the state s towards the answer state $|w\rangle$ using only a two dimensional space. This is because the Grover operator does not evolve state outside of this two dimensional space. The states that have no overlap with $|s\rangle$ and $|w\rangle$ are eigenstates of G with eigenvalue -1 . We use equations (2) and (3) to expand G as:

$$G = 2|s\rangle\langle s| - 2|w\rangle\langle w| - 4\langle s|w\rangle|s\rangle\langle w| - \mathbf{1}, \quad (8)$$

and we can see that if $|t\rangle$ has no overlap with w or s then $G|t\rangle = -|t\rangle$. Although $|s\rangle$ is a superposition of all states in the database basis, the vectors w and s cannot span the entire space. We can give an example of such a vector t . Any superposition of an even number of states with alternating sign is orthogonal to $|s\rangle$, for instance $|t\rangle = H^{\otimes n}|1 \cdots 1\rangle = (1/\sqrt{N}) \sum (-1)^x |x\rangle$. If the superposition of states that being constructed with alternating signs does not include answer states, then we have that this vector is in the $N - 2$ dimensional space orthogonal to $\text{span}\{|w\rangle, |s\rangle\}$.

The trajectory generated by repeated application of G on $|s\rangle$ explores this two dimensional space of $\text{span}\{|w\rangle, |s\rangle\}$. To use an orthonormal basis consider normalized state $|r\rangle$ that is constructed using a method like Gram-Schmidt decomposition [6] given by:

$$|r\rangle = \sqrt{\frac{N}{N-1}}|s\rangle - \sqrt{\frac{1}{N-1}}|w\rangle = \sqrt{\frac{1}{N-1}} \sum_{x \neq w} |x\rangle. \quad (9)$$

From (9) we derive that $|s\rangle = \sqrt{N-1/N}|r\rangle + N^{-1/2}|w\rangle$.

Assuming that G is real we know that G is unitary if and only if its two dimensional matrix representation takes the following form[6]:

$$G = T(\theta) = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}, \quad (10)$$

for some real value of θ . To begin using this geometrical picture, we assume a matrix representation of state $|v\rangle$ is given by:

$$|v\rangle = \begin{bmatrix} \langle w|v\rangle \\ \langle r|v\rangle \end{bmatrix}$$

We will obtain the value of θ by observing that the effect of G on $|r\rangle$ using (8) and (9) yielding,

$$G|r\rangle = (2|s\rangle\langle s| - 2|w\rangle\langle w| - 4\langle s|w\rangle|s\rangle\langle w| - \mathbf{1})|r\rangle \quad (11)$$

$$= \left(1 - \frac{2}{N}\right)|w\rangle - \frac{2\sqrt{N-1}}{N}|r\rangle \quad (12)$$

$$= \sin \theta |w\rangle + \cos \theta |r\rangle \quad (13)$$

To characterize initial state s we'll establish the value of ϕ as the angle between $|s\rangle$ and $|r\rangle$. By inserting this angle of rotation into a rotation matrix we can rotate a vector beginning in state $|r\rangle$ to state $|s\rangle$, and the following equation is satisfied:

$$\begin{bmatrix} \cos \phi & \sin \phi \\ -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \sin \phi \\ \cos \phi \end{bmatrix} = \begin{bmatrix} \langle w|s\rangle \\ \langle r|s\rangle \end{bmatrix}. \quad (14)$$

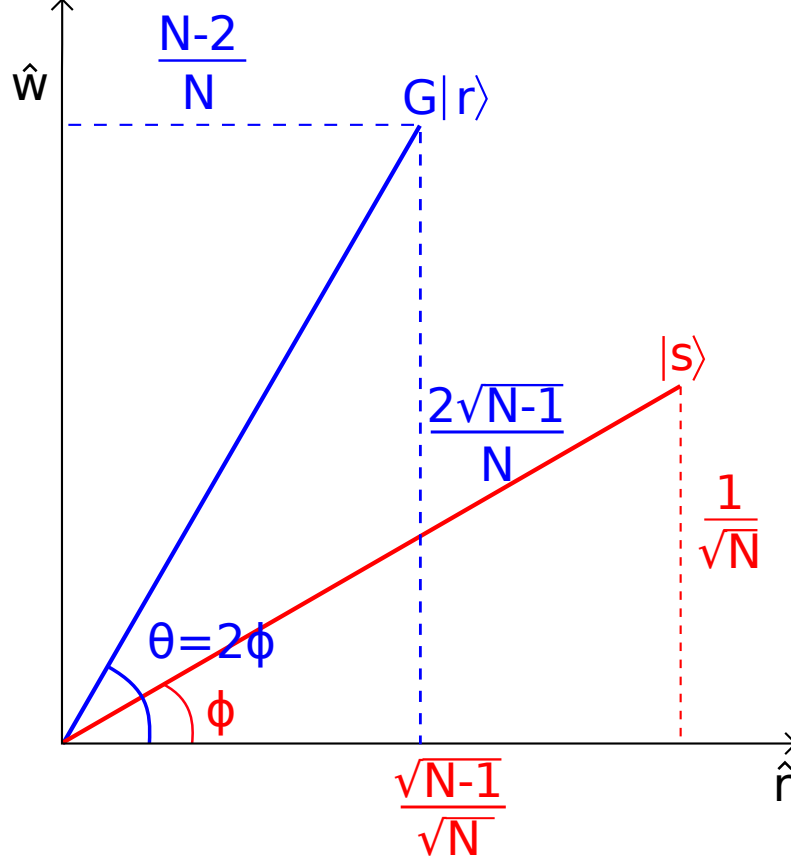


Figure 2: To summarize equations (13), (15), and (18), we depict the states $G|r\rangle$ and $|s\rangle$. The answer state $|w\rangle$ is parallel to the ordinate axis. The abscissa is the orthogonal state $|r\rangle$ defined in (9). The uniform superposition of all states is $|s\rangle$ and is characterized in (15). If ϕ is the angle between the \hat{r} axis and $|s\rangle$ then the Grover operator G performs a rotation of 2ϕ . After $O(\sqrt{N})$ applications of the G to the state $|s\rangle$ there is high probability that measurement will result in marked state w .

From (9) we derive that

$$|s\rangle = \sqrt{\frac{N-1}{N}}|r\rangle + \sqrt{\frac{1}{N}}|w\rangle = \sqrt{\frac{1}{N}} \begin{bmatrix} 1 \\ \sqrt{N-1} \end{bmatrix} \quad (15)$$

and it follows $\cos \phi = \sqrt{(N-1)/N}$ and $\sin \phi = \sqrt{1/N}$.

Using geometric arguments we can show that $2\phi = \theta$. One proof using the double angle formula and (13) is given by:

$$\sin 2\phi = 2(\sin \phi)(\cos \phi) \quad (16)$$

$$= 2\sqrt{\frac{N-1}{N}}\sqrt{\frac{1}{N}} \quad (17)$$

$$= \sin \theta \quad (18)$$

A summary of these geometric results is given in figure 2

Thus after k iterations we have that our initial state with angle ϕ has been rotated to a vector characterized by angle $\varphi_k = k(2\phi) + \phi$. When φ_k is close to $\pi/2$ we have accomplished to goal of creating high overlap with the answer. So if N is large then the state $|s\rangle$ is approximately $|r\rangle$. Then we will need the following to

be approximately satisfied.

$$\varphi_k = \frac{\pi}{2} = (2k + 1)\phi \quad (19)$$

$$\frac{\pi}{2\phi} = 2k + 1 \quad (20)$$

$$k = \frac{\pi}{4\phi} - \frac{1}{2} \quad (21)$$

From (15) we had that $\sin \phi = 1/\sqrt{N}$. Using the small angle approximation we can say $\phi = 1/\sqrt{N}$. Substituting into (21) we have:

$$k = \frac{\pi}{4}\sqrt{N} - \frac{1}{2} \approx \frac{\pi}{4}\sqrt{N}. \quad (22)$$

Hence the algorithm obtains the answer with high probability after a number of iterations that is quadratic in N .

5 Conclusions

At this point, the reader should be familiar with quantum computing and the main ideas of the Grover search. There are caveats and details that we have left unsaid as not to dilute the main message. The hope is that the reader will be able to recognize the main ideas of Grover search and be able to read more if interested. Recommended reading is Chapter 6 of [2] and section 6.4 of [1].

Three extensions are worth mentioning. First, amplitude amplification is a general framework for quantum algorithms such as Grover search and was first introduced in ref. [7]. Second, there is a continuous version of Grover algorithm that was first discussed in [8] and is also discussed in [2]. Finally, we mention that quantum walks have led to many extensions of Grover search. A review can be found by Ambainis [9].

References

- [1] J. Preskill. Physics 219/computer science 219 quantum computation course notes. available at <http://www.theory.caltech.edu/~preskill/ph219/index.html>.
- [2] Michael Nielsen and Isaac Chuang. Quantum computation and quantum information. *Cambridge University Press*, 2001.
- [3] M. Lanzagorta. Quantum algorithms. In *Segunda Escuela Mexicana de Verano en Computación e Información Cuánticas*, 2007.
- [4] C. H. Bennett. Logical reversibility of computation. *IBM J. Res. Develop.*, 17, 1973.
- [5] J. J. Sakurai. *Modern Quantum Mechanics*. Addison-Wesley Publishing Company, 1994.
- [6] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 2005.
- [7] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp. Quantum amplitude amplification and estimation. *arxiv:quant-ph/0005055*, 2000.
- [8] E. Farhi and S. Gutmann. Analog analogue of a digital quantum computer. *Phys. Rev. A*, 57:2403, 1998.
- [9] A. Ambainis. Quantum search algorithms. *arxiv:quant-ph/0504012*, 2005.
- [10] D. Deutsch. The church-turing principle and the universal quantum computer. *Proc. Royal Soc. A*, 400:97–117, 1985.

Original search references

Grover's first papers on search:

- Lov K. Grover's first paper on quantum searching was *A fast quantum mechanical algorithm for database search* .
 - Proceedings, 28th Annual ACM Symposium on the Theory of Computing (STOC), May 1996, pages 212-219. <http://arxiv.org/abs/quant-ph/9605043>
- He wrote a second paper aimed at the physics audience *Quantum Mechanics helps in searching for a needle in a haystack*
 - Phys. Rev. Lett. **79** 325-328 (1997), arXiv:quant-ph/9706033

6 Appendix: Introduction to quantum computing

We will first provide standard textbook [2] quantum computing notions for the uninitiated. Just as classical computation is based around the notion of a bit the basic unit of quantum information is the qubit. Any two level quantum mechanical system can be consider a qubit and there is a wide variety of experimental implementations ranging from superconductors, to nuclear spins, to polarized light. Each qubit lives in a two dimensional Hilbert space and this space is enlarged through the use of tensor products. For finite Hilbert spaces the matrix representation of the tensor product is the Kronecker product. That is for an n by m matrix A and a p by q matrix B , A tensor B is given by an np by mq matrix as follows:

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{bmatrix}.$$

The elements of the Hilbert space are kets which we write as $|\cdot\rangle$. The ket represents a state without specifying the basis we will write it in. Elements of the conjugate space are called the bras and a bra is written as $\langle\cdot|$. The inner product of a bra, $\langle b|$ and a ket, $|k\rangle$ is written as $\langle b|k\rangle$. The outer product of the two elements is then $|k\rangle\langle b|$. Once a matrix representation has been chosen say \mathcal{R}^3 then state $|s\rangle$ becomes

$$|s\rangle = \begin{bmatrix} \langle x|s\rangle \\ \langle y|s\rangle \\ \langle z|s\rangle \end{bmatrix},$$

which follows since the projector onto axis \hat{K} is given by $|k\rangle\langle k|$.

In the two dimensional Hilbert space of a single qubit, we label the upper and lower eigenstates of σ_z as $|0\rangle$ and $|1\rangle$ ². This is called the computational basis, and matrix representation of operators and states are written in this basis unless otherwise stated. A universal set of quantum circuit elements is listed in table 1.

A perfect qubit is decoupled from the environment such that the Schrödinger equations (SE) governs the evolution and only unitary operations are allowed. Non-unitary effects are typically the result of the environment and are grouped into the term decoherence. Usually the effects of the environment are assumed negligible and left as a problem for the experimentalist. Unless otherwise stated this document we will do the same. This restriction on the operations manifest itself in the circuit model of quantum computing by requiring that all circuit elements are unitary with the sole exception of measurement. Note that unitary operations always have an inverse (namely their hermitian conjugate) which implies that computations performed by quantum computer are reversible. Classically, Bennett showed that all classical circuits could be made reversible using extra bits (called ancilla bits) as scratch bits[4].

To show how these ideas lead to increase computational power, let us turn to an example of a quantum algorithm.

²The choice of \hat{Z} as the computational basis is arbitrary





Gate name	Circuit representation	Matrix representation
Hadamard		$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Single qubit rotation ($k = x, y, z$)		$\exp(-i\sigma_k(\theta/2))$
CNOT		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
Measurement		Input $ \psi\rangle$ yields classical bit v with probability $ \langle v \psi\rangle ^2$
Pauli spin- $\frac{1}{2}$ variable	Matrix representation	
σ_0	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	
σ_x	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	
σ_y	$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$	
σ_z	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	

Table 1: This set of gates is universal for unitary quantum computation as well as the Pauli spin variables. All matrix representations are in the σ_z basis and single line represent qubits. The double lines represent classical bits. Note that $|v\rangle$ in the computational basis is given by $v_{comp}^T = [\langle 0|v\rangle \quad \langle 1|v\rangle]$ where v_{comp} is a column vector.

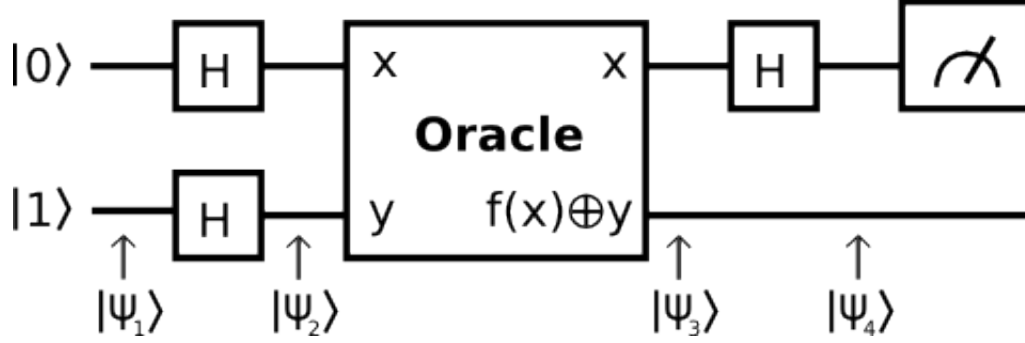


Figure 3: Deutsch’s algorithm is used to determine if a function computed by the oracle is balance or unbalanced. This algorithm requires only one query due to quantum parallelism. In the classical case, two calls to the oracle are always required.

6.1 Deutsch’s algorithm: first illustration of the power of quantum computers

Deutsch’s algorithm is a straight-forward application of quantum computing that provides a speedup over classical computation. The problem it solves can be posed as follows: given a function $f : \{0, 1\} \rightarrow \{0, 1\}$ we want to know if the function is balanced or unbalanced. If $f(x)$ is balanced then $f(0) = f(1)$. If $f(0) \neq f(1)$ then $f(x)$ is unbalanced. The function f is implemented in Deutsch’s algorithm using an oracle.

We will introduce the notion of an oracle. An oracle is black box whose workings are not important but evaluate a useful function; in our case f . For example, consider the problem of factoring a large number, say N . In this example, the oracle could be considered as two numbers and the oracle would just perform multiplication and return one if the product is N and zero otherwise. Some problems may have a very complicated oracle that is ‘expensive’ to access. Any problem that requires the use of the oracle is optimized as to reduce the number of time you’ll have to visit.

For Deutsch’s algorithm, if we label one qubit *top* and the other *bottom* then we require an oracle that evaluates $f(x)$ by performing the following unitary transform: $|x\rangle_{top}|y\rangle_{bottom} \rightarrow |x\rangle_{top}|y \oplus f(x)\rangle_{bottom}$. In this context \oplus denotes addition modulus 2. Note that $x \rightarrow f(x)$ is not unitary if f is not bijective, therefore a single qubit oracle is disallowed in this context since we are considering balance functions. In figure 6.1, the circuit for Deutsch’s algorithm is drawn and we will examine the wave function at each point indicated in the diagram.

The algorithm is initialized with the state of the two qubits as $|0\rangle_t|1\rangle_b$. The Hadamard gates transform between the σ_z basis (the computational basis) and the σ_x basis and is written in matrix form in table 1. The eigenstates of σ_x are $(|0\rangle \pm |1\rangle)/\sqrt{2}$ and we will denote them $|X\pm\rangle$. Now, $|\psi_2\rangle = |X+\rangle|X-\rangle$. Observe the effect of the oracle on $|\psi_2\rangle$:

$$\begin{aligned}
 |\psi_3\rangle = U_o|\psi_2\rangle &= \frac{(|0\rangle + |1\rangle)_t}{\sqrt{2}} \frac{(|0\rangle - |1\rangle)_b}{\sqrt{2}} \\
 &= \frac{1}{2} U_o(|00\rangle - |01\rangle + |10\rangle - |11\rangle)_{tb} \\
 &= \frac{1}{2} (|0\rangle_t |f(0) \oplus 0\rangle_b - |0\rangle_t |f(0) \oplus 1\rangle_b + |1\rangle_t |f(1) \oplus 0\rangle_b - |1\rangle_t |f(1) \oplus 1\rangle_b) \\
 &= \frac{|0\rangle_t}{2} (|f(0)\rangle - |\neg f(0)\rangle)_b + \frac{|1\rangle_t}{2} (|f(1)\rangle - |\neg f(1)\rangle)_b \tag{23}
 \end{aligned}$$

Here, $\neg x$ is the negation of x e.g. if $x = 1$ then $\neg x = 0$.

Suppose the function is balanced and let $c := f(0) = f(1)$. In this case we have $|\psi_3\rangle = (1/2) (|0\rangle + |1\rangle)_t (|c\rangle - |\neg c\rangle)_b$. Considering each value of c shows that we can write

$$|\psi_3\rangle = (-1)^c |X+\rangle|X-\rangle.$$

The unbalanced case, have that $|f(0)\rangle - |\neg f(0)\rangle = |\neg f(1)\rangle - |f(1)\rangle$. We define $c' \equiv f(0)$ and immediately get $|f(0)\rangle - |\neg f(0)\rangle = (-1)^{c'}|-\rangle$. Substituting this into (23),

$$|\psi_3\rangle = \frac{|0\rangle_t - |1\rangle_t}{2} (-1)^{c'}|-\rangle_b = (-1)^{c'}|X+\rangle_t|X-\rangle_b.$$

Applying the final Hadamard gate to the top qubit reveals:

$$|\psi_4\rangle = \begin{cases} (-1)^{c'}|0\rangle|X-\rangle & \text{balanced} \\ (-1)^{c'}|1\rangle|X-\rangle & \text{unbalanced} \end{cases}$$

The value of c'' is not important as the expectation value of an observable is all that can be measured thus global phases are not important [5]. If the first qubit is measured in the σ_z basis as zero then the function is balanced; if it is one then the function is unbalanced.

Thus, this algorithm only takes one call to the oracle whereas the classical algorithm must make two calls. This is a rather simple example but it shook the theory of computation at its core. Computer science and its algorithm based analyses was based around the Turing machine which due to the Church-Turing thesis dictates what is computable. But quantum computing began when it was realized that the physical laws that the computer obeys changes the limitations placed on its computing power. The importance of Deutsch's algorithm was not practical applications of quantum computing but rather its implications on computation. Since the introduction of this algorithm in 1985 [10] there have been other algorithms introduced of theoretical and of practical importance.