Quantum Algorithms and the Fourier Transform

Richard Jozsa
School of Mathematics and Statistics
University of Plymouth
Plymouth, Devon PL4 8AA, England.
Email: rjozsa@plymouth.ac.uk

Abstract: *The quantum algorithms of Deutsch, Simon and Shor are described in a way which highlights their dependence on the Fourier transform. The general construction of the Fourier transform on an Abelian group is outlined and this provides a unified way of understanding the efficacy of the algorithms. Finally we describe an efficient quantum factoring algorithm based on a general formalism of Kitaev and contrast its structure to the ingredients of Shor's algorithm.*

Introduction

The principal quantum algorithms which provide an exponential speedup over any known classical algorithms for the corresponding problems are Deutsch's algorithm [2], Simon's algorithm [4] and Shor's algorithm [5]. Each of these rests essentially on the application of a suitable Fourier transform. In this paper we will outline the construction of the Fourier transform over a general (finite) Abelian group and highlight its origin and utility in the quantum algorithms. This provides a unified way of understanding the special efficacy of these algorithms. Indeed we have described elsewhere [8] how this efficacy may be explicitly seen as a property of quantum entanglement in the context of implementing the large unitary operation which is the Fourier transform.

From our general group-theoretic viewpoint we will see that Simon's and Shor's algorithms are essentially identical in their basic formal structure differing only in the choice of underlying group. Both algorithms amount to the extraction of a periodicity relative to an Abelian group G using the Fourier transform of G in a uniform way. This general viewpoint may also be useful in developing new quantum algorithms by applying the formalism to other groups.

Kitaev [7] has recently formulated a group–theoretic approach to quantum algorithms. We will describe below a special explicit case of his general formalism – an efficient quantum factoring algorithm which appears to be quite different from Shor's. In particular, the Fourier transform as such, is not explicitly used. It is especially interesting to contrast (rather than align!) Shor's and Kitaev's algorithms as this may provide a new method – in addition to the ubiquitous Fourier transform – for constructing quantum algorithms. The quantum searching algorithm of Grover [9] is also based on the Fourier transform but is of a different character from those mentioned above and we will not discuss it here.

Some Notation

We will write $B = \{0, 1\}$ for the additive group of integers mod 2 and denote by $\mathcal{B}$ the Hilbert space of one qubit (i.e. a 2 dimensional Hilbert space) equipped with a standard basis denoted by $\{|0\rangle, |1\rangle\}$. $\mathcal{B}^n$ will denote the Hilbert space of n qubits. The dual basis of $\mathcal{B}$ denoted by $\{|0'\rangle, |1'\rangle\}$ is defined by

$$|0'\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \qquad |1'\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{1}$$

H will denote the fundamental unitary matrix

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{2}$$

Thus $H^2 = I$ and H interchanges the standard and dual bases. In terms of real geometry the dual basis lies on the 45° lines between the orthogonal directions $|0\rangle$ and $|1\rangle$ and H is the transformation given by reflection in a line at angle $\pi/8$ to the $|0\rangle$ direction. Thus the eigenvectors of H (parallel and perpendicular to the mirror line) are $\cos\frac{\pi}{8}|0\rangle \pm \sin\frac{\pi}{8}|1\rangle$ belonging to $\lambda = \pm 1$ respectively. We will see later that H is also the Fourier transform on the group B.

The elements of B^n are n bit strings. If $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$ are in B^n then we write

$$x \oplus y = (x_1 \oplus y_1, \ldots, x_n \oplus y_n) \in B^n$$

$$x \cdot y = (x_1 y_1 \oplus \cdots \oplus x_n y_n) \in B$$

(the operations on the RHS's being addition and multiplication mod 2 in B.) Note that $x \cdot y$ is the parity of the number of places where x and y both have a bit value of 1.

Early Days

The earliest quantum algorithms [1, 2] were concerned with a situation in which we are given a "black box" or oracle that computes a function $f : B^n \rightarrow B$ and we are required to decide whether a certain "global" property (i.e. a joint property of all the function values) holds of f. For quantum computation the black box is given as a unitary transformation $\mathcal{U}_f$ on $n+1$ qubits given in the standard basis by

$$\mathcal{U}_f : \underbrace{|x_1\rangle |x_2\rangle \ldots |x_n\rangle}_{\text{input}} |y\rangle \longrightarrow |x_1\rangle |x_2\rangle \ldots |x_n\rangle |y \oplus f(x_1, \ldots, x_n)\rangle \tag{3}$$

(We will often abbreviate $|x_1\rangle |x_2\rangle \dots |x_n\rangle$ as $|x\rangle$ for $x \in B^n$.) Thus if y is initially set to 0 the value of f may be read from the last qubit.

For our first problem, referred to as Deutsch's XOR problem [1], we have $n = 1$ so that f is one of the four possible functions $f : B \to B$. We are to decide whether $f(0) \oplus f(1)$ is 0 or 1. Equivalently we wish to decide whether f is a constant function or a "balanced" function (where balanced means that f takes one value 0 and one value 1). Clearly any classical computer requires evaluating f twice to decide this. According to Deutsch's original method [1], the problem may be solved on a quantum computer after running $\mathcal{U}_f$ only *once* but the algorithm succeeds only with probability $\frac{1}{2}$ (and we know when it has been successful). The method is simply to run $\mathcal{U}_f$ on the input superposition $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ yielding the state $\frac{1}{\sqrt{2}}(|0\rangle |f(0)\rangle + |1\rangle |f(1)\rangle)$. Writing this state in the dual basis we have the four possibilities given by the two constant functions:

$$\frac{1}{\sqrt{2}}(|0\rangle |f(0)\rangle + |1\rangle |f(1)\rangle) = \begin{cases} \frac{1}{\sqrt{2}}(|0\rangle |0\rangle + |1\rangle |0\rangle) & = & \frac{1}{\sqrt{2}}(|0'\rangle |0'\rangle + |0'\rangle |1'\rangle) \\ \frac{1}{\sqrt{2}}(|0\rangle |1\rangle + |1\rangle |1\rangle) & = & \frac{1}{\sqrt{2}}(|0'\rangle |0'\rangle - |0'\rangle |1'\rangle) \end{cases}$$

and the two balanced functions:

$$\frac{1}{\sqrt{2}}(|0\rangle |f(0)\rangle + |1\rangle |f(1)\rangle) = \begin{cases} \frac{1}{\sqrt{2}}(|0\rangle |0\rangle + |1\rangle |1\rangle) & = & \frac{1}{\sqrt{2}}(|0'\rangle |0'\rangle + |1'\rangle |1'\rangle) \\ \frac{1}{\sqrt{2}}(|0\rangle |1\rangle + |1\rangle |0\rangle) & = & \frac{1}{\sqrt{2}}(|0'\rangle |0'\rangle - |1'\rangle |1'\rangle) \end{cases}$$

Now measure the second qubit in the dual basis. If the result is $0'$ (which occurs with probability $\frac{1}{2}$ in every case) then we have lost all the information about the function f. If the result is $1'$ then measurement of the first qubit will reliably distinguish between constant and balanced functions.

In our second algorithm [2], referred to as Deutsch's algorithm, we are given n and a function $f : B^n \to B$. It is promised that f is either constant or balanced (where balanced means that f takes values 0 and 1 an equal number of times i.e. 2^{n-1} times each). The problem is to decide whether f is balanced or constant. The method, described in detail in [2], involves running $\mathcal{U}_f$ *twice* (and using H $O(n)$ times) to construct the state

$$|f\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in B^n} (-1)^{f(x)} |x\rangle \tag{4}$$

Then $|f\rangle$ for any constant function is orthogonal to the corresponding state for any balanced function and thus we can solve our decision problem with certainty by a suitable measurement on the resulting state. The quantum algorithm always runs in time $O(n)$ whereas any classical algorithm (which gives the result with certainty in every case) will require time of $O(2^n)$ at least in some cases.

Note that Deutsch's XOR problem is the $n = 1$ case of the above decision problem. However the above algorithm, running $\mathcal{U}_f$ twice, offers no advantage over the obvious classical algorithm for $n = 1$. Another distinction between the above two algorithms is that the XOR problem is solved only with probability $1/2$ whereas the second algorithm is *always* succesful. An interesting recent innovation [11] fully unifies and considerably improves the above two algorithms: the XOR problem may be solved with *certainty* and the state in eq. (4) may be constructed by running $\mathcal{U}_f$ only *once*. The improved XOR algorithm is then precisely the $n = 1$ case of the improved Deutsch algorithm. The basic idea is to set the output register to the state $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ before applying $\mathcal{U}_f$. Note that by eq. (3)

$$\mathcal{U}_f : |x\rangle\,(|0\rangle - |1\rangle) \longrightarrow \begin{cases} \;\;\,|x\rangle\,(|0\rangle - |1\rangle) & \text{if } f(x) = 0 \\ -\,|x\rangle\,(|0\rangle - |1\rangle) & \text{if } f(x) = 1 \end{cases}$$

Thus

$$\mathcal{U}_f : \frac{1}{\sqrt{2^n}} \sum_{x \in B^n} |x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \longrightarrow \left(\frac{1}{\sqrt{2^n}} \sum_{x \in B^n} (-1)^{f(x)}\,|x\rangle \right) \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right)$$

giving the state $|f\rangle$ in the first n qubits after only one application of $\mathcal{U}_f$. The last qubit plays a curiously passive role in that its state is unchanged in the process. (This is reminiscent of the similarly passive role of the second register in Shor's algorithm [6, 5]).

The explicit description of the measurement on $|f\rangle$ which distinguishes balanced from constant functions is significant for subsequent developments. We first apply the operation H to each of the n qubits of $|f\rangle$. Denoting the resulting n-qubit operation by H_n we have, for each $x \in B^n$

$$H_n : |x\rangle \to \frac{1}{\sqrt{2^n}} \sum_{y \in B^n} (-1)^{x \cdot y}\,|y\rangle \tag{5}$$

Note that

$$H_n\,|0 \ldots 0\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in B^n} |y\rangle$$

is the equal superposition of all the standard basis states and that up to an overall sign this coincides with $|f\rangle$ for f constant. Since $H_n H_n = I$ it follows that $H_n\,|f\rangle = |0 \ldots 0\rangle$ for f constant. Thus if f is balanced then $H_n\,|f\rangle$ must be orthogonal to $|0 \ldots 0\rangle$ i.e. $|f\rangle$ lies in the span of $\{|x\rangle : x \neq 0 \ldots 0\}$. Hence to distinguish balanced from constant functions we apply H_n to $|f\rangle$ and then read the bits to see whether they are all zero or not.

The above measurement has 2^n natural outcomes (i.e. all n-bit strings) and we may ask if there are special balanced functions which yield with certainty the other outcomes $x \in B^n$ in the same way that constant functions lead to the outcome $0\dots0$. For each $k \in B^n$ consider the function $f_k : B^n \to B$ given by

$$f_k(x) = k \cdot x$$

It is easily verified that each f_k is a balanced function for $k \neq 0\dots0$ (giving a small subset of all possible balanced functions). We will see later that the operation H_n is the Fourier transform on the additive group B^n (also known as the Walsh or Hadamard transform) and the functions f_k are the Fourier (Walsh, Hadamard) basis functions. For these functions we have

$$H_n \left| f_k \right\rangle = \left| k \right\rangle$$

which follows readily by comparing eq. (4) with eq. (5) and the fact that $H_n H_n = I$. Thus our quantum algorithm can reliably distinguish the 2^n functions f_k after evaluating the function only *once*! However this finer use of the measurement outcomes does not represent an exponential advantage over classical computation since the classical evaluation of just n values of f_k on the inputs $10\dots0$, $010\dots0$, up to $0\dots01$ will successively reveal the n bits of k.

A significant feature of the problem of distinguishing balanced from constant functions is the following: if we tolerate any (arbitrarily small) non-zero probability of error in the result then we lose the exponential advantage of the quantum algorithm over classical algorithms. Indeed given any ϵ, if we sample $O(-\log\epsilon)$ random values of f then we can determine within error probability ϵ whether f is balanced or constant by just claiming "constant" if all the sampled values are the same. However the 1 versus n gap between the quantum and classical identification of f_k described above persists even if we tolerate a small probability of error in the result. This led Bernstein and Vazirani [3] to amplify this gap to a super–polynomial size by a recursive procedure, leading to the first example of a problem which could be solved exponentially faster by a quantum algorithm than by any classical algorithm even if a small probability of error is tolerated. Soon thereafter Simon [4] gave a simpler example. Below we will describe the structure of Simon's algorithm and Shor's algorithm emphasising their similarity, which will lead naturally to the general concept of the Fourier transform on an Abelian group.

Simon's Algorithm

We are given a "black box" (or oracle) which computes a function $f : B^n \to B^n$. The function is promised to be a 2-to-1 function and have periodicity $\xi \in B^n$ i.e.

$$f(x) = f(y) \quad \text{iff } y = x \oplus \xi \quad \text{for all } x, y \in B^n \tag{6}$$

Our problem is to find ξ efficiently (i.e. in poly(n) steps, each evaluation of the function counting as one step). More precisely, the function is given as a unitary transformation U_f on $\mathcal{B}^{2n}$ defined by

$$U_f : |x\rangle\,|y\rangle \to |x\rangle\,|y \oplus f(x)\rangle\,.$$

Simon's algorithm (omitting normalisation factors) is the following:

Step 1. Start with the state $|0\ldots 0\rangle \in \mathcal{B}^n$ and apply H_n to get $\sum_x |x\rangle$.

Step 2. Apply U_f to $(\sum |x\rangle)\,|0\rangle$ to get $\sum |x\rangle\,|f(x)\rangle$.

Step 3. Measure the value of register 2 and keep the corresponding state of register 1. By eq. (6) the state of register 1 will have the form $|x_0\rangle + |x_0 \oplus \xi\rangle$ where $x_0 \in B^n$ has been chosen equiprobably.

Remark. Thus we have set up a state involving a periodic superposition of $|x_0\rangle$ and $|x_0 \oplus \xi\rangle$ (noting that $x_0 \oplus \xi \oplus \xi = x_0$ etc.) This contains the desired information of ξ together with an unwanted randomly chosen x_0. A direct measurement of the label would yield any $x \in B^n$ equiprobably, providing no information at all about ξ.

Step 4. Apply H_n to get (c.f. eq. (5))

$$\sum_{y \in B^n} \left((-1)^{x_0 \cdot y} + (-1)^{(x_0 \oplus \xi)\cdot y} \right) |y\rangle = \pm \sum_{y:\, y\cdot\xi = 0} |y\rangle$$

(where the overall sign depends on x_0). Note that if $y \cdot \xi = 1$ then the terms on the LHS will interfere destructively.

Remark. The effect of H_n here is to wash out the unwanted x_0 from the labels and to invert the information of ξ, recoding it as y such that $y \cdot \xi = 0$. A direct measurement of the label will now yield information about ξ. The same formal features will arise in Shor's algorithm below.

Step 5. Measure the register to find a value of y (equiprobably) such that $y \cdot \xi = 0$.

Step 6. Repeat the above to find enough y_i's so that ξ may be determined by solving the linear system $y_1 \cdot \xi = 0, \ldots, y_k \cdot \xi = 0$. It may be shown that $O(n^2)$ repetitions suffice to determine ξ with any prescribed probability $p < 1$.

Shor's Algorithm

Shor's algorithm for factoring a given number N [5, 6] proceeds by solving an equivalent problem: given any y coprime to N find the order r of y mod N. (Note that if $y \leq N$ is chosen *at random* then we may use Euclid's algorithm [6] to efficiently determine whether y is coprime to N or not. If it is not coprime, then the highest common factor of y and N gives a factor of N directly.) The order r of y mod N is the least integer r such that

$$y^r \equiv 1 \bmod N$$

Let $\mathcal{Z}_n$ denote the group of integers mod n. For any q we have a function

$$f : \mathcal{Z}_q \rightarrow \mathcal{Z}_N$$

$$f(x) = y^x \bmod N$$

so that

$$f(x+r) = f(x) \qquad \text{if} \qquad x + r \leq q \tag{7}$$

Note that because of the condition $x + r \leq q$, this function is not wholly periodic on $\mathcal{Z}_q$ unless q is an exact multiple of (the unknown) r. However if q is chosen sufficiently large, then the slight spoiling of the periodicity at x near q (i.e. in one period only) will have a negligible effect. Ideally we would choose $q = \infty$ here for perfect periodicity in every case but in practice we require that q be finite.

Thus Shor's algorithm combines two separate issues: firstly the extraction of the periodicity of f and secondly, dealing with the fact that f is not *perfectly* periodic. In our description below we will focus on the first issue and assume for simplicity that q is an exact multiple of r. We will discuss this assumption and the second issue at the end.

Suppose we are given a fixed y coprime to N and we want to compute its order mod N. The unitary transformation

$$U_f : |x_1\rangle\,|x_2\rangle \rightarrow |x_1\rangle\,|x_2 + y^{x_1} \bmod N\rangle \qquad x_1 \in \mathcal{Z}_q \qquad x_2 \in \mathcal{Z}_N$$

is efficiently computable [5, 6] and will play the same role as U_f in Simon's algorithm. Shor's algorithm proceeds by the following steps which parallel exactly the steps of Simon's algorithm. DFT_q below denotes the discrete Fourier transform for integers mod q. It is defined by

$$DFT_q : |k\rangle \rightarrow \frac{1}{\sqrt{q}} \sum_{l=0}^{q-1} e^{2\pi i \frac{kl}{q}}\,|l\rangle \qquad k \in \mathcal{Z}_q \tag{8}$$

and replaces H_n in Simon's algorithm. As before we will omit normalisation factors.

Step 1. Start with the state $|0\rangle$ (in a q dimensional Hilbert space) and apply DFT_q to get $\sum_{x=0}^{q} |x\rangle$.

Step 2. Apply U_f to $(\sum |x\rangle)\,|0\rangle$ to get $\sum |x\rangle\,|y^x \bmod N\rangle$.

Step 3. Measure the value of register 2 and keep the corresponding state of register 1. This state will have the form $\sum_\lambda |x_0 + \lambda r \bmod q\rangle$, where $x_0 \in \mathcal{Z}_r$ has been chosen equiprobably.

Remark. As in Simon's algorithm a direct measurement of the label will give no information at all about r.

Step 4. Apply DFT_q. Using eq. (8) we get [6] a state of the form

$$\sum_{k\in\mathcal{Z}_r} e^{i\phi_k(x_0)} \left| k\frac{q}{r} \right\rangle$$

Remark. Note that as in Simon's algorithm the random shift x_0 has been eliminated from the labels and the information of r has been inverted as kq/r.

Step 5. Measure the register to get a multiple $c = k(q/r)$ where $k \in \mathcal{Z}_r$ has been chosen equiprobably. Thus $c/q = k/r$ where c and q are known.

Step 6. Repeat the above until we get a result corresponding to k being coprime to r. Then r is obtained by cancelling c/q down to its lowest terms. It may be shown [5, 6] that $O(\log N)$ repetitions will suffice to determine r with any prescribed probability $p < 1$.

Thus we see that Simon's and Shor's algorithms are structurally identical (in the ideal case that q is an exact multiple of r or $q = \infty$). The group B^n and the operation H_n have been replaced respectively by the group $\mathcal{Z}_q$ and operation DFT_q. We will see in the next section that these operations are just the Fourier transforms for the respective Abelian groups and the general construction of the Fourier transform will clarify their role in the preceeding algorithms.

In general q cannot be guaranteed to be a multiple of r. Let us write $q = Kr + a$ with $a < r < N$ and let $q_0 = Kr$. In step 3 of the algorithm, instead of

$$|\psi_{q_0}\rangle = \frac{1}{\sqrt{K}} \sum_{\lambda=0}^{K-1} |x_0 + \lambda r\rangle$$

we will get

$$|\psi_q\rangle = \frac{1}{\sqrt{K+1}} \sum_{\lambda=0}^{K} |x_0 + \lambda r\rangle$$

possibly containing at most one extra term (as written) if $x_0 < a$. Thus for sufficiently large K, $|\psi_{q_0}\rangle$ and $|\psi_q\rangle$ may be as close as desired. In step 4 we will apply DFT_q to $|\psi_q\rangle$ rather than DFT_{q_0} to $|\psi_{q_0}\rangle$. However $q - q_0 = a < N$ so if q is chosen sufficiently large compared to N we may expect that the two actions will result in close outcomes. In step 5 c will not be an exact multiple of q/r but will be near to such a multiple with high probability. These intuitive remarks may be formalised [5, 6] to show that a choice of q of order N^2 suffices determine r. In step 5 the fraction k/r is then uniquely determined from the suitably close rational approximation c/q by using the theory of continued fractions [6].

The Fourier Transform on an Abelian Group

Let G be a (finite) Abelian group and let $\mathcal{H}$ be a Hilbert space with an orthonormal basis $\{|g\rangle : g \in G\}$ (the "standard" basis) labelled by the elements of G. There is a natural unitary shifting action of G on $\mathcal{H}$ given by

$$h : |g\rangle \to |hg\rangle \qquad h, g \in G \tag{9}$$

Note that we use multiplicative notation for the operation in the group G and we use the same symbol (e.g. h in eq. (9) above) to denote a group element and its unitary action on $\mathcal{H}$.

Let $f : G \to X$ be a function on the group (taking values in some set X) and consider

$$K = \{k \in G : f(kg) = f(g) \text{ for all } g \in G\}$$

K is necessarily a subgroup of G called the stabiliser or symmetry group of f. It characterises the periodicity of f with respect to the group operation of G. Given a device that computes f, our aim is to determine K. More precisely we wish to determine K in time $O(\text{poly}(\log |G|))$ where $|G|$ is the size of G and the evaluation of f on an input counts as one computational step. (Note that we may easily determine K in time $O(\text{poly}(|G|))$ by simply evaluating and examining all the values of f). Further discussion of this time constraint will be given in the next section.

We begin by constructing the state

$$|f\rangle = \frac{1}{\sqrt{|G|}} \sum_{g \in G} |g\rangle \, |f(g)\rangle$$

and read the second register. Assuming that f is suitably non-degenerate – in the sense that $f(g_1) = f(g_2)$ iff $g_1 g_2^{-1} \in K$ i.e. that f is one-to-one within each period – we will obtain in the first register

$$|\psi(g_0)\rangle = \frac{1}{\sqrt{|K|}} \sum_{k \in K} |g_0 k\rangle \tag{10}$$

corresponding to seeing $f(g_0)$ in the second register and g_0 has been chosen at random.

Examples. In Simon's algorithm G is the additive group B^n and K is the cyclic subgroup $\{0, \xi\}$ generated by ξ. In Shor's algorithm G is the additive group $\mathcal{Z}_q$ and K is the cyclic subgroup $\{0, r, 2r, \ldots\}$ generated by r. In each case K is specified by giving its generator. The state (10) is obtained in step 3 of the algorithm. □

Remark. The construction leading to the state (10) applies in a more general context than just a function on a group. Suppose we have any mathematical object F with an action of the group G on it:

$$g : F \to gF \qquad \text{such that } (g_1 g_2)F = g_1(g_2 F).$$

The symmetry group of F is the subgroup $K = \{k \in G : kF = F\}$. By constructing $\sum_g |g\rangle$, applying it to a suitable state description $|F\rangle$ of F and reading the second register we obtain the state $\sum_k |g_0 k\rangle$ as in eq. (10). □

In eq. (10) we have an equal superposition of labels corresponding to a randomly chosen coset of K in G. Now G is the disjoint union of all the cosets so that if we read the label in eq. (10) we will see a random element chosen equiprobably from all of G yielding no information at all about K. The Fourier transform will provide a way of eliminating g_0 from the labels which may then provide direct information about K. We first construct a basis $|\chi_i\rangle$ of states which are *shift invariant* in the sense:

$$g\,|\chi_i\rangle = e^{\phi_i(g)}\,|\chi_i\rangle \text{ for all } g \in G$$

Such states are guaranteed to exist since the shift operations g are unitary and they all commute. Next note that the state in eq. (10) may be written as a g_0-shifted state:

$$\sum_{k \in K} |g_0 k\rangle = g_0 \left(\sum_{k \in K} |k\rangle \right)$$

Hence if we write this state in the basis $\{|\chi_i\rangle, i = 1, \ldots, |G|\}$ then $\sum_k |k\rangle$ and $\sum_k |g_0 k\rangle$ will contain the same pattern of labels, determined by the subgroup K only. The Fourier transform is simply defined to be the unitary operation which transforms the shift-invariant basis into the standard basis. After applying it to eq. (10) we may read the shift-invariant basis label by reading in the standard basis. This explains the essential role of the Fourier transform in step 4 of the algorithms.

The shift-invariant states $|\chi_i\rangle$ are constructed using some basic group representation theory [12]. Consider any (nonzero) complex valued function on the group

$$\chi : G \to \mathcal{C}$$

which respects the group operation in the sense that

$$\chi(g_1 g_2) = \chi(g_1)\chi(g_2) \tag{11}$$

For Abelian groups these are the irreducible representations [12] of G. By listing the values χ may also be viewed as a complex vector of dimension $|G|$.

For our purposes the essential properties of these functions are the following (c.f. [12] for a full discussion and proofs).

(A) Any value $\chi(g)$ is a $|G|^{\rm th}$ root of unity.

(B) Orthogonality (Schur's lemma): For each i and j

$$\frac{1}{|G|}\sum_{g\in G}\chi_i(g)\overline{\chi_j(g)}=\delta_{ij} \tag{12}$$

(where the overline denotes complex conjugation).

(C) There are always exactly $|G|$ different functions χ satisfying eq. (11).

It is remarkable that the simple condition eq. (11) has such strong consequences. In particular the orthogonality condition **(B)** entails the fact that the Fourier transform as a linear transformation is *unitary* rather than just invertible. This appears to make no significant difference for classical computation but it is crucial for quantum computation!

Since **(B)** provides the fundamental connection to quantum computation we give a simple proof of it (incorporating also **(A)**). Note that by (11) $\chi(e)=1$ where e is the identity of G. Also (by Lagrange's theorem) we have $g^{|G|}=e$ for all $g\in G$. Hence $\chi(g)$ is always a $|G|^{\rm th}$ root of unity so $\overline{\chi(g)}=\chi(g^{-1})$. Now for any χ_1, χ_2 consider:

$$\begin{aligned}
\chi_1(h)\left(\sum_{g\in G}\chi_1(g)\chi_2(g^{-1})\right) &= \sum_{g\in G}\chi_1(hg)\chi_2(g^{-1})\\
=\sum_{\tilde g\in G}\chi_1(\tilde g)\chi_2(\tilde g^{-1}h) &\qquad (\text{putting } \tilde g = hg)\\
=\left(\sum_{\tilde g}\chi_1(\tilde g)\chi_2(\tilde g^{-1})\right)\chi_2(h)&
\end{aligned} \tag{13}$$

Hence for every $h\in G$

$$(\chi_1(h)-\chi_2(h))\sum_{g\in G}\chi_1(g)\overline{\chi_2(g)}=0$$

giving orthogonality if $\chi_1\neq\chi_2$. If $\chi_1=\chi_2=\chi$ then

$$\sum_g\chi(g)\overline{\chi(g)}=\sum\chi(g)\chi(g^{-1})=\sum\chi(e)=\sum 1=|G|$$

completing the proof of (12).

For any function χ_i satisfying eq. (11) consider the state

$$|\chi_i\rangle=\frac{1}{\sqrt{|G|}}\sum_{g\in G}\overline{\chi_i(g)}\,|g\rangle$$

The orthogonality relation (12) implies that the states $\{|\chi_i\rangle : i = 1, \ldots, |G|\}$ form an orthonormal basis of $\mathcal{H}$, called the Fourier basis. Furthermore these basis states are shift-invariant in the required sense:

$$h\,|\chi_i\rangle = \chi_i(h)\,|\chi_i\rangle \qquad h \in G \tag{14}$$

which is easily verified using eqs. (9), (11) and making the same replacement as in eq. (13).

Let us choose an ordering $g_1, g_2, \ldots, g_{|G|}$ of the elements of G. The Fourier transform FT on G (with respect to the ordering) is defined to be the unitary transformation which maps $|\chi_i\rangle$ to $|g_i\rangle$. Thus in the ordered basis $\{|g_i\rangle\}$ the matrix of FT is formed by listing the values of the functions χ_i as rows:

$$[FT]_{ij} = \frac{1}{\sqrt{|G|}}\chi_i(g_j) \tag{15}$$

Examples. If $G = \mathcal{Z}_q$ then the q functions χ_k are defined by

$$\chi_k(1) = e^{2\pi i k/q} \qquad k = 0, \ldots, q-1$$

and by (11) $\chi_k(m) = \chi_k(1)^m = \exp 2\pi i\, km/q$ for all $m \in \mathcal{Z}_q$. These values scaled by $\sqrt{q}$ are the rows of the matrix of DFT_q.
For $G = B^n$ the 2^n χ functions are

$$\chi_\sigma(x) = (-1)^{x\cdot\sigma} \qquad \text{for all } x, \sigma \in B^n$$

which (scaled by $\sqrt{2^n}$) are the rows of the Hadamard transform H_n (c.f. eq. (5)).

Efficient Computation of the Fourier Transform

The Fourier transform FT on G is a unitary operation of size $|G|$. It is known [1, 6] that any unitary operation of size d may be implemented in time $O(d^2)$ but this does not suffice for our application of FT. In Simon's algorithm $|G| = 2^n$ but we want the algorithm to run in poly(n) time and in Shor's algorithm $|G| = O(q) = O(N^2)$ and we want the algorithm to run in poly$(\log N)$ time. Thus we want to implement FT in poly$(\log|G|)$ time.

In classical computation the application of a matrix of size $|G|$ requires time $O(|G|^2)$. The classical fast Fourier transform (FFT) algorithm (applicable to certain groups) improves this to $O(|G|\log|G|)$ but this, in itself, does not suffice for our quantum algorithms since it is still exponetial in $\log|G|$. It may be seen that in a quantum context the implementation of the FFT algorithm combines with extra non-classical properties of entanglement to provide an algorithm which runs in $O(\text{poly}(\log|G|))$ time. This feature has been elaborated in [8] and is also discussed in [13].

Kitaev's Algorithm

An approach to the construction of quantum algorithms based on group-theoretic principles (for Abelian groups) has recently been developed by Kitaev [7]. We describe here an explicit example of his general formalism – an alternative efficient quantum factoring algorithm. This algorithm, in contrast to Shor's, does not explicitly require the Fourier transform to be performed and appears to be based on different principles.

Kitaev's algorithm, like Shor's, proceeds by finding the order r of a number y coprime to N. Let $U : \mathcal{H}_N \to \mathcal{H}_N$ be the unitary operator on an N dimensional Hilbert space given by "multiplication by y" (easily implementable efficiently):

$$U : |m\rangle \mapsto |my \bmod N\rangle \qquad m = 0, \ldots, N-1 \tag{16}$$

Thus we will be focussing on the *multiplicative* structure of the integers mod N (rather than the additive structure) and working in a Hilbert space of dimension N. We do not need to choose a $q \approx O(N^2)$ as in Shor's algorithm and the associated complications of q not being an exact multiple of r do not arise.

Since $U^r = I$ we see that the eigenvalues of U are r^{th} roots of unity i.e. $\lambda_k = \exp(-2\pi i k/r), k = 0, \ldots, r-1$. It is straightforward to verify that the following states $|\lambda_k\rangle$ are eigenstates of U belonging respectively to the eigenvalues λ_k:

$$|\lambda_k\rangle = \frac{1}{\sqrt{r}} \sum_{l=0}^{r-1} \exp(2\pi i \frac{lk}{r}) \left| y^l \bmod N \right\rangle \qquad k = 0, \ldots, r-1 \tag{17}$$

and that

$$|1\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} |\lambda_k\rangle \tag{18}$$

Remark. The fact that (17) are eigenstates of U is closely related to our previous construction of shift invariant states. Indeed the multiplicative group of powers of y mod N is isomorphic to the additive group $\mathcal{Z}_r$ (where we associate y^l with $l \in \mathcal{Z}_r$). Under this isomorphism the operation U becomes the shift operation of "adding 1" in $\mathcal{Z}_r$. Then (17) gives precisely the shift invariant states of $\mathcal{Z}_r$ but written with multiplicative labels y^l mod N rather than the additive labels $l \in \mathcal{Z}_r$.

Eq. (18) is simply derived by noting that each $|\lambda_k\rangle$ in (17) contains $|1\rangle$ with amplitude $1/\sqrt{r}$. Hence the sum in (18) contains $|1\rangle$ with amplitude 1 so that all other $|k\rangle$'s with $k \neq 1$ must have amplitude 0 as (18) is a normalised state. This equation also has a group-theoretic origin. It may be shown [12] that for any group G if we sum all the χ_i functions we get:

$$\frac{1}{|G|} \sum_{i=1}^{|G|} \chi_i(g) = \begin{cases} 1 \text{ if } g = e \\ 0 \text{ if } g \neq e \end{cases}$$

Then (18) follows immediately using the above interpretation of $|\lambda_k\rangle$ as shift invariant states.$\Box$

Suppose now that we have an efficient procedure for measuring the eigenvalues of a unitary operator. More precisely, given a quantum device which computes an n-qubit operation U and an eigenstate $|\lambda\rangle$ of U, suppose that we can compute the value of λ efficiently i.e. in time $O(\mathrm{poly}(\log n))$. Suppose furthermore that on an input *superposition* of eigenstates $\sum a_k |\lambda_k\rangle$ the procedure returns some one of the eigenvalues λ_k with probability $|a_k|^2$. Then applying this procedure to U and the state $|1\rangle$ above, we will be able to efficiently find a value of k/r chosen equiprobably for $k = 0, \ldots, r-1$. As in Shor's analysis, this suffices to factor N efficiently. It is remarkable that the apparently humdrum state $|1\rangle$ (when viewed appropriately as in eq. (18)) contains the information to factorise *any* given number!

How to Measure the Eigenvalues of U [7]

Suppose we are given a "black box" which computes $U : \mathcal{B}^n \to \mathcal{B}^n$, a unitary operation on n qubits, and also an eigenstate $|\lambda\rangle$ of U with $\lambda = \exp 2\pi i\phi$. We want to measure ϕ. The basic idea is to set up a state $|\alpha\rangle = \sqrt{p_0}\,|0\rangle + \sqrt{p_1}\,|1\rangle$ whose amplitudes depend on ϕ. Then by sufficiently many measurements on copies of $|\alpha\rangle$ we can estimate the probabilities p_0, p_1 and hence ϕ.

We first describe how to implement $\Lambda(U)$, the "controlled-U" operation on $n+1$ qubits (which includes one "control qubit").
Let $\tau : \mathcal{B}^{2n} \to \mathcal{B}^{2n}$ on two n-qubit registers $\mathcal{X}, \mathcal{Y}$ be the addition of n-bit strings:

$$\tau : |x\rangle\,|y\rangle \mapsto |x\rangle\,|x \oplus y\rangle \qquad x, y \in \mathcal{B}^n$$

Let $\Lambda(\tau) : \mathcal{B}^{2n+1} \to \mathcal{B}^{2n+1}$, on a 1-qubit control register $\mathcal{C}$ with $\mathcal{X}$ and $\mathcal{Y}$, be the controlled τ operation:

$$\Lambda(\tau) : |0\rangle\,|x\rangle\,|y\rangle \mapsto |0\rangle\,|x\rangle\,|y\rangle \qquad \Lambda(\tau) : |1\rangle\,|x\rangle\,|y\rangle \mapsto |1\rangle\,|x\rangle\,|x \oplus y\rangle$$

Similarly let $\Lambda(U) : \mathcal{B}^{n+1} \to \mathcal{B}^{n+1}$ on registers $\mathcal{C}$ and $\mathcal{X}$ be the controlled-U operation:

$$\Lambda(U) : |0\rangle\,|x\rangle \mapsto |0\rangle\,|x\rangle \qquad \Lambda(U) : |1\rangle\,|x\rangle \mapsto |1\rangle\; U\,|x\rangle$$

Let N be the operation of negation in the register $\mathcal{C}$.

Suppose that $U\,|0\rangle = |0\rangle$. Then $\Lambda(U)$ can be implemented as follows. In addition to the $n-$qubit register $\mathcal{X}$ of U we introduce a 1–qubit control register $\mathcal{C}$ and an extra $n-$qubit register $\mathcal{Y}$. Consider the sequence of operations (reading from left to right) in which the square brackets denote the registers to which the operations are applied:

$$N[\mathcal{C}]\;\; \Lambda(\tau)[\mathcal{C},\mathcal{X},\mathcal{Y}]\;\; \Lambda(\tau)[\mathcal{C},\mathcal{Y},\mathcal{X}]\;\; U[\mathcal{X}]\;\; \Lambda(\tau)[\mathcal{C},\mathcal{Y},\mathcal{X}]\;\; \Lambda(\tau)[\mathcal{C},\mathcal{X},\mathcal{Y}]\;\; N[\mathcal{C}]$$

If $\mathcal{Y}$ is initially set to $|0\rangle$ then after these operations $\mathcal{Y}$ will again be $|0\rangle$ and $\Lambda(U)$ will have been effected on the registers $[\mathcal{C}, \mathcal{X}]$. This is readily seen by a straightforward calculation. The $\Lambda(\tau)$ operations on either side of $U[\mathcal{X}]$ simply serve to swap the states of the registers $\mathcal{X}$ and $\mathcal{Y}$. Thus if $\mathcal{C}$ is $|0\rangle$ the states in $\mathcal{X}$ and $\mathcal{Y}$ are swapped and U is merely applied to $|0\rangle$. If $\mathcal{C}$ is $|1\rangle$ then the states are not swapped and U is applied to the original contents of $\mathcal{X}$.

To measure ϕ consider the following procedure **PROC**:
Start with registers $[\mathcal{C}, \mathcal{X}]$ in state $|0\rangle|\lambda\rangle$. Apply H to $\mathcal{C}$, then $\Lambda(U)$ to $[\mathcal{C}, \mathcal{X}]$, then H to $\mathcal{C}$ again. This results in the following state in $[\mathcal{C}, \mathcal{X}]$:

$$|\psi_\lambda\rangle|\lambda\rangle = \left(\frac{1}{2}(1+\exp 2\pi i\phi)|0\rangle + \frac{1}{2}(1-\exp 2\pi i\phi)|1\rangle\right)|\lambda\rangle$$

Note that the eigenstate in $\mathcal{X}$ has not been corrupted and may be used again. Finally measure the control register. This will yield 0 or 1 with probability distribution $\mathcal{P}$ given by:

$$p_0 = \frac{1}{2}(1+\cos 2\pi\phi) \qquad p_1 = \frac{1}{2}(1-\cos 2\pi\phi)$$

To get the information of ϕ we just repeat **PROC** for many independent control qubits, sampling the distribution $\mathcal{P}$ sufficiently many times to get an adequate estimate of p_0. Suppose we apply **PROC** t times successively, starting with t control qubits and ending in the state $|\psi_\lambda\rangle|\psi_\lambda\rangle\ldots|\psi_\lambda\rangle|\lambda\rangle$ and then sample $\mathcal{P}$ t times. Let y be the number of times that outcome "0" occurs. Then by the weak law of large numbers, for any $\delta > 0$

$$\text{Prob}\left(|\frac{y}{t} - p_0| > \delta\right) \leq \frac{2}{\sqrt{2\pi}} \exp\left(-\frac{\delta^2 t}{2p_0p_1}\right) \equiv \epsilon \tag{19}$$

Thus with t repetitions we can measure p_0 (i.e. ϕ) to precision δ with error probability ϵ. Note that for fixed δ the error probability ϵ decreases exponentially with t i.e. $t = O(\log(1/\epsilon)$ but the precision δ (for fixed ϵ) cannot be efficiently improved – for each extra bit of precision, $\delta \to \delta/2$, we require $t \to 4t$ in (19) to maintain a constant level of ϵ. Hence by this direct method, the number of bits of precision can be improved only by a correspondingly *exponential* increase in computing effort – $O(4^l)$ steps for l bits of precision. This is unacceptable. To get around this difficulty let us suppose that not only U is efficiently computable (i.e. in poly(n) steps) but also that:

$$\textbf{Assumption}\text{: } U^{(2^j)} \text{ can be computed in poly}(j,n) \text{ steps} \tag{20}$$

This assumption is valid in our application of U being "multiplication by y". U^{2^j} is then "multiplication by y^{2^j}" which can be implemented by a sequence of j repeated squarings, starting with y. It will not, however, be valid for a general unitary transformation U.

Now assuming (20) we can efficiently improve the precision δ as follows i.e. obtain l bits of p_0 with computing effort poly(l). Note that $|\lambda\rangle$ is an eigenstate of U^{2^j} with eigenvalue $\exp(2\pi i[2^j\phi \bmod 1])$. To obtain l bits of ϕ with error probability $\leq \epsilon$ we measure (as above) the values of $2^j\phi \bmod 1$ for $j = 0, \ldots, l-1$, to a fixed precision $\delta = 1/8$ with error probability $\leq \epsilon/l$. Now if we write ϕ in binary then $2^j\phi$ has the point shifted j places to the right and "mod 1" removes the integer part. Thus knowing $2^j\phi \bmod 1$ to $\pm\frac{1}{8}$ gives the first few bits of $2^j\phi \bmod 1$ i.e. bits j and $j+1$ of ϕ itself. Hence we get about l bits of precision of ϕ. The probability that all these bits are correct exceeds $(1-\epsilon/l)^l \geq 1-\epsilon$. This completes the efficient approximation of ϕ under the assumption (20) above.

Generally (as in Kitaev's factoring algorithm) we will not have available a pure eigenstate of U but instead some superposition $\sum a_\lambda |\lambda\rangle$. If we apply **PROC** to this state with t control bits we will obtain $\sum a_\lambda |\psi_\lambda\rangle \ldots |\psi_\lambda\rangle |\lambda\rangle$ so that a measurement of the control bits will yield one of the eigenvalues λ with probabilities $|a_\lambda|^2$. i.e. if we trace out the eigenstate register $|\lambda\rangle$ the t control qubits are in a mixture of the repeated states $|\psi_\lambda\rangle \ldots |\psi_\lambda\rangle$ with probabilities $|a_\lambda|^2$. Note that we must apply **PROC** t times *before* any measurement of the control qubits is made. Otherwise each successive measurement will provide information about a different eigenvalue and finally we will only obtain information about the *average* value of the λ's weighted by $|a_\lambda|^2$, rather than about some *one* of the λ's.

In most cases the eigenvalues $exp(2\pi i\phi)$ will have *rational* values of ϕ, $\phi = a/b$. This is because the U's of interest will have finite order i.e. $U^m = I$ for some m so that $\phi = k/m$ for some k. For example if U is "multiplication by y" then $U^r = I$ so that ϕ must have the form k/r (as noted previously). In this situation we can find ϕ exactly, rather than just approximately, by choosing a suitably high precision δ. The minimum separation between any two rational numbers with denominators r is $1/r$ so we can get $\phi = k/r$ exactly by measuring it to precision $1/2r > 1/2N$ i.e. $1 + \log N$ bits.

Thus we obtain an efficient factoring algorithm based on the novel idea of determining an eigenvalue of a given simple unitary operation. Some of the formalism of Kitaev's algorithm may be related to Shor's method by using the decomposition of the Fourier transform given in [14] but it would be interesting to consider other problems that might be formulatable in terms of the determination of eigenvalues.

Conclusions

We have seen that the principal known quantum algorithms all revolve around one essential construction, that of the Fourier transform on an Abelian group. Furthermore the quantum computational speedup provided by these algorithms may be attributed to (non-classical) properties of entanglement operating within the implementation of classical fast Fourier transform algorithms on a quantum computer [8].

Clearly it would be of great interest to have other basic ingredients for the construction of new quantum algorithms. Kitaev's formalism [7] as we have illustrated, appears to involve such an ingredient. The mathematical construction of the Fourier transform also extends to non-Abelian groups and it would be interesting to investigate problems which can be formulated in terms of non-Abelian Fourier transforms and the possiblity of their implementation on a quantum computer. This line of development has also been advocated by Hoyer [13].

Acknowledgements

The author is grateful for the hospitality of the Institute for Theoretical Physics, University of California, Santa Barbara during the Quantum Computers and Quantum Coherence program during which time a preliminary version of this work was completed. This work was supported in part by the National Science Foundation under Grant No. PHY94-07194 and by the Royal Society, London.

References

[1] Deutsch, D. (1985) *Proc. Roy. Soc. London Ser. A* **400**, 97.

[2] Deutsch, D. and Jozsa, R. (1992) *Proc. Roy. Soc. London Ser A* **439**, 553-558.

[3] Bernstein, E. and Vazirani, U. (1993) *Proc. 25th Annual ACM Symposium on the Theory of Computing*, (ACM Press, New York), p. 11-20 (Extended Abstract). Full version of this paper appears in *S. I. A. M. Journal on Computing* **26** (Oct 1997).

[4] Simon, D. (1994) *Proc. of 35th Annual Symposium on the Foundations of Computer Science*, (IEEE Computer Society, Los Alamitos), p. 116 (Extended Abstract). Full version of this paper appears in *S. I. A. M. Journal on Computing* **26** (Oct 1997).

[5] Shor, P. (1994) *Proc. of 35th Annual Symposium on the Foundations of Computer Science*, (IEEE Computer Society, Los Alamitos), p. 124 (Extended Abstract). Full version of this paper appears in *S. I. A. M. Journal on Computing* **26** (Oct 1997) and is also available at LANL quant-ph preprint archive 9508027.

[6] Ekert, A. and Jozsa, R. (1996) *Rev. Mod. Phys.* **68**, 733.

[7] Kitaev, A. (1995) *"Quantum Measurements and the Abelian Stabiliser Problem"*, preprint available at LANL quant-ph preprint archive 9511026.

[8] Jozsa, R. (1997) *"Entanglement and Quantum Computation"* in *Geometric Issues in the Foundations of Science* eds. S. Huggett, L. Mason, K. P. Tod, S. T. Tsou and N. M. J. Woodhouse (Oxford University Press).

[9] Grover, L. (1996) *Proc. 28th Annual ACM Symposium on the Theory of Computing*, (ACM Press, New York), p. 212-219.

[10] Papadimitriou, C. H. (1994) *Computational Complexity* (Addison-Wesley, Reading, MA).

[11] Attributed to R. Cleve, private communication, July (1996).

[12] Fulton, W. and Harris, J. (1991) *"Representation Theory – A First Course"*, chapters 1 and 2, (Springer Verlag).

[13] Hoyer, P. (1997) *"Efficient Quantum Algorithms"*, preprint available a quant-ph/9702028.

[14] Griffiths, R. B. and Niu, C-S. (1996) *Phys. Rev. Lett.* **76**, 3228.