Quantum algorithms: entanglement–enhanced information processing

Artur Ekert and Richard Jozsa

Phil. Trans. R. Soc. Lond. A 1998 **356**, 1769-1782
doi: 10.1098/rsta.1998.0248

Quantum algorithms: entanglement-enhanced information processing

By Artur Ekert[1] and Richard Jozsa[2]

[1] *Clarendon Laboratory, University of Oxford, Parks Road, Oxford OX1 3PU, UK*
[2] *School of Mathematics and Statistics, University of Plymouth, Plymouth, Devon PL4 8AA, UK*

We discuss the fundamental role of entanglement as the essential non-classical feature providing the computational speed-up in the known quantum algorithms. We review the construction of the Fourier transform on an Abelian group and the principles underlying the fast Fourier transform (FFT) algorithm. We describe the implementation of the FFT algorithm for the group of integers modulo 2^n in the quantum context, showing how the group-theoretic formalism leads to the standard quantum network, and identify the property of entanglement that gives rise to the exponential speed-up (compared to the classical FFT). Finally, we outline the use of the Fourier transform in extracting periodicities, which underlies its utility in the known quantum algorithms.

Keywords: algorithms; quantum computational speed-up; Fourier transform; fast Fourier transform; Abelian groups; periodicity determination

1. Introduction

In 1982 Feynman noted a profound difference in the nature of physical evolution governed by the laws of quantum physics compared to evolution under the laws of classical physics. He observed that quantum mechanics (apparently) cannot be *efficiently* (Papadimitriou 1994) simulated on a classical computer (or by any classical means), i.e. that the simulation of a general quantum evolution by any classical means appears to involve an unavoidable exponential slow-down in running time. This observation embodies the essence of the subject of quantum computation, so we will begin by elaborating its meaning in terms of a simple example.

Consider a discrete sequential quantum process defined as follows. We start with a row of qubits (i.e. two level systems with a preferred basis labelled $\{|0\rangle, |1\rangle\}$) all initially in state $|0\rangle$:

$$\begin{array}{cccccc} |0\rangle & |0\rangle & |0\rangle & \cdots & |0\rangle & \ldots \\ \text{qubit } 1 & \text{qubit } 2 & \text{qubit } 3 & \ldots & \text{qubit } j & \ldots . \end{array}$$

We are also given a fixed two-qubit interaction (or two-qubit 'quantum gate') U, which is a unitary operation that may be applied to any selected pair (i, j) of qubits. Furthermore, we have a program of instructions specifying the pairs of qubits to which the gate should be sequentially applied. Thus step k of the program is 'apply U to qubits (i_k, j_k) and replace them in the row', for $k = 1, \ldots, n$. After n steps in this process we measure qubit 1 in its preferred basis, obtaining 0 or 1 according to a probability distribution $\mathcal{P}_n = \{p_n(0), p_n(1)\}$. Thus by implementing this process in an actual quantum physical system we can sample the distribution $\mathcal{P}_n$ in time $O(n)$, i.e. after a time which grows linearly with n.

Our problem is to mimic this process by classical means. More precisely, we wish to describe a classical probabilistic process which enables us to sample the distribution $\mathcal{P}_n$ defined by the above quantum process. A simple way of achieving this is the following. The operation U is just a 4×4 unitary matrix and, given the starting state with the program, we can sequentially compute by hand—using simple matrix multiplication—the quantum state at each successive stage k. Then knowing the state at stage n, the rules of quantum measurement theory enable us to calculate $p_n(0)$ and $p_n(1)$, so finally we toss a correspondingly biased coin.

This classical simulation has the following notable characteristic feature. The quantum state after k steps is generally a k-qubit entangled state requiring $O(2^k)$ coefficients for its description (as typically an extra qubit may be brought in at each step). Thus, because of entanglement, we have an exponential growth in time of the information needed to describe the state. Hence the classical simulation will slow down exponentially in time under the weight of this exponentially growing information that needs to be processed in each step. To sample $\mathcal{P}_n$, our classical simulation will require $O(2^n)$ time while the quantum process marches ahead in unflagging *linear* time. There is no more efficient classical method known to solve this problem. Thus according to the laws of quantum mechanics, remarkably nature is able to process information exponentially more efficiently than can be achieved by any classical means!

Note that if only *product* states of qubits were available then the information needed to describe the state would grow only *linearly* with n (being n times the amount of information needed to describe a typical single-qubit state). Thus the exponential speed-up in our example of quantum information processing is fundamentally a feature of quantum *entanglement*. This point has been elaborated in Jozsa (1998*a*). Indeed it provides an extraordinary manifestation of entanglement which is entirely independent of the auxiliary notion of non-locality.

We may attempt to mimic the quantum process using classical *waves* which admit the possibility of *superposition* of modes. For example, we might represent each qubit by a vibrating elastic string with fixed end points and select two lowest-energy modes of vibration to represent the states $|0\rangle$ and $|1\rangle$. It is then possible to construct the general superposition corresponding to $a|0\rangle + b|1\rangle$. However, regardless of how much the strings interact with each other in their subsequent (externally driven) vibrational evolution, their joint state is always a *product* state of n separate vibrations. The total state space of the total classical system is the *Cartesian* product of the individual state spaces of the subsystems, whereas quantum mechanically, it is the *tensor* product. This crucial distinction between Cartesian and tensor products is precisely the phenomenon of quantum entanglement. Nevertheless, we may yet attempt to represent entanglement using classical waves in the following manner. The state of n qubits is a 2^n-dimensional space and can be isomorphically viewed as the state space of a *single* particle with 2^n levels. Thus we simply interpret certain states of a single 2^n-level particle as ‘entangled’ via their correspondence under a chosen isomorphism between $\otimes^n \mathcal{H}_2$ and $\mathcal{H}_{2^n}$ (where $\mathcal{H}_k$ denotes a Hilbert space of dimension k.) In this way, 2^n modes of a classical vibrating system can apparently be used to mimic general entanglements of n qubits. However, the physical implementation of this correspondence appears always to involve an exponential overhead in some physical resource so that the isomorphism is *not* a valid correspondence for considerations of complexity, i.e. when the amount of physical resources required to achieve the representation is taken into account. For example, suppose that the 2^n levels of the one-particle quantum system, or corresponding classical system, are equally spaced energy levels. A general state of n qubits requires an amount of energy that grows *linearly* with n (since we will need, at most, to excite each qubit to its upper level), whereas a general state of the 2^n-level quantum or classical system requires an amount of energy that grows *exponentially* with n. To physically realize a system in a general superposition of 2^n modes we need exponential resources classically and linear resources quantum mechanically *because of the existence of entanglement.*

Our discussion above about the information needed to describe a state indicates that n qubits have an exponentially larger capacity to represent information than n classical bits. Note that although n classical bits have 2^n possible states, each of these states may be described by just n bits, in contrast to the quantum situation where $O(2^n)$ superposition components may be involved in a single state. However, the information embodied in the quantum state has a further remarkable feature—most of it is *inaccessible* to being read by any possible means! Indeed quantum measurement theory places severe restrictions on the amount of information that we can obtain about the identity of a given unknown quantum state. This intrinsic inaccessibility of the information may be quantified (Holevo 1973; Fuchs & Peres 1996) in terms of Shannon's information theory (Cover & Thomas 1991). In the case of a general state of n qubits, with its $O(2^n)$ information content, it turns out that at most n classical bits of information about its identity may be extracted from a single copy of the state by any physical means whatsoever. This coincides with the maximum information capacity of n classical bits.

The full (largely inaccessible) information content of a given unknown quantum state is called quantum information. Natural quantum physical evolution may be thought of as the processing of quantum information. Thus the viewpoint of computational complexity reveals a new bizarre distinction between classical and quantum physics: to perform natural quantum physical evolution, nature must process vast amounts of information at a rate that cannot be matched by any classical means, yet at the same time, most of this processed information is kept hidden from us! However, it is important to point out that the inherent inaccessibility of quantum information does *not* cancel out the possibility of exploiting this massive information processing capability for useful computational purposes. Indeed, small amounts of information may be extracted about the overall identity of the final state which would still require an exponential effort to obtain by classical means. The ability to sample the probability distribution $\mathcal{P}_n$ above provides an example. A more computationally useful example is given by the technique of 'computation by quantum parallelism' (Deutsch 1985; Deutsch & Jozsa 1992), according to which a superposition $\sum_{i=1}^{2^n}|i\rangle$ of exponentially many input values i for a function f may be set up in linear time and a *single* subsequent function evaluation will provide exponentially many function values in superposition as $|f\rangle = \sum_i |i\rangle|f(i)\rangle$. The full quantum information of this state incorporates the information of all the individual function values $f(i)$ but this is not accessible to any measurement. However, certain global properties of the collection of all the function values *may* be determined by suitable measurements on $|f\rangle$ which are not diagonal in the standard basis $\{|i\rangle|j\rangle\}$. For example, if f is a periodic function, we may determine the value of the period (Shor 1994), which falls far short of characterizing the individual function values but would generally still require an exponential number of function evaluations to obtain reliably by classical means.

2. Entanglement-enhanced information processing

Suppose that we have a physical system of n qubits in some entangled state $|\psi\rangle$ and we apply a one-qubit operation U to the first qubit. This would count as one step in a quantum computation (or rather a constant number of steps independent of n if U needs to be fabricated from other basic operations provided by the computer). Consider now the corresponding classical computation. $|\psi\rangle$ may be described in components (relative to the product basis of the n qubits) by $a_{i_1\ldots i_n}$, where each subscript is 0 or 1, and U is represented by a 2×2 unitary matrix U_i^j. The application of U corresponds to the matrix multiplication

$$a^{(\text{new})}_{i_1\cdots i_n} = \sum_j U^j_{i_1} a_{ji_2\ldots i_n}. \tag{2.1}$$

Thus the 2×2 matrix multiplication needs to be performed 2^{n-1} times, once for each possible value of the string $i_2 \ldots i_n$, requiring a computing effort which grows *exponentially* with n. On a quantum computer, because of entanglement, this 2^{n-1} repetition is unnecessary.

Consider now a unitary transformation U of n qubits (or more precisely a family of such transformations labelled by n). U may be described by a $2^n \times 2^n$ matrix and the computation of $U|\psi\rangle$ classically by direct matrix multiplication requires $O(2^n 2^n)$ operations. Even on a quantum computer U needs to be fabricated ('programmed') out of the basic operations provided by the computer, each of which operate only on some constant number of qubits. In general, U will require an exponential number of such basic operations for its implementation. It may be shown (Ekert & Jozsa 1996; Deutsch 1985) that $O(2^n 2^n)$ operations will always suffice to program U to any desired accuracy.

Suppose now that U has the following special form. Let c be any constant, independent of n. Suppose that U consists of the sequential application of $p(n)$ unitary operations V_i, $i = 1, \ldots, p(n)$, where each V_i operates on only some c out of the n qubits and $p(n)$ is a *polynomial* in n. An immediate generalization of the argument above shows that each V_i may be classically implemented (by matrix multiplication) in $O(c^2 2^{n-c}) = O(2^n)$ steps so that the classical computation of U now requires $O(p(n)2^n)$ steps. This represents an exponential saving over a general U which required $O(2^n 2^n)$ steps, but it is still exponential in n. An important example of this partial exponential speed-up for classical computation is the so-called fast Fourier transform (FTT) algorithm (Maslen & Rockmore 1995), compared to the regular Fourier transform algorithm. On a quantum computer each V_i requires some constant (independent of n) number of steps to implement (programming the c-qubit operation V_i in terms of the basic operations) so that U requires only $p(n)$ steps to implement. In summary, if U has the special form given above then it still requires exponential time to compute classically (although it *does* provide a partial exponential benefit here already) but it requires only polynomial time to compute on a quantum computer. Note, however, that after the quantum computation only a small amount of information about the transformed data is accessible to measurement, whereas the classical computation allows the full information to be accessed.

3. The super-fast quantum Fourier transform

The Fourier transform on a finite Abelian group G is a large unitary operation which arises naturally in the mathematical formalism of group representation theory. Furthermore it factorizes in the special way described in the previous section if the group has some additional structure and it is known to be a basic tool for various useful computational tasks, in particular the problem of determining periodicity. Consequently, in view of the discussion above, it can lead to quantum algorithms (Jozsa 1998*b*; Deutsch & Jozsa 1992; Bernstein & Vazirani 1993; Simon 1994; Shor 1994; Ekert & Jozsa 1996; Kitaev 1995; Grover 1996) which run substantially faster than any known classical algorithm for the corresponding computational task. In this section we will outline the construction of the Fourier transform and describe its factorization into unitary operations of a constant size.

Let $(G,+)$ be any finite Abelian group where we write the group operation in additive notation. Let $|G|$ denote the number of elements of G. An irreducible representation of G is a function

$$\chi : G \to C^*$$

(where C^* denotes the non-zero complex numbers) satisfying

$$\chi(g_1+g_2) = \chi(g_1)\chi(g_2), \tag{3.1}$$

i.e. χ is a group homomorphism from the additive group G to the multiplicative group C^*. The condition (3.1) has the following consequences (see, for example, Fulton & Harris (1991) and Jozsa (1998*b*) for proofs).

(A) Any value $\chi(g)$ is a $|G|$th root of unity. Thus χ may be viewed as a group homomorphism $\chi : G \to S^1$ where S^1 is the circle group of all unit modulus complex numbers.

(B) Orthogonality (Schur's lemma): if χ_i and χ_j are any two such functions then

$$\frac{1}{|G|}\sum_{g\in G}\chi_i(g)\overline{\chi_j(g)} = \delta_{ij} \tag{3.2}$$

(where the overline denotes complex conjugation).

(C) There are always exactly $|G|$ different functions χ satisfying equation (3.1).

In view of (C) these functions may be exhaustively labelled by the elements of G. Let $\{\chi_g : g \in G\}$ be any such chosen labelling. Then the Fourier transform on G is the $|G| \times |G|$ matrix $\mathcal{F}$ whose rows are formed by listing the values of the functions $(1/\sqrt{|G|})\chi_g$:

$$\mathcal{F}_{gk} = \frac{1}{\sqrt{|G|}}\chi_g(k), \quad g,k \in G. \tag{3.3}$$

Note that by (B) $\mathcal{F}$ is always a *unitary* matrix.

In the context of quantum computation we will have a Hilbert space $\mathcal{H}$ of dimension $|G|$ with a basis $\{|g\rangle : g \in G\}$ labelled by the elements of G. Thus there is a natural shifting action of G on $\mathcal{H}$ given by

$$U(k) : |g\rangle \to |g+k\rangle, \quad k,g \in G. \tag{3.4}$$

These operations all commute since G is Abelian so there exists a basis of simultaneous eigenstates of all the shifting operators. According to (B) the states

$$|\chi_k\rangle = \frac{1}{\sqrt{|G|}}\sum_{g\in G}\overline{\chi_k(g)}|g\rangle, \quad k \in G \tag{3.5}$$

form an orthonormal basis of $\mathcal{H}$ and using equation (3.1) we get

$$U(g)|\chi_k\rangle = \mathrm{e}^{\chi_k(g)}|\chi_k\rangle,$$

so that $\{|\chi_g\rangle : g \in G\}$ is the basis of common eigenstates of the shift operators. This basis is also called the Fourier basis. The Fourier transform $\mathcal{F}$ is a unitary operation on $\mathcal{H}$, and using equation (3.5) with (3.3) and property (B) we readily get

$$\mathcal{F}|\chi_g\rangle = |g\rangle, \tag{3.6}$$

so that the Fourier transform interchanges the standard and Fourier bases.

Let Z_q denote the additive group of integers mod q. It is well known (Fraleigh 1994) that any finite Abelian group G is isomorphic to a direct product of the form

$$G \cong Z_{m_1} \times Z_{m_2} \times \cdots \times Z_{m_r}. \tag{3.7}$$

(Furthermore we may require that m_i divides m_{i+1} and then the numbers m_i are unique.) If we assume (usually without loss of generality) that the group G is presented as a product of the form (3.7), then we can explicitly describe the irreducible representations (3.1) and obtain a canonical labelling of them by the elements of G. Suppose first that $G = Z_m$. Consider the group homomorphism given by

$$\left.\begin{aligned} \tau : G \times G &\to S^1, \\ (a, b) &\to \mathrm{e}^{2\pi\mathrm{i}(ab/m)}. \end{aligned}\right\} \tag{3.8}$$

It is easily verified that for each fixed $a \in G$ the function $\chi_a : G \to S^1$ given by $\chi_a(b) = \tau(a, b)$ satisfies equation (3.1) and there are $|G|$ such functions. Thus we have obtained an explicit formula for the irreducible representations, labelled in a natural way by the elements of G. For the general case of a product $G = Z_{m_1} \times Z_{m_2} \times \cdots \times Z_{m_r}$ we simply multiply the corresponding factors in equation (3.8), obtaining

$$\left.\begin{aligned} \tau : G \times G &\to S^1, \\ ((a_1, \ldots, a_r), (b_1, \ldots, b_r)) &\to \exp 2\pi\mathrm{i}\left(\frac{a_1 b_1}{m_1} + \frac{a_2 b_2}{m_2} + \cdots + \frac{a_r b_r}{m_r}\right), \end{aligned}\right\} \tag{3.9}$$

and again

$$\chi_{g_1}(g_2) = \tau(g_1, g_2) \tag{3.10}$$

provides the irreducible representations labelled by the elements of G.

As an example, consider the group $(Z_2)^n$ of all n-bit strings. From equations (3.9), (3.10) and (3.3) we see that the Fourier transform is just

$$\mathcal{F}_{\sigma\nu} = \frac{1}{\sqrt{2^n}}\mathrm{e}^{2\pi\mathrm{i}((\sigma\cdot\nu)/2)} = \frac{1}{\sqrt{2^n}}(-1)^{\sigma\cdot\nu},$$

where $\sigma \cdot \nu = s_1 t_1 + \cdots + s_n t_n \bmod 2$ if $\sigma = s_1 \ldots s_n$ and $\nu = t_1 \ldots t_n$. Thus in this case the Fourier transform coincides with the Hadamard (Walsh) transform. If $G = Z_{2^n}$ then we see, by using equations (3.8) and (3.10), that

$$\mathcal{F}_{ab} = \frac{1}{\sqrt{2^n}}\mathrm{e}^{2\pi\mathrm{i}((ab)/2^n)}, \quad a, b = 0, \ldots 2^n - 1,$$

giving the familiar discrete Fourier transform modulo 2^n.

As a unitary matrix the Fourier transform will act on vectors of length $|G|$. We may view any such vector as a function $f : G \to C$ on G whose list of values $f(g_1), \ldots, f(g_{|G|})$ defines the vector. The Fourier transform of f is then given by

$$\tilde{f}(k) = \sum_{g \in G} \mathcal{F}_{kg} f(g) = \frac{1}{\sqrt{|G|}} \sum_{g \in G} \chi_k(g) f(g), \quad k \in G. \tag{3.11}$$

We now describe the basic factorization property of this large unitary transformation which is necessary for its *efficient* (i.e. polynomial time) implementation in the context of quantum computation. The factorization will be carried out relative to a subgroup H of G and again the key ingredient will be the property given by equation (3.1). The basic technique was developed by Cooley & Tukey (1965) leading to the FFT algorithm in classical computation (which provides the partial exponential speed-up noted in the previous section) but the essential idea occurs already in the work of Gauss (1886).

Let H be a subgroup of G with index $I = |G|/|H|$. Let $k_1+H, k_2+H, \ldots, k_I+H$ be a complete list of the cosets of H, where $k+H \subseteq G$ denotes the subset given by $\{k+h : h \in H\}$. Thus G is partitioned as a disjoint union $(k_1+H) \cup (k_2+H) \cup \cdots \cup (k_I+H)$. Hence the elements $g \in G$ may be written in a unique way in terms of the cosets as $g = k_i + h$. Using equations (3.11) and (3.1) we get

$$\begin{aligned}\tilde{f}(l) &= \frac{1}{\sqrt{|G|}} \sum_{g \in G} f(g) \chi_l(g) = \frac{1}{\sqrt{|G|}} \sum_{i=1}^{I} \sum_{h \in H} f(k_i + h) \chi_l(k_i + h) \\ &= \frac{1}{\sqrt{|G|}} \sum_{i=1}^{I} \chi_l(k_i) \sum_{h \in H} f_i(h) \chi_l(h),\end{aligned} \tag{3.12}$$

where f_i for $i = 1, \ldots, I$ are the functions on H defined by the restrictions of f to the cosets: $f_i(h) = f(k_i + h)$. The functions χ_l restricted to the subgroup H satisfy equation (3.1) on H so they are irreducible representations of H. Hence the sum over H in equation (3.12) amounts to evaluating the Fourier transform on H of the functions f_i. Thus equation (3.12) expresses a decomposition of the Fourier transform on G into the evaluation of I Fourier transforms on H, whose results are then combined linearly in sums of length I with coefficients $\chi_l(k_i)$, done for each $l \in G$. Hence the number of operations required is

$$O(|H|^2 \times I + |G| \times I) = O(|G|(|H| + I)), \tag{3.13}$$

where we have used $I = |G|/|H|$. This is generally better than the $O(|G| \cdot |G|)$ operations for the direct (matrix multiplication) calculation of the Fourier transform on G. For example, we choose H so that I is small, say $I = 2$ giving $|H| = \frac{1}{2}|G|$ and then equation (3.13) represents an approximate halving of running time.

To enhance this benefit we iterate the construction on a *tower* of subgroups

$$G \supset H_1 \supset H_2 \supset \cdots \supset H_n \supset \{0\}$$

of greatest possible length, ultimately expressing the Fourier transform of G in terms of that on the (small) subgroup H_n. An extensive survey of this technique is given in Maslen & Rockmore (1995). We will illustrate it here only for the group Z_{2^n} and discuss the effect of the resulting decomposition on the quantum computational implementation. Z_{2^n} has an optimal tower of subgroups with each successive inclusion having the minimal possible index of 2:

$$Z_{2^n} \supset Z_{2^{n-1}} \supset Z_{2^{n-2}} \supset \cdots \supset Z_2 \supset \{0\}.$$

(Here $Z_{2^{n-1}}$ is the subgroup $\{0, 2, 4, \ldots, 2^n - 2\}$ of all even integers in Z_{2^n}, $Z_{2^{n-2}}$ is the subgroup $\{0, 4, 8, \ldots\}$ of all multiples of 4, etc., and Z_2 is the subgroup $\{0, 2^{n-1}\}$.) Consider a general position $Z_{2^m} \supset Z_{2^{m-1}}$ in this chain and let FT_{2^m} denote the Fourier transform on Z_{2^m}. The irreducible representations of Z_{2^m} are

$$\chi_j(k) = (w^j)^k, \quad \text{for } j, k = 0, \ldots 2^m - 1, \tag{3.14}$$

where $w = \exp(2\pi \mathrm{i}/2^m)$. Then equation (3.12) becomes (writing out the i-sum explicitly)

$$\begin{aligned} \tilde{f}(j) &= \sum_{k=0}^{2^m-1} f(k) \frac{\chi_j(k)}{\sqrt{2^m}} \\ &= \frac{1}{\sqrt{2}} \left(\sum_{k=0}^{2^{m-1}-1} f(2k) \frac{w^{2jk}}{\sqrt{2^{m-1}}} + w^j \sum_{k=0}^{2^{m-1}-1} f(2k+1) \frac{w^{2jk}}{\sqrt{2^{m-1}}} \right). \end{aligned} \tag{3.15}$$

Here the $f(2k)$ in the first sum and $f(2k+1)$ in the second sum give the function f restricted, respectively, to the cosets of $Z_{2^{m-1}} \subset Z_{2^m}$ (i.e. the even and odd positions in Z_{2^m}). Note that the irreducible representations of $Z_{2^{m-1}}$ are the functions given in equation (3.14) with w replaced by $w^2 = \exp(2\pi i/2^{m-1})$. Thus the two k-sums on the right-hand side of equation (3.15) are just $FT_{2^{m-1}}$ of the even and odd labelled values of f. As j in equation (3.15) runs through the values 0 to $2^m - 1$, we cycle twice through the 2^{m-1} components of the $FT_{2^{m-1}}$ (noting that $(w^2)^{2^{m-1}} = 1$). If we restrict j to running through the values 0 to $2^{m-1} - 1$ then $\tilde{f}(j)$ and $\tilde{f}(j + 2^{m-1})$ are both obtained from the jth components of the two $FT_{2^{m-1}}$ transforms on the right-hand side of equation (3.15), combined, respectively, with the coefficients $(1/\sqrt{2})(1, w^j)$ and $(1/\sqrt{2})(1, w^{j+2^{m-1}}) = (1/\sqrt{2})(1, -w^j)$. Thus equation (3.15) may be described as

$$\left.\begin{aligned} \tilde{f}(j) &= \frac{1}{\sqrt{2}}(j\text{th cpt. of } FT(f_{\text{even}}) + w^j \cdot j\text{th cpt. of } FT(f_{\text{odd}})), \\ \tilde{f}(j + 2^{m-1}) &= \frac{1}{\sqrt{2}}(j\text{th cpt. of } FT(f_{\text{even}}) - w^j \cdot j\text{th cpt. of } FT(f_{\text{odd}})), \end{aligned}\right\} \tag{3.16}$$

where f_{even} and f_{odd} refer, respectively, to the 2^{m-1} even and odd labelled values of f and j ranges from 0 to $2^{m-1} - 1$. Now if $C(2^m)$ denotes the number of operations required to (classically) compute FT_{2^m} then equation (3.15) shows that

$$C(2^m) = 2C(2^{m-1}) + O(2^m),$$

where the $O(2^m)$ arises from the extra additions and multiplications needed for the 2^m j-values in equation (3.15), to linearly combine the results of the two $FT_{2^{m-1}}$ operations. The solution of this recursion relation is

$$C(2^n) = O(n2^n),$$

giving the partial exponential speed-up (compared to $O(2^n 2^n)$) noted previously.

In the context of quantum computation, the data values $f(j)$ for $j = 0, \ldots, 2^m - 1$ reside in the amplitudes of an entangled state $|f\rangle$ of m qubits. Writing j in binary as an m-bit string we have

$$|f\rangle = \sum_{j_0, j_1, \ldots, j_{m-1}=0}^{1} |f(j_{m-1} \ldots j_1 j_0)\rangle |j_{m-1}\rangle \ldots |j_1\rangle |j_0\rangle,$$

and the qubits are numbered $0, 1, \ldots, m-1$ from right to left. The two $FT_{2^{m-1}}$ operations in equation (3.15), which operate on even and odd numbered components, respectively, may then be implemented by a *single* $FT_{2^{m-1}}$ operation on qubits $m - 1, m-2, \ldots, 1$, since the values 0 and 1 of the remaining rightmost index, respectively, determine the even and odd labelled positions (cf. the discussion of equation (2.1)). The jth component of $FT_{2^{m-1}}(f_{\text{even}})$ (respectively, $FT_{2^{m-1}}(f_{\text{odd}})$) then resides as the amplitude in dimension $2j$ (respectively, $2j + 1$). Thus to perform the linear recombination of the two $FT_{2^{m-1}}$, equation (3.16) shows that we need to:

(a) perform the unitary operation

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & w^j \\ 1 & -w^j \end{pmatrix}$$

on dimensions $(2j, 2j + 1)$ for each $j = 0, \ldots, 2^{m-1} - 1$; and

(b) reorder the answers according to the permutation $(2j, 2j + 1) \rightarrow (j, j + 2^{m-1})$ for each $j = 0, \ldots, 2^{m-1} - 1$ to get $\tilde{f}(j)$ as the amplitude in dimension j.

This would appear to involve exponentially many operations (for the 2^{m-1} values of j) but using the entanglement effects discussed in equation (2.1), we can achieve the result with only $O(m)$ operations as follows. Note first that

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & w^j \\ 1 & -w^j \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & w^j \end{pmatrix} \equiv H \cdot B_j.$$

The operation B_j in dimensions $(2j, 2j + 1)$ leaves the even dimension unchanged and applies a w^j phase shift in the odd dimension. This may be achieved for all j values *simultaneously* by applying a two-qubit gate C_p to qubits 1 and p for each $p = 1, \ldots, m - 1$. Here C_p is the conditional phase shift of $w^{2^{p-1}}$ applied to qubit p only if both qubits 0 and p are 1. In the standard basis of qubits 0 and p we have

$$C_p = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & w^{2^{p-1}} \end{pmatrix}.$$

Using the entanglement effects described at equation (2.1) we see that the successive application of the $m-1$ operations C_p builds up a phase of w^j in dimension $2j+1$ for each j. The requirement that the zeroth qubit have value 1 selects the odd positions and the conditional phase shift in C_p builds up the value w^j successively for each '1' in the binary expansion of j. All (exponentially many) values of j with '1' in the pth place are treated simultaneously. Finally the 1-qubit operation H is applied just *once* to qubit 0, which simultaneously applies H to all pairs $(2j, 2j + 1)$ given by all possible values of the remaining indices for qubits 1 to $m - 1$ (cf. equation (2.1)).

To implement (b), i.e. the permutation of dimension labels given by

$$\text{even labels: } 2j \rightarrow j, \quad \text{odd labels: } (2j + 1) \rightarrow (j + 2^{m-1}),$$

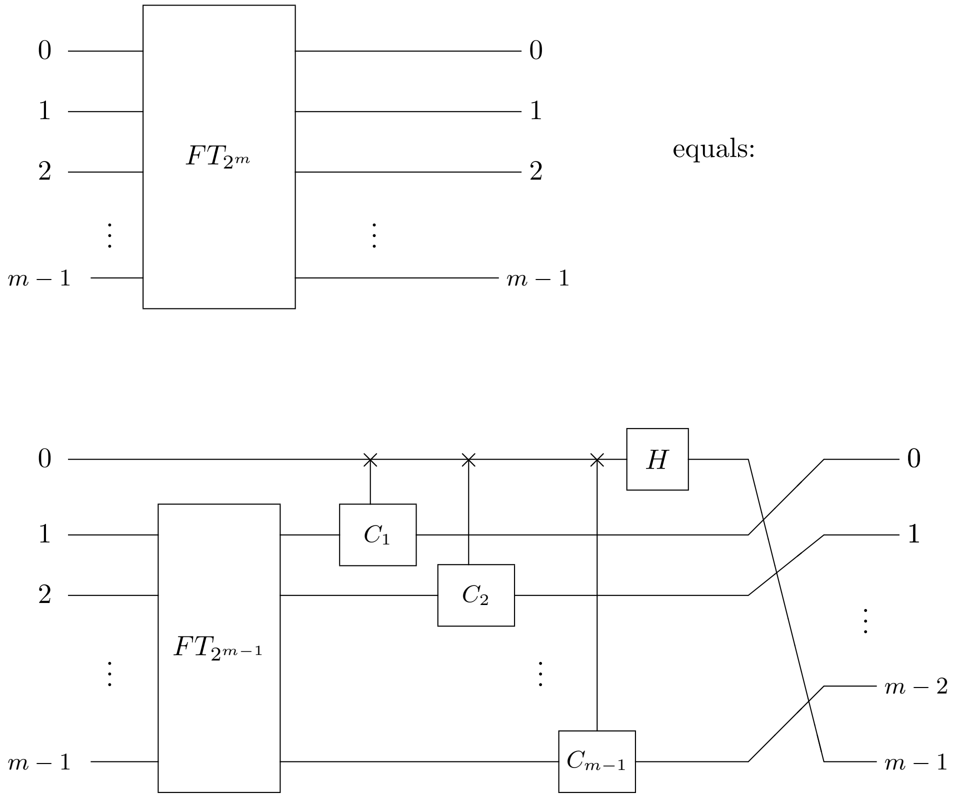

Figure 1. The network diagram for the decomposition of FT_{2^m} into $FT_{2^{m-1}}$ and $O(m)$ extra operations. The conditional phase shift C_p on the pth qubit is denoted by a box on the pth qubit line and a connection across to the zeroth qubit with a cross to denote the fact that its operation on the pth qubit is 'controlled' by the requirement that the zeroth qubit has value 1.

we simply cyclically permute the qubit labels as m-bit strings:

$$i_{m-1} \dots i_1 i_0 \rightarrow i_0 i_{m-1} \dots i_1.$$

If the label was even (i.e. $i_0 = 0$) then the value is halved, and if it was odd ($i_0 = 1$) then the cycling of i_0 to the leading position adds 2^{m-1} and the residual even part is halved. This cycling may be physically achieved by $m-1$ state swaps, of qubits 0 and 1, then 1 and 2, etc., up to qubits $m-2$ and $m-1$. Alternatively we may just reorder the output wires as shown in figure 1.

Iterating this construction for $FT_{2^{m-1}}$ in terms of $FT_{2^{m-2}}$, etc., yields the standard network for the FFT on Z_{2^n}, as given, for example, in Ekert & Jozsa (1996).

If $Q(2^m)$ denotes the number of operations needed to implement $FT(2^m)$ in the quantum context then the above description shows that

$$Q(2^m) = Q(2^{m-1}) + O(m),$$

giving

$$Q(2^n) = O(n^2).$$

This quadratic time quantum algorithm for $FT(2^n)$ is used in Shor's factoring algorithm (Ekert & Jozsa 1996; Jozsa 1998*b*).

4. Utility of the Fourier transform

The utility of the Fourier transform $\mathcal{F}$ in the algorithms of Deutsch, Simon, and Shor (Deutsch & Jozsa 1992; Simon 1994; Shor 1994) has been described in Jozsa (1998*b*). We will here outline in general terms its fundamental application to the determination of periodicities. A different interpretation of $\mathcal{F}$ in terms of the problem of phase estimation has been given in Cleve *et al.* (1998).

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

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

K is necessarily a subgroup of G, called the stabilizer or symmetry group of f. It characterizes 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 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.) 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 $g_1 - g_2 \in K$, i.e. that f is one-to-one within each period—we will obtain in the first register

$$|\psi(g_0)\rangle = \frac{1}{\sqrt{|K|}} \sum_{k \in K} |g_0 + k\rangle, \tag{4.1}$$

corresponding to seeing $f(g_0)$ in the second register and g_0 has been chosen at random. In equation (4.1) 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 equation (4.1) we will see a random element of a random coset, i.e. a label 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. Consider the basis $\{|\chi_g\rangle : g \in G\}$ of shift invariant states introduced in equation (3.5). Next note that the state in equation (4.1) may be written as a g_0-shifted state:

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

Hence, if we write this state in the basis $\{|\chi_g\rangle, g \in G\}$ 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. According to equation (3.6), the Fourier transform converts the shift-invariant basis into the standard basis. Thus after applying $\mathcal{F}$ to equation (4.1) we may read the shift-invariant basis label by reading in the standard basis, yielding information about K.

In terms of the presentation of G given in equation (3.7) and the associated formulae for the irreducible representations given by equations (3.9) and (3.10), we may compute explicitly the pattern of labels associated with a subgroup $K \subset G$. As an example consider $G = Z_{mn}$ and $K = mZ = \{0, m, 2m, \dots, (n-1)m\}$ with $|K| = n$. Then the Fourier transform of the fundamental periodic state $|K\rangle = (1/\sqrt{n})\sum_{k\in K}|k\rangle$ is

$$\mathcal{F}|K\rangle = \frac{1}{n\sqrt{m}}\sum_{l\in G}\left(\sum_{k\in K}\chi_l(k)\right)|l\rangle. \tag{4.2}$$

Thus the labels appearing are precisely those $l \in Z_{mn}$ for which

$$\sum_{k\in K}\chi_l(k) \neq 0. \tag{4.3}$$

To sort out this condition we introduce a further elementary property of irreducible representations. For any group G, the constant function $\chi(g) = 1$ for all $g \in G$ is clearly an irreducible representation (the trivial representation) and using the orthogonality property (B) between χ and any *other* irreducible representation χ' we see that

$$\sum_{g\in G}\chi'(g) = 0.$$

Now χ_l restricted to the subgroup K is an irreducible representation of K, so equation (4.3) can hold if and only if

$$\chi_l(k) = 1, \quad \text{for all } k \in K.$$

According to equations (3.9) and (3.10) we have

$$\chi_l(k) = \exp 2\pi \mathrm{i}\frac{kl}{mn} = \exp 2\pi \mathrm{i}\frac{cl}{n},$$

where we have introduced c using the fact that $k = cm$ is always a multiple of m, by definition of K. This will equal 1 for all $c = 0, \dots, (m-1)$ if and only if l is a multiple of n, i.e. $l = 0, n, 2n, \dots, (m-1)n$. Thus the pattern of labels associated with $mZ \subset Z_{mn}$ is nZ and, furthermore, in equation (4.2) each such label will appear with equal amplitude $1/\sqrt{m}$. A similar calculation for the subgroup $\{0, \xi\} \subset (Z_2)^n$ (where ξ is a chosen n-bit string) shows that the resulting pattern of labels, after applying the Fourier transform for $(Z_2)^n$ to the periodic state $(1/\sqrt{2})(|0\rangle + |\xi\rangle)$, is $\{\nu : \xi \cdot \nu = 0\}$. This fact forms the basis of Simon's algorithm (Simon 1994; Jozsa 1998*b*; Brassard & Hoyer 1997).

5. Conclusion

Let $|\psi\rangle$ be an n-qubit entangled state and U a one-qubit unitary operation. We have seen that the one-step physical operation of applying U to (say) the first qubit of $|\psi\rangle$ corresponds to a state transformation which generally requires an exponential (in n) effort to compute classically. Indeed, mathematically the transformation is represented by a *tensor* product $U \otimes I_2 \otimes \cdots \otimes I_2$ (where I_2 is the 2×2 identity matrix, which represents the operation of 'doing nothing' on the corresponding qubits). The tensor product spreads the effect of U into an exponentially large matrix. Stated otherwise, we can say that the physical operation of *doing nothing* to a subsystem of an entangled system is a highly non-trivial operation and gives rise to an exponentially enhanced information processing capability (when performed in conjunction with some operation on another small part of the system).

We have given an analysis of the implementation of the FFT algorithm in a quantum context and shown that its exponential speed-up (as compared to the corresponding classical computation) derives wholly from the above tensor product property. We have also given a general discussion of the role of entanglement in quantum computation and the utility of the Fourier transform in the known quantum algorithms.

This work was supported in part by the European TMR Research Network ERB-FMRX-CT96-0087 and the National Institute for Theoretical Physics at the University of Adelaide.

References

Bernstein, E. & Vazirani, U. 1993 *Proc. 25th A. Symp. on the Theory of Computing*, pp. 11–20. New York: ACM.

Brassard, G. & Hoyer, P. 1997 An exact polynomial-time algorithm for Simon's problem. quant-ph/9704027.

Cleve, R., Ekert, A., Macchiavello, C. & Mosca, M. 1998 *Proc. R. Soc. Lond.* A **454**, 339–354.

Cooley, J. W. & Tukey, J. W. 1965 *Math. Comp.* **19**, 297.

Cover, T. & Thomas, J. 1991 *Elements of information theory*. New York: Wiley.

Deutsch, D. 1985 *Proc. R. Soc. Lond.* A **400**, 97.

Deutsch, D. & Jozsa, R. 1992 *Proc. R. Soc. Lond.* A **439**, 553–558.

Ekert, A. & Jozsa, R. 1996 *Rev. Mod. Phys.* **68**, 733.

Feynman, R. P. 1982 *Int. J. Theor. Phys.* **21**, 467.

Fraleigh, J. B. 1994 *A first course in abstract algebra*, 5th edn. Reading, MA: Addison-Wesley.

Fuchs, C. & Peres, A. 1996 *Phys. Rev.* A **53**, 2038–2045.

Fulton, W. & Harris, J. 1991 *Representation theory: a first course*, ch. 1, 2. New York: Springer.

Gauss, C. F. 1886 *Theoria interpolationis methodo nova tractata*. Gauss' collected works vol. 3.

Grover, L. 1996 *Proc. 28th A. Symp. on the Theory of Computing*, pp. 212–219. New York: ACM.

Holevo, A. S. 1973 *Probl. Inf. Transm.* **9**, 177.

Jozsa, R. 1998*a* Entanglement and quantum computation. In *The geometric universe* (ed. S. Huggett, L. Mason, K. P. Tod, S. T. Tsou & N. M. J. Woodhouse), pp. 369–379. Oxford University Press.

Jozsa, R. 1998*b* Quantum algorithms and the Fourier transform. *Proc. R. Soc. Lond.* A **454**, 323–337.

Kitaev, A. 1995 Quantum measurements and the Abelian stabiliser problem. quant-ph/9511026.

Maslen, D. K. & Rockmore, D. N. 1995 Generalised FFTs: a survey of some recent results. In *Proc. DIMACS Workshop on Groups and Computation* (ed. L. Finkelstein & W. Kantor). DIMACS Series in Disc. Math. and Comp. Sci., vol. 28, pp. 183–238.

Papadimitriou, C. H. 1994 *Computational complexity* Reading, MA: Addison-Wesley.

Shor, P. 1994 *Proc. 35th A. Symp. on the Foundations of Computer Science*, p. 124. Los Alamitos, CA: IEEE Computer Society.

Simon, D. 1994 *Proc. 35th A. Symp. on the Foundations of Computer Science*, p. 116. Los Alamitos, CA: IEEE Computer Society.

Discussion

Th. Beth (*University of Karlsruhe, Germany*). There are more types of generalized Fourier transforms on non-Abelian groups than those mentioned by Professor Jozsa, for example, affine groups going hand in hand with modern wavelet theory. Taking the point of view that quantum computers are most likely to be high-power correlation machines, the generalized FT algorithms will play an important role in future.

The fast algorithms for this type of transform were published 14 years ago (Beth 1984).

R. Jozsa. Some quantum wavelet transforms have been considered by Hoyer (1997) and it would certainly be interesting to further consider their applicability. However, their quantum implementation appears to involve no new physical effects beyond those that provide the speed-up in the basic quantum Abelian Fourier transform as described in the paper.

Additional references

Beth, Th. 1984 *Uefahren der schnellen Fourier Transformation.* Stuttgart: B. G. Teubner.

Hoyer, P. 1997 Efficient quantum algorithms. quant-ph/9702028.