doi:10.1006/aama.2000.0699, available online at http://www.idealibrary.com on IDEAL®

On Quantum Algorithms for Noncommutative Hidden Subgroups

Mark Ettinger

Los Alamos National Laboratory, Mail Stop B-230, Los Alamos, New Mexico 87545
E-mail: ettinger@lanl.gov

and

Peter Høyer

BRICS[1]*, Department of Computer Science, University of Aarhus, Ny Munkegade, Bldg. 540, DK-8000 Aarhus C, Denmark*
E-mail: hoyer@brics.dk

Received March 26, 1999; accepted May 8, 2000

Quantum algorithms for factoring and finding discrete logarithms have previously been generalized to finding hidden subgroups of finite Abelian groups. This paper explores the possibility of extending this general viewpoint to finding hidden subgroups of noncommutative groups. We present a quantum algorithm for the special case of dihedral groups which determines the hidden subgroup in a linear number of calls to the input function. We also explore the difficulties of developing an algorithm to process the data to explicitly calculate a generating set for the subgroup. A general framework for the noncommutative hidden subgroup problem is discussed and we indicate future research directions.

1. INTRODUCTION

All known quantum algorithms which run super-polynomially faster than the most efficient probabilistic classical algorithm solve special cases of what is called the Abelian hidden subgroup problem. This general formulation

[1] BRICS—Basic Research in Computer Science, Centre of the Danish National Research Foundation.

0196-8858/00 $35.00

includes Shor's celebrated algorithms for factoring and finding discrete logarithms [16]. A very natural question to ask is if quantum computers can efficiently solve the hidden subgroup problem in *noncommutative* groups. This question has been raised regularly [1, 9–11, 14] and seems important since many computational problems generally believed not to be *NP*-hard reduce to finding hidden subgroups, for example the problem of determining if two graphs are isomorphic.

The heart of the idea behind the quantum solution to the Abelian hidden subgroup problem is Fourier analysis on Abelian groups. The difficulties of Fourier analysis on noncommutative groups makes the noncommutative version of the problem very challenging.

In this paper, we present the first known quantum algorithm for a noncommutative subgroup problem. We focus on dihedral groups because they are well-structured noncommutative groups, and because they contain an exponentially large number of different subgroups of small order, making classical guessing infeasible. Our main result is that there exists a quantum algorithm that solves the dihedral subgroup problem using only a linear number of evaluations of the function which is given as input. This is the first time such a result has been obtained for a noncommutative group.

However, we hasten to add that our algorithm does *not* run in polynomial time, even though it only uses few evaluations of the given function. The reason for this is as follows: Our algorithm first applies a certain polynomial-time quantum subroutine a linear number of times, each time producing some output data, and each time using just one application of the given input function. The collection of all the output data determines the hidden subgroup with high probability. We know how to find the subgroup from those data in exponential time, but we do not know if this task can be done efficiently. (See the end of Section 3.)

Two important questions are left open. The first question is if there exists a polynomial-time algorithm (classical or quantum) to postprocess the output data from our quantum subroutine.

The second open question is for what other noncommutative groups similar results can be obtained. A key idea in our algorithm is a way to circumvent the need of a Fourier transform for the dihedral group by utilizing a Fourier transform for an Abelian group. By adapting that idea, Rötteler and Beth [14] have recently found a polynomial-time algorithm for the wreath product $\mathbb{Z}_2^n \wr \mathbb{Z}_2$. More generally, it could prove useful to try to characterize the noncommutative groups for which the subgroup problem can be solved via Abelian Fourier transforms.

In Section 2, we first give the definition of the general hidden subgroup problem. We then discuss the known results for Abelian groups, and finally we define the dihedral groups and state our main result that the dihedral subgroup problem can be solved with few applications of the given input

function. Our main result is stated as Theorem 2.3 and we prove it in Section 3. The solution to the Abelian subgroup problem can perhaps most easily be understood in terms of group representation theory. In Section 4 we review this approach, and in Section 5 we discuss a possibly useful generalization of it to arbitrary noncommutative groups.

2. THE HIDDEN SUBGROUP PROBLEM

The *hidden subgroup problem* is defined as follows:

- **Given**: A function γ: $G \to R$, where G is a finite group and R is an arbitrary finite set.
- **Promise**: There exists a subgroup $H \leq G$ such that γ is *constant* and *distinct* on the left cosets of H.
- **Problem**: Find a generating set for H.

We say of such a function γ that it *fulfills the subgroup promise* with respect to H. We also say of γ that it *has hidden subgroup* H. Note that we are not given the order of H. Without loss of generality we assume that γ is constant and distinct on *left* cosets of H because we may formally rename group elements and convert multiplication on the right to multiplication on the left. We assume throughout this paper that function γ is given as a black box, so that it is not possible to obtain knowledge about it by any other means than evaluating it on points in its domain.

If G is Abelian, then we refer to this problem as the Abelian subgroup problem. Similarly, if the given group is dihedral, then we refer to it as the dihedral subgroup problem.

Classically, if γ is given as a black box, then the hidden subgroup problem is intractable, even in the Abelian case. Simon [17] showed that for $G = \mathbb{Z}_2^n$, it takes time exponential in n just to determine if H is non-trivial or not. Here $\mathbb{Z}_2$ denotes the cyclic group of order 2.

THEOREM 2.1 [17, 5]. *Let γ: $\mathbb{Z}_2^n \to R$ be a function with hidden subgroup H. Suppose γ is given as a black box and that $H = \{0, s\}$ is promised to have order 2. Then any classical algorithm that computes γ on at most $2^{n/3}$ elements of $\mathbb{Z}_2^n$ cannot guess whether the parity of s is even or odd with probability better than $\frac{1}{2} + 2 \times 2^{-n/3}$. Here the parity of $s = (s_1, \ldots, s_n) \in \mathbb{Z}_2^n$ is even if $\sum_{i=1}^n s_i = 0$, and it is odd if $\sum_{i=1}^n s_i = 1$.*

The main idea in the proof of the above theorem is that if the classical algorithm evaluates γ on at most T points, then it can only rule out at most $\binom{T}{2}$ of the $2^n - 1$ possible hidden subgroups of order 2. Thus, if T is small compared to $2^{n/2}$, then close to half of the remaining

$2^n - 1 - \binom{T}{2}$ possible subgroups have generators of odd parity, leaving no hope for the algorithm to guess the parity with probability much better than $1/2$. (See [17, 5] for details.)

In contrast, the Abelian subgroup problem can be solved efficiently on a quantum computer [3–5, 7, 11, 16, 17].

THEOREM 2.2 (Abelian case). *Let $\gamma: G \to R$ be a function that fulfills the* Abelian *subgroup promise with respect to H. There exists a quantum algorithm that outputs a subset $X \subseteq H$ such that X is a generating set for H with probability at least $1 - 1/|G|$, where $|G|$ denotes the order of G. The algorithm uses $O(\log |G|)$ evaluations of γ, and runs in time polynomial in $\log |G|$ and in the time required to compute γ.*

We remark that the above quantum algorithm is efficient in the following strong sense. Namely, it requires only $O(\log |G|)$ evaluations of γ, and it also only requires additional polynomial time.

In the rest of this section and in the succeeding section, we present our algorithm for the dihedral subgroup problem. In Section 4, we then review the quantum solution to the Abelian subgroup problem in terms of group representation theory. For other reviews, see for example [4, 10]. In Section 5, we discuss some of the many challenges arising in non-Abelian cases.

The dihedral group of order $2N$ is the symmetry group of an N–sided polygon. It is isomorphic to a semidirect product of the two cyclic groups $\mathbb{Z}_N$ and $\mathbb{Z}_2$ of order N and 2, respectively,

$$D_N = \mathbb{Z}_N \rtimes_\phi \mathbb{Z}_2, \tag{1}$$

with multiplication defined by

$$(a_1, b_1)(a_2, b_2) = \big(a_1 + \phi(b_1)(a_2),\, b_1 + b_2\big).$$

The homomorphism $\phi: \mathbb{Z}_2 \to \mathrm{Aut}(\mathbb{Z}_N)$ is given by $1 \mapsto \phi(1)(a) = -a$. An element $(a, b) \in D_N$ is a *rotation* if $b = 0$, and a *reflection* if $b = 1$. The group D_N contains N rotations and N reflections, and the N rotations comprise the cyclic subgroup $\mathbb{Z}_N \times \{0\} \leq D_N$ of index 2.

Theorem 2.3 constitutes our main result that the dihedral subgroup problem can be solved with few applications of the given function γ.

THEOREM 2.3 (main theorem) *Let $\gamma: D_N \to R$ be a function that fulfills the* dihedral *subgroup promise with respect to H. There exists a quantum algorithm that, given γ, uses $\Theta(\log N)$ evaluations of γ and outputs a subset $X \subseteq H$ such that X is a generating set for H with probability at least $1 - \frac{2}{N}$.*

We remark that our algorithm (mentioned in Theorem 2.3) is not efficient in the strong sense we discussed above. Specifically, it requires only $O(\log N)$ evaluations of γ, but it does not run in polynomial time. We leave

it as a challenging open question to determining if there exists an algorithm that is efficient in the strong sense.[2] In comparison, any classical algorithm must use exponentially many evaluations of γ just to determine if H is trivial or not with probability bounded away from $1/2$. This holds for the same reasons as in the case of $\mathbb{Z}_2^n$ explained above and proved in [17, 5]. Thus, in terms of the number of evaluations of γ, we achieve an exponential separation of bounded-error quantum computers against bounded-error classical computers.

3. ALGORITHM FOR DIHEDRAL GROUPS

The essential part of the proof of Theorem 2.3 is that it is possible to find a hidden reflection.

THEOREM 3.1 (finding a reflection). *Let γ: $D_N \to R$ be a function that fulfills the dihedral subgroup promise with respect to H. Suppose we are promised that $H = \{0\}$ is either trivial, or $H = \{0, r\}$ is generated by a reflection $r \in D_N$. Then there exists a quantum algorithm that, given γ, outputs either "trivial" or the reflection r. If H is trivial then the output is always "trivial"; otherwise the algorithm outputs r with probability at least $1 - \frac{1}{2N}$. The algorithm uses at most $89\log_2(N) + 7$ evaluations of γ and it runs in time $O(N^{1/2})$.*

We now give the reduction of the general problem stated in Theorem 2.3 to the special case of order-2 subgroups in Theorem 3.1. The key point is that the dihedral group D_N has a large subgroup whose subgroups are all normal in D_N. This allows us to reduce to the original problem on D_N to a smaller dihedral group.

Proof of Theorem 2.3. The following commutative diagram illustrates our approach:

$$\begin{array}{ccccc} H_1 & \subset\longrightarrow & H & \longrightarrow & H/H_1 \\ \downarrow & & \downarrow & & \downarrow \\ \mathbb{Z}_N \times \{0\} & \subset\longrightarrow & \mathbb{Z}_N \rtimes_\phi \mathbb{Z}_2 = D_N & \longrightarrow & D_N/H_1 \end{array}$$

Let $H_1 = H \cap (\mathbb{Z}_N \times \{0\})$ denote the elements of the hidden subgroup H that are contained in the Abelian subgroup of index 2. We start by finding H_1 by applying Theorem 2.2 with γ restricted to $\mathbb{Z}_N \times \{0\}$. This produces a subset $X_1 \subseteq H_1$ such that X_1 generates H_1 with probability at least

[2]The reader may be interested to learn that the authors disagree on the likelihood that the answer is in the affirmative.

$1 - 1/N$, and it uses $O(\log N)$ queries to γ. Let $\langle X_1 \rangle$ denote the subgroup generated by X_1.

The subgroup $\langle X_1 \rangle$ is normal in D_N, and the quotient group $D_N/\langle X_1 \rangle$ is isomorphic to D_M with $M = [\mathbb{Z}_N \times \{0\} : \langle X_1 \rangle]$. Define $\gamma_2 \colon D_N/\langle X_1 \rangle \to R$ by $\gamma_2(g + \langle X_1 \rangle) = \gamma(g)$. Then γ_2 has hidden subgroup $H/\langle X_1 \rangle$.

Suppose $\langle X_1 \rangle = H_1$. Then $H/\langle X_1 \rangle \leq D_N/\langle X_1 \rangle$ is either trivial or generated by a reflection $r_2 + \langle X_1 \rangle$. Apply the algorithm in Theorem 3.1 with γ_2 a number of $t = \lceil \log_2(2N)/\log_2(2M) \rceil$ times, ensuring we find $r_2 + \langle X_1 \rangle$ with probability at least $1 - 1/2N$, provided it exists.

Finally, output X_1, and output also the coset representative $r_2 \in D_N$ if it exists. The overall success probability is at least $(1 - 1/N)(1 - 1/2N) > 1 - 2/N$. The total number of evaluations of γ is at most $O(\log N) + t(89 \log_2(M) + 7)$, as each evaluation of γ_2 requires just one evaluation of γ. ∎

In the rest of this section, we consider only hidden subgroups that are trivial or generated by a reflection. We assume that the reader is familiar with the basic notions of quantum computation. For an excellent introduction to the area, we refer the reader to [2].

The quantum algorithm we shall use to prove Theorem 3.1 uses three registers; the first two hold an element of D_N and the third register holds an element of R, the codomain of function γ. The algorithm is

$$\mathcal{V}_\gamma = \left(\mathbf{F}_N \otimes \mathbf{W} \otimes \mathbf{I}\right) \circ \mathbf{U}_\gamma \circ \left(\mathbf{F}_N^{-1} \otimes \mathbf{W} \otimes \mathbf{I}\right). \tag{2}$$

Here $\mathbf{I}$ is the identity operator and $\mathbf{U}_\gamma$ is any unitary operator that satisfies that

$$\mathbf{U}_\gamma |a\rangle|b\rangle|0\rangle = |a\rangle|b\rangle|\gamma(a, b)\rangle \tag{3}$$

for all elements $(a, b) \in D_N$. The operator

$$\mathbf{F}_N = \frac{1}{N^{1/2}} \sum_{i,\, j=0}^{N-1} \omega_N^{ij} |j\rangle\langle i|$$

is the quantum Fourier transform for the cyclic group $\mathbb{Z}_N$, where $\omega_N = \exp(2\pi\sqrt{-1}/N)$ is the Nth principal root of unity. When $N = 2$, then the Fourier transform $\mathbf{F}_2$ is equal to the Walsh–Hadamard transform $\mathbf{W}$ which maps a qubit in state $|b\rangle$ to the superposition $\frac{1}{\sqrt{2}}(|0\rangle + (-1)^b|1\rangle)$.

Suppose for a moment that we were not given a function defined on the dihedral group $D_N = \mathbb{Z}_N \rtimes_\phi \mathbb{Z}_2$, but instead a function defined on the Abelian group $\mathbb{Z}_N \times \mathbb{Z}_2$. Or equivalently, suppose for the moment that $\phi \colon \mathbb{Z}_2 \to \mathrm{Aut}(\mathbb{Z}_N)$ is the trivial homomorphism. Then we can find any hidden subgroup with probability exponentially close to 1 by applying the experiment

$$(a, b) = \mathcal{M}_{1,2} \circ \mathcal{V}_\gamma |0\rangle|0\rangle|0\rangle \tag{4}$$

a number of $O(\log N)$ times (see Section 4 below). Here $\mathcal{M}_{1,2}$ denotes a measurement of the first two registers with outcome (a, b). A natural question to ask is, how much information, if any, would we gain by performing the experiment given in (4) when γ is defined on D_N and not on $\mathbb{Z}_N \times \mathbb{Z}_2$. Rewriting the state $\mathcal{V}_\gamma|0\rangle|0\rangle|0\rangle$ as a superposition over the basis states shows that we indeed learn something, as quantified in the following lemma.

LEMMA 3.2. *Let γ: $D_N \to R$ fulfill the subgroup promise with respect to $H = \{0, r\}$, where $r = (k_0, 1)$ is a reflection. Then, if we apply quantum algorithm $\mathcal{V}_\gamma$ on the initial state $|0\rangle|0\rangle|0\rangle$, the probability that a measurement of the first two registers yields $(a, 0)$, is*

$$\frac{1}{2N}\big(1 + \cos(2\pi k_0 a/N)\big) = \frac{1}{N}\cos^2(\pi k_0 a/N). \tag{5}$$

Furthermore, the probability that the outcome is $(a, 1)$, is $\frac{1}{N}\sin^2(\pi k_0 a/N)$.

Let $\mathbf{Z}$ denote the discrete random variable defined by the probability mass function

$$\text{Prob}[\mathbf{Z} = z] = \alpha\cos^2(\pi k_0 z/N) \qquad (0 \le z < N), \tag{6}$$

where $\alpha = 1/N$ if $k_0 = 0$ or $2k_0 = N$, and $\alpha = 2/N$ otherwise. Lemma 3.2 provides us with a quantum algorithm for sampling from $\mathbf{Z}$. Intuitively, since the random variable $\mathbf{Z}$ depends on k_0, the more samples we draw from $\mathbf{Z}$, and the more knowledge we gather about k_0 and the hidden reflection $r = (k_0, 1)$. The crucial question therefore becomes, how many samples from $\mathbf{Z}$ do we need to be able to identify k_0 correctly with high probability. Theorem 3.3 below states that we only need a logarithmic number of samples.

THEOREM 3.3. *Let $m \ge \lceil 64 \ln N \rceil$, and let $z_1, \ldots, z_m$ be m independent samples from $\mathbf{Z}$. Let $\kappa \in \{1, \ldots, \lfloor N/2 \rfloor\}$ be such that the sum $\sum_{i=1}^m \cos(2\pi\kappa z_i/N)$ is maximal. Then $\kappa = \min\{k_0, N - k_0\}$ with probability at least $1 - \frac{1}{2N}$.*

The proof of Theorem 3.3 requires two lemmas, the first of them being a result by Hoeffding [8] on the sum of bounded random variables. Hoeffding's lemma says that the probability that the sum of m independent samples is off from its expected value by a constant fraction in m drops exponentially in m.

LEMMA 3.4 (Hoeffding). *Let $\mathbf{X}_1, \ldots, \mathbf{X}_m$ be independent identically distributed random variables with $\ell \le \mathbf{X}_1 \le u$. Then, for all $\alpha > 0$,*

$$\text{Prob}\big[\mathbf{S} - \text{E}[\mathbf{S}] \ge \alpha m\big] \le e^{-2\alpha^2 m/(u-\ell)^2},$$

where $\mathbf{S} = \sum_{i=1}^m \mathbf{X}_i$.

Let $0 < k < N$, and suppose we want to test if $k \stackrel{?}{=} k_0$ or $k \stackrel{?}{=} N - k_0$, where k_0 is given as in Lemma 3.2. Clearly, we can answer that question just by testing if $\gamma(0,0) \stackrel{?}{=} \gamma(k,1)$ or $\gamma(0,0) \stackrel{?}{=} \gamma(N-k,1)$. Lemma 3.5 provides us with another probabilistic method: First draw m samples $\{z_i\}_{i=1}^m$ from $\mathbf{Z}$, and then compute the sum $\sum_{i=1}^m \cos(2\pi k z_i/N)$. Conclude that $k \neq k_0$ and $k \neq N - k_0$ if and only if that sum is at most $m/4$.

LEMMA 3.5. *Fix an integer k with $0 < k < N$. Let $z_1, \ldots, z_m$ be m independent samples from $\mathbf{Z}$. Then with probability at most $e^{-m/32}$, we have*

$$\sum_{i=1}^{m} \cos(2\pi k z_i/N) \leq m/4$$

if $k = k_0$ or $k = N - k_0$, and

$$\sum_{i=1}^{m} \cos(2\pi k z_i/N) \geq m/4$$

otherwise.

Proof. Let f denote the function of $\mathbf{Z}$ defined by $f(z) = \cos(2\pi k z/N)$, and let $\mathbf{X} = f(\mathbf{Z})$ denote the random variable defined by f. Then $-1 \leq \mathbf{X} \leq 1$ and the expected value of $\mathbf{X}$ is

$$\mathrm{E}[\mathbf{X}] = \begin{cases} 1 & \text{if } 2k = 2k_0 = N \\ \frac{1}{2} & \text{if either } k = k_0 \text{ or } k = N - k_0 \\ 0 & \text{otherwise.} \end{cases}$$

If $k \neq k_0$ and $k \neq N - k_0$, then apply Hoeffding's lemma on m independent random variables all having the same probability distribution as $\mathbf{X}$. If $k = k_0$ or $k = N - k_0$, then apply Hoeffding's lemma on m independent random variables all having the same probability distribution as the random variable $\mathrm{E}[\mathbf{X}] - \mathbf{X}$. ∎

We are not only concerned about testing for a specific $0 < k \leq N/2$ if $k \stackrel{?}{=} k_0$ or $k \stackrel{?}{=} N - k_0$, but in testing every one of them. Fortunately, the probability $e^{-m/32}$ (mentioned in Lemma 3.5) is diminutive, so we can reuse the same m samples in all $N/2$ tests, and still it is very likely that the sum $\sum_{i=1}^m \cos(2\pi k z_i/N)$ is larger than $m/4$ if and only if $k = k_0$ or $k = N - k_0$.

Proof of Theorem 3.3. This is a simple consequence of Lemma 3.5. Let $k_0' = \min\{k_0, N - k_0\}$. The probability that $\sum_{i=1}^m \cos(2\pi k_0' z_i/N) \leq m/4$ is at most $e^{-m/32} \leq 1/N^2$. Furthermore, for every integer $0 < k \leq N/2$ not equal to k_0', the probability that $\sum_{i=1}^m \cos(2\pi k z_i/N) \geq m/4$ is also at most $1/N^2$. If $\kappa \neq k_0'$, then one of these $\lfloor N/2 \rfloor$ events must have happened, and the probability for that is upper bounded by $\lfloor N/2 \rfloor 1/N^2 \leq \frac{1}{2N}$. ∎

With this, we now have all the ingredients we need to prove Theorem 3.1.

Proof of Theorem 3.1. The algorithm starts by disposing the possibility that $r = (0, 1)$ by evaluating $\gamma(0, 0)$ and $\gamma(0, 1)$. If the two values are equal, then the algorithm outputs the reflection $(0, 1)$ and stops. If N is even, then the algorithm proceeds by disposing the possibility that $r = (N/2, 1)$, too.

Now, the algorithm applies the quantum experiment given in (4) a number of $m' = 2\lceil 64 \ln N \rceil$ times. Let m denote the number of times it measures zero in the second register. Let $\{a_1, \ldots, a_m\}$ denote the outcomes in the first register, conditioned to that the measurement of the second register yields a zero.[3]

Suppose $m \geq m'/2$, so that we have a sufficient number of samples to apply Theorem 3.3. The algorithm continues with classical post-processing: It finds $1 \leq \kappa \leq \lfloor N/2 \rfloor$ such that the sum $\sum_{i=1}^{m} \cos(2\pi\kappa a_i/N)$ is maximized. It then computes $\gamma(\kappa, 1)$ and compares it with $\gamma(0, 0)$. If they are equal, it outputs the reflection $(\kappa, 1)$ and stops. Otherwise, it performs the same test for $\gamma(N - \kappa, 1)$. If that one also fails, it outputs "trivial."

If $m < m'/2$, then the algorithm performs the same classical post-processing, except that it uses the $m' - m$ measurements for which the output in the second register is 1, and except that it now seeks to maximize $\sum_{i=1}^{m} \sin(2\pi\kappa a_i/N)$.

If H is trivial, then the algorithm returns "trivial" with certainty. If $H = \{0, r\}$, then it outputs $r = (k_0, 1)$ with probability at least $1 - 1/2N$ by Theorem 3.3. The total number of evaluations of γ is at most $m' + 5 < 89 \log_2(N) + 7$. ∎

This concludes the proof of our main theorem. We would like to make a comment on the statement given in Theorem 3.3. We want to find κ that maximizes the sum $\sum_{i=1}^{m} \cos(2\pi\kappa z_i/N)$. This is easy to do in time linear in N, namely just by computing the sum for every possible value of κ. On the one hand, this way of finding the maximum does not require any evaluations of function γ at all, but on the other hand, it unfortunately takes time exponential in $\log N$. We do not know if finding the maximum can be done in time polynomial in $\log N$, with or without additional evaluations of γ, or with or without the help of quantum computers.

4. ABELIAN HIDDEN SUBGROUPS

Theorem 2.2 in Section 2 states that the Abelian subgroup problem can be solved efficiently on a quantum computer. The algorithm which accomplishes this is most easily understood using some basic representation the-

[3] Alternatively, we could apply amplitude amplification [5, 6] to ensure that we will always measure 0 in the second register, instead of as here, only with probability 1/2.

ory for finite Abelian groups which we now briefly review. For more details see the excellent references [12, 13]. For any Abelian group G the group algebra $\mathbb{C}[G]$ is the Hilbert space of all complex-valued functions on G equipped with the standard inner product. A *character* of G is a homomorphism from G to $\mathbb{C}$. The set of characters admits a natural group structure via pointwise multiplication and is a basis for the group algebra. The *Fourier transform* is the linear transformation from the point mass basis of the group algebra to the basis of characters. Further, for any subgroup $H \leq G$, there exists a subgroup of the character group called the orthogonal subgroup $H^{\perp}$ which consists of all characters χ such that $\chi(h) = 1$ for all $h \in H$.

We now sketch the quantum algorithm for solving the Abelian hidden subgroup problem. In the interest of clarity we omit all normalization factors in our description. The algorithm uses two registers; the first register holds an element of the Abelian group G, and the second register holds an element of R, the codomain of the given function γ: $G \to R$. The state of the computer is initialized in the superposition

$$\sum_{g \in G} |g\rangle|\gamma(g)\rangle.$$

We observe the second register with outcome, say, $q \in R$. This action serves to place the first register into a superposition of all elements that map to q under γ. Because γ is constant and distinct on left cosets of H we may write the new state of the computer as

$$\sum_{h \in H} |sh\rangle|q\rangle$$

for some coset sH chosen by the observation of the second register. We then apply the quantum Fourier transform for G on the first register, producing the state

$$\sum_{\chi \in H^{\perp}} \chi^*(s)\, |\chi\rangle|q\rangle,$$

where $\chi^*(s)$ denotes the complex conjugate of $\chi(s)$. Finally, we observe the first register. Notice that this results in a uniformly random sample from $H^{\perp}$.

It can easily be shown that by repeating this experiment of order $\log |H^{\perp}|$ times, we find a subset $X \subseteq H^{\perp}$ that generates $H^{\perp}$ with probability exponentially close to 1. The hidden subgroup $H \leq G$ can then be calculated efficiently from $H^{\perp}$ on a classical computer, essentially by linear algebra.

In summary, the sole purpose of the quantum machine in the above algorithm is to sample uniformly from $H^\perp$. It is known that an arbitrary good approximation to the quantum Fourier transform can be performed efficiently for any finite Abelian group [11], so, assuming the given function γ can be computed in polynomial time, the complete algorithm runs in polynomial time.

5. A GENERALIZED $H^\perp$

We now briefly discuss the main ideas of harmonic analysis on groups, stating as facts the main results that we require. For more detailed information see for example [12, 13]. Let G be a (possibly noncommutative) finite group. A representation of G is a homomorphism ρ: $G \to \mathrm{GL}(V_\rho)$ where V_ρ is called the *representation space* of the representation. The dimension of V_ρ, denoted d_ρ, is called the dimension of the representation.

The representation ρ is *irreducible* if the only invariant subspaces of V_ρ are 0 and V_ρ itself. Two representations ρ_1 and ρ_2 are *equivalent* if there exists an invertible linear map S: $V_{\rho_1} \to V_{\rho_2}$ such that $\rho_1(g) = S^{-1}\rho_2(g)S$ for all $g \in G$.

Let $\Gamma = \{\rho_1, \rho_2, \ldots, \rho_r\}$ be a complete set of inequivalent, irreducible representations of G. Then the identity $\sum_{i=1}^r d_{\rho_i}^2 = |G|$ holds. Furthermore, we may assume that the representations are unitary, i.e., that $\rho(g)$ is a unitary matrix for all $g \in G$ and all $\rho \in \Gamma$. The functions defined by $\rho_{ij} = \rho(g)_{ij}$ for $1 \le i, j \le d_\rho$ are called *matrix coefficients*, and by the previous identity it follows that there are $|G|$ matrix coefficients. It is a fundamental fact that the set of all *normalized* matrix coefficients obtained from any fixed Γ is an orthonormal basis of the group algebra $\mathbb{C}[G]$. The *Fourier transform* with respect to a chosen Γ is a change of basis transformation of the group algebra from the basis of point masses to the basis of matrix coefficients.

If G is commutative, then these definitions reduce to those discussed in Section 4, since in that case, all representations are one-dimensional and each matrix coefficient is just a character. If G is noncommutative, then there exists at least one irreducible representation of G with higher dimension, and in this case the Fourier transform depends on the choice of bases for the irreducible representations. It seems as though this is what complicates the extension of the quantum algorithm for commutative groups to the noncommutative scenario.

It turns out that for our present application it is most useful to use an equivalent notion of the Fourier transform. One may also think of the matrix coefficients as collected together in matrices. In this view the Fourier transform is a matrix-valued function on Γ. For each $f \in \mathbb{C}[G]$, we define the value of the Fourier transform at an irreducible representation $\rho \in \Gamma$

to be

$$\hat{f}(\rho) = \sqrt{\frac{d_\rho}{|G|}} \sum_{g \in G} f(g)\rho(g).$$

If we take individual entries of these matrices, then we recover the coefficients in the basis of matrix coefficients. There is a Fourier inversion formula and therefore f is determined by the matrices $\{\hat{f}(\rho)\}_{\rho \in \Gamma}$.

We may now describe the noncommutative version of $H^\perp$. Let V_ρ^H be the elements of V_ρ that are *pointwise* fixed by H,

$$V_\rho^H = \{v \in V_\rho \mid \rho(h)v = v \text{ for all } h \in H\}.$$

Let P_ρ^H be the projection operator onto V_ρ^H. Then define

$$H^\perp = \{P_\rho^H\}_{\rho \in \Gamma}.$$

The significance of this definition follows from the following elementary result.

THEOREM 5.1. *Let I_H be the indicator function on the subgroup $H \leq G$. Then, for all $\rho \in \Gamma$, we have that $\widehat{I_H}(\rho) = P_\rho^H$.*

COROLLARY 5.2. *Let sH be any coset of $H \leq G$. Then Theorem* 5.1 *immediately yields, for all $\rho \in \Gamma$, that we have $\widehat{I_{sH}}(\rho) = \rho(s)P_\rho^H$.*

Let us summarize the role of this result in the quantum algorithm. If we straightforwardly apply the quantum algorithm described in the previous section to the case where G is noncommutative, then we must determine the resulting probability amplitudes and the information gained by sampling according to these amplitudes.

Recall that the state of the quantum system after the first observation is a superposition of states corresponding to the members of one coset. Thus the state may be described by the indicator function of a coset I_{sH}. The final observation results in observing the name of a matrix coefficient $|\rho, i, j\rangle$. The probability of observing $|\rho, i, j\rangle$ is given by $|c_{\rho,i,j}|^2$ where $c_{\rho,i,j}$ is the coefficient of ρ_{ij} in the expansion of I_{sH} in the basis of matrix coefficients. The corollary above allows us, in theory, to compute these probability amplitudes.

The algorithm for the dihedral groups described in the first part of this paper may be derived from these general methods. By choosing as a basis for the two-dimensional representations of the dihedral group the canonical bases [15, p. 37] conjugated by $\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 \\ 0 & -\sqrt{-1} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$, we obtain the same distribution as specified by (6) in Section 3. For a general noncommutative group it seems as if these methods are necessary for an analysis of the resulting probability amplitudes.

ACKNOWLEDGMENTS

We would like to thank Dan Rockmore, David Maslen, and Hans J. Munkholm from whom we learned noncommutative Fourier analysis, and Richard Hughes, Robert Beals, Joan Boyar, and Umesh Vazirani for helpful conversations on this problem.

REFERENCES

1. R. Beals, Quantum computation of Fourier transforms over symmetric groups, *in* "Proc. 29th Annual ACM Symposium on Theory of Computing," pp. 48–53, The Association for Computing Machinery, New York, 1997.
2. A. Berthiaume, Quantum computation, *in* "Complexity Theory Retrospective II" (L. A. Hemaspaandra and A. L. Selman, Eds.), Chap. 2, pp. 23–51, Springer-Verlag, New York, 1997.
3. D. Boneh and R. Lipton, Quantum cryptoanalysis of hidden linear functions (extended abstract), *in* "Advances in Cryptology—CRYPTO '95," Lecture Notes of Computer Science, Vol. 963, pp. 424–437, Springer-Verlag, Berlin, 1995.
4. G. Brassard and P. Høyer, On the power of exact quantum polynomial time, available on Los Alamos e-Print archive (`http://xxx.lanl.gov`) as quant-ph/9612017.
5. G. Brassard and P. Høyer, An exact quantum polynomial-time algorithm for Simon's problem, *in* "Proc. Fifth Israeli Symposium on Theory of Computing and Systems," pp. 12–23, IEEE Computer Society Press, Los Alamitos, CA, 1997.
6. G. Brassard, P. Høyer, and A. Tapp, Quantum counting, *in* "Proc. 25th International Colloquium on Automata, Languages, and Programming," Lecture Notes of Computer Science, Vol. 1443, pp. 820–831, Springer-Verlag, Berlin, 1998.
7. D. Grigoriev, Testing shift-equivalence of polynomials by deterministic, probabilistic and quantum machines, *Theoret. Comput. Sci.* **180** (1997), 217–228.
8. W. Hoeffding, Probability inequalities for sums of bounded random variables, *J. Amer. Statist. Assoc.* **58** (1963), 13–30.
9. P. Høyer, Efficient quantum transforms, available on Los Alamos e-Print archive (`http://xxx.lanl.gov`) as quant-ph/9702028.
10. R. Jozsa, Quantum algorithms and the Fourier transform, *Proc. Roy. Soc. London Ser. A* **454** (1998), 323–337.
11. A. Kitaev, Quantum measurements and the Abelian stabilizer problem, available on Los Alamos e-Print archive (`http://xxx.lanl.gov`) as quant-ph/9511026.
12. D. Maslen and D. Rockmore, Generalized FFTs—A survey of some recent results, *in* "Proc. 1996 DIMACS Workshop in Groups and Computation," pp. 183–238, Am. Math. Soc., Providence, 1997.
13. D. Rockmore, Some applications of generalized FFTs, *in* "Proc. 1996 DIMACS Workshop in Groups and Computation," pp. 329–370, Am. Math. Soc., Providence, RI, 1997.
14. M. Rötteler and T. Beth, Polynomial-time solution to the hidden subgroup problem for a class of non-Abelian groups, available on Los Alamos e-Print archive (`http://xxx.lanl.gov`) as quant-ph/9812070.
15. J.-P. Serre, "Linear Representations of Finite Groups," Springer-Verlag, Berlin/New York, 1977.
16. P. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM J. Comput.* **26** (1997), 1484–1509.
17. D. Simon, On the power of quantum computation, *SIAM J. Comput.* **26** (1997), 1474–1483.