Hidden shift problem for complex functions

Serge Adonsou[1], Peter Bruin[2], Maris Ozols[3,4], Joppe Stokvis[2,3]

[1]Department of Mathematics and Statistics, University of Guelph, Canada
[2]Mathematical Institute, Leiden University, Netherlands
[3]QuSoft, Amsterdam, Netherlands
[4]Institute for Logic, Language and Computation, and Korteweg-de Vries Institute for Mathematics, University of Amsterdam, Netherlands
Email addresses: sadonsou@uoguelph.ca, p.j.bruin@math.leidenuniv.nl, marozols@gmail.com, j.a.stokvis@math.leidenuniv.nl

July 28, 2025

Abstract

We study quantum algorithms for the hidden shift problem of complex scalar- and vector-valued functions on finite abelian groups. Given oracle access to a shifted function and the Fourier transform of the unshifted function, the goal is to find the hidden shift. We analyze the success probability of our algorithms when using a constant number of queries. For bent functions, they succeed with probability 1, while for arbitrary functions the success probability depends on the 'bentness' of the function.

Contents

1 Introduction

Quantum computers are known to be able to solve problems that classical computers are practically incapable of. For example, the integer factoring and discrete logarithm problems are presumed to be computationally hard on classical computers; this lies at the basis of cryptosystems like RSA [RSA83] and elliptic curve cryptography [Kob87, Mil85]. However, Shor showed in 1994 that both problems are efficiently solvable on a quantum computer [Sho94]. A natural generalization of these problems is the *hidden subgroup problem*.

Problem 1.1 (Hidden subgroup problem)**.** Consider a finite group G, a set S and a function $f : G \to S$. Suppose there exists a subgroup $H \leq G$ such that f is constant on each coset of H, and distinct on different cosets: $f(g) = f(g')$ if and only if $gH = g'H$. Given oracle access to f, find H.

Other instances of the hidden subgroup problem are Simon's problem [Sim97] and the discrete logarithm problem [Sho99]. Problem 1.1 can be solved by a polynomial-time quantum algorithm for finite abelian groups; more precisely, H can be determined in time $\mathrm{poly}(\log |G|)$ [CvD10]. In the instances that can be efficiently solved, the non-abelian case has applications to the graph automorphism problem, the graph isomorphism problem and lattice problems [CvD10]. An efficient solution to the non-abelian hidden subgroup problem for dihedral groups would have great impact on lattice cryptosystems [Reg02]. In contrast to the abelian case, the question whether there is a polynomial-time quantum algorithm to solve the problem for non-abelian groups remains open, except in some special cases [CvD10].

A problem related to the hidden subgroup problem is the *hidden shift problem*.

Problem 1.2 (Hidden shift problem)**.** Let G be a finite group, let S be a set and let $f, g : G \to S$ be functions such that there exists a unique $s \in G$ satisfying $g(x) = f(x - s)$ for all $x \in G$. Given oracle access to f and g, determine s.

From now on, we will only consider the case where the group G is abelian, and we write G additively. One approach to solve hidden shift problem is via Fourier sampling methods. The hidden shift problem for G is equivalent to the hidden subgroup problem for the group $G \rtimes_\varphi \mathbb{Z}/2\mathbb{Z}$, where $\varphi : \mathbb{Z}/2\mathbb{Z} \to \mathrm{Aut}(G)$ is given by $\varphi(0) : x \mapsto x$ and $\varphi(1) : x \mapsto -x$. In particular, the hidden shift problem for finite cyclic groups G is equivalent to the dihedral hidden subgroup problem [CvD10]. The main idea of Fourier sampling methods is to apply a function (e.g. hiding a subgroup) to a uniform superposition, then apply a Fourier transform and measure in the Fourier basis. Sufficient sampling should then give information about the hidden structure of the function. No polynomial-time quantum algorithm is known for the general abelian hidden shift problem. However, some algorithms are known to solve the problem more efficiently than by brute force, in particular Kuperberg's algorithm and its variants [Kup05, Reg04, Kup13, Pei20], which solve the problem in subexponential time, and another algorithm in [FIM$^+$14], which solves the problem efficiently for solvable groups with constant exponent and constant derived length. In the case of boolean functions $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ the query complexity of problem 1.2 depends heavily on the type of functions, specifically on the flatness of their Fourier spectra. The extreme cases are bent functions and delta functions, needing $O(n)$ queries [GRR11] and $\Theta(2^n)$ queries [Gro96] respectively. All other boolean functions lie somewhere in between [CKOR13].

An alternative formulation of the hidden shift problem replaces the finite set S in problem 1.2 with (a finite subset of) the complex numbers $\mathbb{C}$. Previous work has mostly focused on the case of boolean ± 1-valued functions [Röt09, Röt10, GRR11, CKOR13] or functions valued at roots of unity [vDHI06]. We relax this restriction and allow the values of f to be arbitrary complex numbers or even arbitrary vectors in $\mathbb{C}^d$. Following [Röt09, Röt10], we assume oracle access to $\hat{f}$, the Fourier transform of f, which we use as a reference for determining the hidden shift s of g.

Problem 1.3 (Complex-valued hidden shift problem)**.** Let G be a finite abelian group and let $f, g : G \to \mathbb{C}^d$ be functions such that there exists a unique $s \in G$ satisfying $g(x) = f(x - s)$ for all $x \in G$. Given oracle access to g and the Fourier transform of f, determine s.

The standard approach [vDHI06] for solving this version of the problem also starts by applying the oracle for the shifted function g to the uniform superposition and then applying the Fourier transform. Then, instead of measuring, we apply the oracle for $\hat{f}$ to remove all information about f except for the hidden shift s, which can be extracted by applying the inverse Fourier transform (see algorithm 3.5 for more details). In contrast to the Fourier sampling approach, this approach does not involve intermediate measurements and relies purely on constructive and destructive interference.

A class of functions that is particularly amenable to this approach are the so-called bent functions (see definition 1.4). For boolean bent functions, problem 1.3 can be solved efficiently using only a constant number of oracle queries [Röt09, Röt10]. However, it was recently shown that for the class of Maiorana–McFarland bent functions of bounded degree Rötteler's algorithm also admits an efficient classical simulation [AS24]. Earlier, the same ideas as in Rötteler's algorithm had also been applied to multiplicative characters of finite fields and of rings of the form $\mathbb{Z}/N\mathbb{Z}$ [vDHI06], which are almost bent functions.

In this paper, we extend the above methods for problem 1.3 to a larger class of functions. While boolean functions and group characters map to complex roots of unity, we allow for vector-valued functions with image beyond the unit circle or unit sphere. After treating the case of bent functions, we follow the idea in [vDHI06] and look at complex functions that are close to bent (we formalize a notion of approximate 'bentness' in the next section). In line with the results on boolean functions [CKOR13], the success probability of the algorithms we present increases the more bent a complex function is (for precise statements, see section 1.2).

The quantum algorithms we present show quite some similarities with the algorithm for the forrelation problem [AA15]. Given two functions f and g, the forrelation problems asks to determine whether there is a high or low correlation between f and the Fourier transform of g. For bent functions, the forrelation algorithm [AA15, Proposition 6] is identical to algorithm 3.5 for hidden shifts. In appendix C, we extend the definition of forrelation to abelian groups.

1.1 Precise problem statement

The functions for which we solve the hidden shift problem are generalizations of bent functions. The definition below generalizes *boolean* bent functions [Tok15], for which G is a product of copies of $\mathbb{Z}/2\mathbb{Z}$ and f takes values in $\{\pm 1\}$, by making use of the Fourier transform from definition 2.1.

Definition 1.4 (Bent function)**.** Let G be a finite abelian group. A function $f : G \to \mathbb{C}$ is called *bent* if $|f(x)| = 1$ for all $x \in G$ and $|\hat{f}(\phi)| = 1$ for all $\phi \in \hat{G}$.

Remark 1.5. The term 'bent function' is usually reserved for boolean functions $f : \{0,1\}^n \to \{0,1\}$ [Tok15]. This notion relates to our definition 1.4 by letting $G = (\mathbb{Z}/2\mathbb{Z})^n$ and treating the output of f as ± 1-valued via $(-1)^{f(x)}$. This easily extends to arbitrary prime characteristic p by letting $G = \mathbb{F}_{p^n}$ and treating the output of f as a power of ω_p, a p-th root of unity [Mes16, Chapter 13]. The concept we introduce in definition 1.4 is more general in that the values of f must be on the unit circle but need not be roots of unity. The same concept has been introduced independently in different areas and goes by several different names. Most commonly, the vector $|f\rangle = \sum_{g \in G} f(g)|g\rangle$ associated to a bent function f is called *biunimodular* [FR15, GS02]. Another established term is *constant amplitude zero autocorrelation (CAZAC) sequence* [BCM19].

Example 1.6. For $G = \mathbb{Z}/2\mathbb{Z}$, the function f such that $f(0) = 1$ and $f(1) = i$ is bent. More generally, the bent functions on $\mathbb{Z}/2\mathbb{Z}$ are exactly the functions f such that $|f(0)| = 1$ and $f(1) = \pm i f(0)$.

The following two definitions relax bentness by only imposing certain bounds on the function and its Fourier transform, either on the whole domain or on a subset thereof.

Definition 1.7 ($(R, \hat{r})$-bounded)**.** Let G be a finite abelian group and let $R, \hat{r} > 0$ be real numbers. A function $f : G \to \mathbb{C}$ is called $(R, \hat{r})$*-bounded* if

$$\max_{x \in G} |f(x)| \leq R \quad \text{and} \quad \min_{\phi \in \hat{G}} |\hat{f}(\phi)| \geq \hat{r}.$$

Example 1.8. For $G = \mathbb{Z}/2\mathbb{Z}$, the function given by $f(0) = 1$ and $f(1) = 2i$ is $(2, \sqrt{5/2})$-bounded.

Definition 1.9 ($(r, R, \hat{r}, \hat{R}, \alpha, \hat{\alpha})$-bounded)**.** Let G be a finite abelian group and let $0 \leq r \leq R$, $0 \leq \hat{r} \leq \hat{R}$ be real numbers. A function $f : G \to \mathbb{C}$ is called $(r, R, \hat{r}, \hat{R}, \alpha, \hat{\alpha})$*-bounded* if there exist subsets $A \subseteq G, \hat{A} \subseteq \hat{G}$ of sizes $|A| = \alpha|G|, |\hat{A}| = \hat{\alpha}|\hat{G}|$ such that

$$|f(x)| \in [r, R] \text{ for all } x \in A, \quad \text{and} \quad |\hat{f}(\phi)| \in [\hat{r}, \hat{R}] \text{ for all } \phi \in \hat{A}.$$

Example 1.10. All bent functions are $(1,1)$-bounded. All $(R,\hat{r})$-bounded functions are $(0,R,\hat{r},\infty,1,1)$-bounded.

We can generalize the above definitions even further by looking at multidimensional bent functions and their relaxations.

Definition 1.11. A multidimensional complex function $f : G \to \mathbb{C}^d$ is called *bent* if $\|f(x)\| = 1$ for all $x \in G$ and $\|\hat{f}(\phi)\| = 1$ for all $\phi \in \hat{G}$ (the Fourier transform of f is determined coordinate-wise). We call f $(r,R,\hat{r},\hat{R},\alpha,\hat{\alpha})$*-bounded* if there exist subsets $A \subseteq G, \hat{A} \subseteq \hat{G}$ of sizes $|A| = \alpha|G|, |\hat{A}| = \hat{\alpha}|\hat{G}|$ such that

$$\|f(x)\| \in [r,R] \text{ for all } x \in A, \quad \text{and} \quad \|\hat{f}(\phi)\| \in [\hat{r},\hat{R}] \text{ for all } \phi \in \hat{A}.$$

With these generalizations of bent functions, we can define the oracles in problem 1.3. Let $f : G \to \mathbb{C}^d$ be an $(r,R,\hat{r},\hat{R},\alpha,\hat{\alpha})$-bounded function and consider its shift $g(x) = f(x-s)$ for some fixed unknown $s \in G$. We model oracle access to g by an operator O_g acting by permuting the standard basis vectors of $\mathbb{C}^G \otimes \mathbb{C}^2 \otimes \mathbb{C}^{2^n}$ via

$$O_g|x\rangle|a\rangle|b\rangle = |x\rangle|a \oplus \delta_{x-s\in A}\rangle|b \oplus g(x)\rangle, \tag{1.1}$$

where $x \in G$ in the first register denotes the input to g, the second or *indicator* register indicates whether the (shifted) input belongs to $A \subseteq G$, and the third register returns the value of g encoded in n bits (the addition in last two registers is modulo two). Note from definition 1.11 that $\delta_{x-s\in A} = 1$ if and only if $\|g(x)\| \in [r,R]$, so the indicator register can be post-selected to ensure that g is within these bounds. We will assume throughout that all function values $g(x) \in \mathbb{C}^d$ can be stored in an n-bit register with complete precision. In practice, these complex vectors (or numbers) may need to be approximated to fit within n bits (see appendix A for more details).

Quantum access to $\hat{f}$ is similarly modeled by an oracle $O_{\hat{f}}$ acting on $\mathbb{C}^{\hat{G}} \otimes \mathbb{C}^2 \otimes \mathbb{C}^{2^n}$ via

$$O_{\hat{f}}|\phi\rangle|a\rangle|b\rangle = |\phi\rangle|a \oplus \delta_{\phi\in\hat{A}}\rangle|b \oplus \hat{f}(\phi)\rangle. \tag{1.2}$$

Note from definition 1.11 that $\delta_{\phi\in\hat{A}} = 1$ if and only if $\|\hat{f}(\phi)\| \in [\hat{r},\hat{R}]$. Both oracles are linear extensions of bijections, meaning they can also be used classically, allowing for a fair comparison. Since addition is modulo two, both oracles are self-inverse: $O_g^2 = O_{\hat{f}}^2 = I$.

Remark 1.12. For an $(r,R,\hat{r},\hat{R},\alpha,\hat{\alpha})$-bounded function with $\alpha = 1$ (resp. $\hat{\alpha} = 1$) we ignore the indicator register for O_g (resp. $O_{\hat{f}}$), as the indicator is then the constant function 1.

Using these definitions we state a precise version of problem 1.3 that we will solve.

Problem 1.13 (Complex-valued hidden shift problem, precise statement)**.** Let G be a finite abelian group and consider an $(r,R,\hat{r},\hat{R},\alpha,\hat{\alpha})$-bounded function $f : G \to \mathbb{C}^d$. Consider a function $g : G \to \mathbb{C}^d$ such that there is a unique $s \in G$ satisfying $g(x) = f(x-s)$ for all $x \in G$. Given access to the quantum oracles O_g and $O_{\hat{f}}$, determine s.

The above statement captures the most general problem that we solve in this paper. For the purpose of presentation, we first consider special cases where some of the parameters are dropped to make the problem simpler. We summarize our results in the next section.

1.2 Our results

We present a suite of increasingly general quantum algorithms for the hidden shift problem, the most general of which applies to functions $f : G \to \mathbb{C}^d$ for any abelian group G and dimension $d \geq 1$. Our algorithms are adapted from [vDHI06, Section 3] and they find s using only four oracle queries. For bent functions, the algorithms are exact, but for other classes of functions their success probability depends on the 'bentness' of the function f. In line with previous results on the hidden shift problem for boolean functions [CKOR13], our algorithms become less effective when the function is further from bent and closer to a delta function.

The paper is organized as follows. In section 2, we recall the notions of quantum circuits and gate complexity, give some basic results on characters of finite abelian groups and define the Fourier transform on a finite abelian group. The next four sections contain algorithms that can be applied to increasing generalizations of bent functions, and their success probabilities are analyzed. Specifically, we obtain the following results.

- In section 3 we present a classical and quantum algorithm (algorithms 3.1 and 3.5) to solve the hidden shift problem for one-dimensional bent functions with probability one (theorems 3.2 and 3.6).

- In section 4.1 we prove theorem 4.3: Let $f : G \to \mathbb{C}$ be an $(R, \hat{r})$-bounded function. Then there exists a quantum algorithm (algorithm 4.2) using four queries that finds the hidden shift s with probability
$$p(s) = \left(\frac{\hat{r}}{R}\right)^2. \tag{1.3}$$
When f is a bent function, we have $\hat{r} = R = 1$ and we find the shift with certainty.

- In section 4.2 we prove theorem 4.5: Let $f : G \to \mathbb{C}$ be an $(r, R, \hat{r}, \hat{R}, \alpha, \hat{\alpha})$-bounded function. Then there exists a quantum algorithm (algorithm 4.4) using four queries that finds the hidden shift s with probability
$$p(s) = \left(\frac{\hat{r}}{R}\right)^2 \left| \hat{\alpha} - \frac{1}{|G|^{3/2}} \sum_{\phi \in \hat{A}} \sum_{x \notin A+s} \phi(x)\overline{\phi(s)} \frac{g(x)}{\hat{f}(\phi)} \right|^2. \tag{1.4}$$
By taking $A = G$, $\hat{A} = \hat{G}$, $\hat{R} = \infty$ and $r = 0$ this can be shown to reduce to eq. (1.3). Not all parameters in the set $(r, R, \hat{r}, \hat{R}, \alpha, \hat{\alpha})$ can directly be found in the success probability. We elaborate in section 4.4 why they are still included in the definition.

- In section 5 we look at higher-dimensional bent functions and prove theorem 5.9: Let $f = (f_0, \dots, f_{d-1}) : G \to \mathbb{C}^d$ be a d-dimensional $(r, R, \hat{r}, \hat{R}, \alpha, \hat{\alpha})$-bounded function. Then there exists a quantum algorithm (algorithm 5.8) using four queries that finds the hidden shift s with probability
$$p(s) = \left(\frac{\hat{r}}{R}\right)^2 \left| \hat{\alpha} - \frac{1}{|G|^{3/2}} \sum_{\phi \in \hat{A}} \sum_{x \notin A+s} \phi(x)\overline{\phi(s)} \sum_{i=0}^{d-1} \frac{g_i(x)\overline{\hat{f}_i(\phi)}}{\|\hat{f}(\phi)\|^2} \right|^2.$$
It reduces to eq. (1.4) by taking $d = 1$ and noting that in one dimension $\frac{\overline{\hat{f}(\phi)}}{\|\hat{f}(\phi)\|^2} = \frac{1}{\hat{f}(\phi)}$.

The above generalizations are presented in several steps to highlight the different parts of the most basic algorithm (algorithm 3.5) that need to be adapted to solve the most general version of the problem (problem 1.13).

Since $\hat{r} \leq R$ due to Parseval's identity (2.5), the probability in eq. (1.3) equals one only when $\hat{r} = R = 1$, meaning that the function is bent. In section 6 we investigate whether the success probability for non-bent functions can be increased by allowing a more powerful oracle that contains additional tunable phase degrees of freedom θ and χ. We show that these phases lead to additional interference when utilizing only one instead of two ancillary qubits, and prove in theorem 6.2 that this modified algorithm achieves success probability
$$p(s) = \left| \frac{\hat{r}}{R} + \frac{1}{|G|^{3/2}} \sum_{x \in G} \sum_{\phi \in \hat{G}} \phi(x) e^{i\theta(x+s)} \sqrt{1 - \left|\frac{f(x)}{R}\right|^2} e^{-i\chi(\phi)} \sqrt{1 - \left|\frac{\hat{r}}{\hat{f}(\phi)}\right|^2} \right|^2.$$
By appropriately tuning θ and χ, we can obtain success probability one even for some non-bent functions (theorem 6.6).

2 Preliminaries

2.1 Quantum circuits and complexity

We implicitly use the quantum circuit model (see [NC12, Chapter 4]) which describes quantum algorithms by sequences of quantum gates (such as single-qubit gates and CNOT) and measurements. More precisely, a quantum algorithm for a computational problem is a family of circuits where the number of qubits depends only on the input size.

We do not consider parallel gates, and define the time complexity of a quantum algorithm as the number of gates in the corresponding quantum circuit. Viewed as a function of the input size, time complexity is often expressed in big-O notation. A quantum algorithm is called *efficient* if the number of gates is polynomial in the input size.

As mentioned in the introduction, we need to store the elements of a finite subset of $\mathbb{C}^d$ in an n-qubit register, and we assume this can be done with complete precision. We assume the encoding has been chosen such that certain unitary operators appearing in our algorithms can be implemented efficiently, in particular the operators S, U_1, V_{rot} and U_2 defined in eqs. (3.2) and (5.1) to (5.3). Furthermore, when considering a finite abelian group G, we assume that both the elements of G and of its character group $\hat{G}$ are encoded in quantum registers in such a way that the quantum Fourier transform operator F defined in eq. (2.2) is efficiently implementable. For example, it suffices to describe G as a product of cyclic groups, so we can use the implementation from [Kit95].

In the remainder of this paper, we will not explicitly mention time complexity any further, but under the above assumptions all the quantum algorithms we give are efficient in the above sense.

2.2 Group characters and Fourier transform

Throughout the paper, G is a finite abelian group and $f : G \to \mathbb{C}$ is a complex function. We denote by $|G|$ the order of G. A *character* of G is a map $\phi : G \to \mathbb{C}^\times$ such that $\phi(x+y) = \phi(x)\phi(y)$ for all $x, y \in G$, i.e., ϕ is a homomorphism. The characters of G form a group under point-wise multiplication: the product of ϕ_1 and ϕ_2 is the character ϕ given by $\phi(x) = \phi_1(x)\phi_2(x)$ for all $x \in G$. This group is called the *character group* of G and denoted by $\hat{G}$. It is well known that G and $\hat{G}$ are (non-canonically) isomorphic.

If $G = \mathbb{Z}/N\mathbb{Z}$ for a positive integer N, then the map $\hat{G} \to \mathbb{C}^\times$ defined by $\phi \mapsto \phi(1)$ is a group isomorphism from $\hat{G}$ to the group of N-th roots of unity, with inverse given by $\zeta \mapsto (a \mapsto \zeta^a)$. More generally, if $G = \mathbb{Z}/N_1\mathbb{Z} \times \cdots \times \mathbb{Z}/N_l\mathbb{Z}$ for positive integers N_j, then $\hat{G}$ is generated by $\phi_1, \ldots, \phi_l$, where

$$\phi_j(x_1, \ldots, x_l) = \exp(2\pi i x_j / N_j) \quad \text{for } 1 \le j \le l. \tag{2.1}$$

Taking 0 and ϕ_0 as the zero elements of the groups G and $\hat{G}$ respectively, we have

$$\sum_{x \in G} \phi(x) = \begin{cases} |G| & \text{if } \phi = \phi_0, \\ 0 & \text{otherwise,} \end{cases} \qquad \sum_{\phi \in \hat{G}} \phi(x) = \begin{cases} |G| & \text{if } x = 0, \\ 0 & \text{otherwise} \end{cases}$$

thanks to orthogonality of characters.

A central tool in our quantum algorithms is the Fourier transform for abelian groups.

Definition 2.1 (Fourier transform)**.** For any function $f : G \to \mathbb{C}$, the *Fourier transform* of f is the function

$$\begin{aligned} \hat{f} : \hat{G} &\to \mathbb{C} \\ \phi &\mapsto \frac{1}{|G|^{1/2}} \sum_{x \in G} \phi(x) f(x). \end{aligned}$$

The following lemma describes a key property of the Fourier transform: it translates shifts into point-wise multiplication. Since $g(x) = f(x-s)$ is the convolution of f with the delta function at s, this is a special case of the fact that the Fourier transform translates convolution into point-wise multiplication.

Lemma 2.2. *Let $f : G \to \mathbb{C}$ be a function, let $s \in G$, and let $g(x) = f(x-s)$. Then*

$$\hat{g}(\phi) = \phi(s)\hat{f}(\phi) \quad \textit{for all } \phi \in \hat{G}.$$

Proof. This follows by computing the Fourier transform of g, using $\phi(x) = \phi(s)\phi(x-s)$, and making the change of variables $y = x - s$:

$$\begin{aligned}\hat{g}(\phi) &= \frac{1}{|G|^{1/2}} \sum_{x\in G} \phi(x) g(x) \\ &= \frac{1}{|G|^{1/2}} \sum_{x\in G} \phi(x) f(x-s) \\ &= \frac{1}{|G|^{1/2}} \sum_{x\in G} \phi(s)\phi(x-s) f(x-s) \\ &= \phi(s) \frac{1}{|G|^{1/2}} \sum_{y\in G} \phi(y) f(y) \\ &= \phi(s)\hat{f}(\phi). \end{aligned}$$ □

Lemma 2.3. *If $f : G \to \mathbb{C}$ is bent, then so is $g(x) = f(x - s)$ for any $s \in G$.*

Proof. Clearly, we have $|g(x)| = 1$ for all $x \in G$. Using lemma 2.2 and the fact that $|\phi(s)| = 1$, we obtain $|\hat{g}(\phi)| = 1$ for all $\phi \in \hat{G}$. □

2.3 The quantum Fourier transform operator

Let $\mathbb{C}^G$ and $\mathbb{C}^{\hat{G}}$ be finite-dimensional complex vector spaces with bases indexed by G and $\hat{G}$, respectively, and equipped with the standard hermitian inner product for which these bases are orthonormal. Formally,

$$\mathbb{C}^G = \left\{ \sum_{x\in G} \alpha_x |x\rangle : \alpha_x \in \mathbb{C} \text{ for all } x \in G \right\},$$

and similarly for $\mathbb{C}^{\hat{G}}$.

The *Fourier transform* on G is the unitary operator $F : \mathbb{C}^G \to \mathbb{C}^{\hat{G}}$ defined as

$$F = \frac{1}{|G|^{1/2}} \sum_{\phi\in\hat{G}} \sum_{x\in G} \phi(x) |\phi\rangle\langle x|, \tag{2.2}$$

and its inverse is the unitary operator $F^\dagger : \mathbb{C}^{\hat{G}} \to \mathbb{C}^G$ defined as

$$F^\dagger = \frac{1}{|G|^{1/2}} \sum_{x\in G} \sum_{\phi\in\hat{G}} \overline{\phi(x)} |x\rangle\langle\phi|.$$

The operators F and $F^\dagger$ act on the basis vectors of their respective input spaces as follows:

$$F|x\rangle = \frac{1}{|G|^{1/2}} \sum_{\phi\in\hat{G}} \phi(x)|\phi\rangle, \quad \text{and} \quad F^\dagger|\phi\rangle = \frac{1}{|G|^{1/2}} \sum_{x\in G} \overline{\phi(x)}|x\rangle. \tag{2.3}$$

In particular, F maps any superposition with amplitudes f to one with amplitudes $\hat{f}$:

$$F \cdot \frac{1}{|G|^{1/2}} \sum_{x\in G} f(x)|x\rangle = \frac{1}{|G|^{1/2}} \sum_{x\in G} \frac{1}{|G|^{1/2}} \sum_{\phi\in\hat{G}} \phi(x) f(x) |\phi\rangle = \frac{1}{|G|^{1/2}} \sum_{\phi\in\hat{G}} \hat{f}(\phi)|\phi\rangle. \tag{2.4}$$

Thanks to unitarity of F we get *Parseval's identity*

$$\sum_{x\in G} |f(x)|^2 = \sum_{\phi\in\hat{G}} |\hat{f}(\phi)|^2. \tag{2.5}$$

3 Exact algorithm for bent functions

Recall from definition 1.4 that a function $f : G \to \mathbb{C}$ is bent if $|f(x)| = |\hat{f}(\phi)| = 1$ for all $x \in G$ and $\phi \in \hat{G}$. In this section, we present classical and quantum algorithms that solve the hidden shift problem 1.13 with $d = 1$ for the class of bent functions exactly, i.e. with success probability 1.

3.1 Classical algorithm

We present a simple classical algorithm for problem 1.13, based on lemma 2.2. We do not claim that it has optimal time or query complexity, but we state it for the sake of having something to compare our quantum algorithms against.

Let G be a finite abelian group given as a product

$$G = \prod_{j=1}^{l} \mathbb{Z}/N_j\mathbb{Z}$$

of cyclic groups. The dual group $\hat{G}$ is then generated by characters ϕ_j as described in eq. (2.1). For an unknown bent function $f : G \to \mathbb{C}$, given oracle access to a shifted function $g : G \to \mathbb{C}$ and $\hat{f} : \hat{G} \to \mathbb{C}$, the goal is to determine the unique hidden shift $s = (s_1, \ldots, s_l)$.

Algorithm 3.1.

1. *Query g on all elements of G and store the values.*
2. *For $j \in \{1, \ldots, l\}$, repeat the following steps:*
   (a) *compute $\hat{g}(\phi_j) = \frac{1}{\sqrt{|G|}} \sum_{x\in G} \exp(2\pi i x_j/N_j) g(x)$ using the stored values of g,*
   (b) *query $\hat{f}(\phi_j)$ and compute $\phi_j(s) = \frac{\hat{g}(\phi_j)}{\hat{f}(\phi_j)}$,*
   (c) *compute $s_j = \frac{N_j \log(\phi_j(s))}{2\pi i} \bmod N_j$.*
3. *Return $s = (s_1, \ldots, s_l)$.*

Theorem 3.2. *Algorithm 3.1 finds the hidden shift s with certainty using $|G|$ queries to O_g, l queries to $O_{\hat{f}}$ and $O(l|G|)$ arithmetic operations in $\mathbb{C}$ (including complex exponentials and logarithms).*

Proof. The first step uses $|G|$ queries to O_g. The l iterations of step 2(a) use $O(l|G|)$ arithmetic operations in $\mathbb{C}$. (The fast Fourier transform is of no use here since we only need to compute l values of $\hat{g}$.) In the l iterations of step 2(b), we indeed obtain the correct values of $\phi_j(s)$ by lemma 2.2, using l queries to $O_{\hat{f}}$ and l arithmetic operations. Finally, we obtain the correct $s_1, \ldots, s_l$ in step 2(c) because of eq. (2.1), using $O(l)$ arithmetic operations in $\mathbb{C}$. □

3.2 Quantum algorithm

For the quantum algorithm we assume the same setting of a finite abelian group G and two bent functions $f, g : G \to \mathbb{C}$ such that $g(x) = f(x - s)$ for some unique shift s. We assume access to the quantum oracles O_g and $O_{\hat{f}}$ as given in eq. (1.1) and eq. (1.2), where we ignore the indicator register. Because the images of g and $\hat{f}$ are on the unit circle, we can define new quantum oracles $\tilde{O}_g \in \mathrm{U}(\mathbb{C}^G)$ and $\tilde{O}_{1/\hat{f}} \in \mathrm{U}(\mathbb{C}^{\hat{G}})$ by

$$\tilde{O}_g|x\rangle = g(x)|x\rangle, \quad \tilde{O}_{1/\hat{f}}|\phi\rangle = \hat{f}(\phi)^{-1}|\phi\rangle. \tag{3.1}$$

We also assume that we have access to an operator S acting on a quantum register containing complex numbers z of absolute value 1 by

$$S|z\rangle = z|z\rangle. \tag{3.2}$$

In practice, z will have finite precision, so we can implement S using a sequence of controlled rotation operators. This may introduce numerical errors, the effects of which are discussed in appendix A.

Lemma 3.3. *The quantum oracles $\tilde{O}_g$ and $\tilde{O}_{1/\hat{f}}$ in eq. (3.1) can be efficiently implemented using two queries to O_g and to $O_{\hat{f}}$, respectively.*

Proof. Since $|g(x)| = 1$ for all $x \in G$, we can implement $\tilde{O}_g$ using O_g and the operator S defined in eq. (3.2) as

$$\begin{aligned}|x\rangle|0\rangle &\xmapsto{O_g} |x\rangle|g(x)\rangle \\ &\xmapsto{I\otimes S} g(x)|x\rangle|g(x)\rangle \\ &\xmapsto{O_g} g(x)|x\rangle|0\rangle.\end{aligned}$$

One can similarly implement $\tilde{O}_{1/\hat{f}}$ using O_f and the inverse of S. □

Remark 3.4. Under somewhat stronger assumptions on g and O_g, a single call to O_g suffices to implement $\tilde{O}_g$. Namely, let us assume that g takes values in a known finite subgroup C of $\mathrm{U}(\mathbb{C}) = \{z \in \mathbb{C} : |z| = 1\}$. Then the dual group $\hat{C}$ is canonically isomorphic to $\mathbb{Z}/|C|\mathbb{Z}$, and the quantum Fourier transform F_C sends the uniform superposition $|C|^{-1/2} \sum_{z \in C} |z\rangle$ to $|0\rangle$, where 0 denotes the trivial element of $\hat{C}$. Furthermore, suppose that we have access to a modified version of O_g that takes a state of the form $|x\rangle|z\rangle$ with $x \in G$ and $z \in C$ and sends it to $|x\rangle|zg(x)\rangle$. Then we can implement $\tilde{O}_g$ using the following variant of phase kickback (cf. [Röt10, proof of Theorem 4.1]):

$$\begin{aligned}
|x\rangle|0\rangle &\xmapsto{I \otimes F_C^\dagger} \frac{1}{|C|^{1/2}} \sum_{z \in C} |x\rangle|z\rangle \\
&\xmapsto{I \otimes S^{-1}} \frac{1}{|C|^{1/2}} \sum_{z \in C} z^{-1} |x\rangle|z\rangle \\
&\xmapsto{O_g} \frac{1}{|C|^{1/2}} \sum_{z \in C} z^{-1} |x\rangle|zg(x)\rangle \\
&= \frac{1}{|C|^{1/2}} \sum_{w \in C} (wg(x)^{-1})^{-1} |x\rangle|w\rangle = \frac{g(x)}{|C|^{1/2}} \sum_{w \in C} w^{-1} |x\rangle|w\rangle \\
&\xmapsto{I \otimes S} \frac{g(x)}{|C|^{1/2}} \sum_{w \in C} |x\rangle|w\rangle \\
&\xmapsto{I \otimes F_C} g(x)|x\rangle|0\rangle.
\end{aligned}$$

Similarly, we can implement $\tilde{O}_{1/\hat{f}}$ using a single call to a modified oracle for $\hat{f}$ under analogous assumptions.

Below we describe an exact and efficient quantum algorithm for problem 1.13, which uses the oracles from eq. (3.1) and generalizes an earlier algorithm for *boolean* bent functions [Röt10, Theorem 4.1]. Our algorithm uses the state space $\mathbb{C}^G$.

Algorithm 3.5.

1. *Prepare the uniform superposition* $|G|^{-1/2} \sum_{x \in G} |x\rangle$.
2. *Apply the operator* $F^\dagger \tilde{O}_{1/\hat{f}} F \tilde{O}_g$.
3. *Measure in the standard basis.*

Theorem 3.6. *If f is a bent function, then algorithm 3.5 determines the hidden shift s with certainty using one call to the oracle $\tilde{O}_g$ and one call to the oracle $\tilde{O}_{1/\hat{f}}$.*

Proof. The state changes as

$$\begin{aligned}
\frac{1}{|G|^{1/2}} \sum_{x \in G} |x\rangle \xmapsto{\tilde{O}_g} \frac{1}{|G|^{1/2}} \sum_{x \in G} g(x)|x\rangle \xmapsto{F} \frac{1}{|G|^{1/2}} \sum_{\phi \in \hat{G}} \hat{g}(\phi)|\phi\rangle = \frac{1}{|G|^{1/2}} \sum_{\phi \in \hat{G}} \phi(s)\hat{f}(\phi)|\phi\rangle \\
\xmapsto{\tilde{O}_{1/\hat{f}}} \frac{1}{|G|^{1/2}} \sum_{\phi \in \hat{G}} \phi(s)|\phi\rangle \xmapsto{F^\dagger} |s\rangle,
\end{aligned}$$

where we made use of eqs. (2.3) and (2.4) and lemma 2.2. □

All our subsequent algorithms rely on the same principle as algorithm 3.5. Namely, the state remains in a uniform superposition at key steps throughout the algorithm, with all information encoded in relative phases between standard or Fourier basis states. We manipulate the phase function with oracle calls and Fourier transform in the following way:

$$1 \xmapsto{\tilde{O}_g} g \xmapsto{F} \hat{g} = \hat{\delta}_s \cdot \hat{f} \xmapsto{\tilde{O}_{1/\hat{f}}} \hat{\delta}_s \xmapsto{F^\dagger} \delta_s, \tag{3.3}$$

allowing us to learn s at the final step. Although in subsequent sections the algorithms and calculations become more difficult, the underlying idea stays the same.

4 Approximate algorithms for bent-like functions

The hidden shift problem for bent functions can be solved with certainty by algorithm 3.5 using one query to each of the two phase oracles $\tilde{O}_g$ and $\tilde{O}_{1/\hat{f}}$ from eq. (3.1). If we want to extend the algorithm to functions $f : G \to \mathbb{C}$ that are not bent, we cannot use phase oracles anymore. Recall from lemma 3.3 that each phase oracle $\tilde{O}_g$ and $\tilde{O}_{1/\hat{f}}$ can be implemented by two calls to one of the original oracles O_g and $O_{1/\hat{f}}$ from eqs. (1.1) and (1.2), respectively. In all subsequent extensions of algorithm 3.5 we rely on these two original oracles.

The main idea is to shift the encoding of f from phase to the whole amplitude. Since G is finite, there exists an $R > 0$ such that $|f(x)| \leq R$ for all $x \in G$, so the function $f(\cdot)/R$ maps to the unit disc and hence

$$\underbrace{\frac{f(x)}{R}|0\rangle}_{\text{good}} + \underbrace{\sqrt{1 - \left|\frac{f(x)}{R}\right|^2}|1\rangle}_{\text{bad}} \tag{4.1}$$

is a valid qubit state. If $|f|$ is valued very close to R, most of the amplitude is at $|0\rangle$. The more amplitude ends up in the 'good part' the better the algorithm performs. In particular, for bent functions we can take $R = 1$ and all of the amplitude ends up in the good part.

In this section, all algorithms require two extra qubits, one for O_g and one for $O_{1/\hat{f}}$, to encode the oracle output into amplitudes according to eq. (4.1).

4.1 First generalization: $(R, \hat{r})$-bounded functions

Any complex function on a finite abelian group is bounded below and above. In the first generalization of algorithm 3.5, we additionally need the Fourier transform to vanish nowhere. In other words, we assume our function f to be $(R, \hat{r})$-bounded as in definition 1.7.

Proposition 4.1. *For all $(R, \hat{r})$-bounded functions f,*

1. *$\|f\|_\infty \leq R$ and $\|1/\hat{f}\|_\infty \leq 1/\hat{r}$,*
2. *$\hat{r} \leq R$, and if $\hat{r} = R$ then $|f(x)| = R$ and $|\hat{f}(\phi)| = \hat{r}$ for all $x \in G$ and $\phi \in \hat{G}$,*
3. *for any $s \in G$ the function $g(x) = f(x - s)$ is also $(R, \hat{r})$-bounded.*

Proof. Claim 1 follows immediately from definition 1.7. For claim 2, note that Parseval's identity (2.5) implies

$$|G|\hat{r}^2 \leq \sum_{\phi \in \hat{G}} |\hat{f}(\phi)|^2 = \sum_{x \in G} |f(x)|^2 \leq |G|R^2.$$

Hence $\hat{r} \leq R$, and if $\hat{r} = R$ then both inequalities are equalities, which together with definition 1.7 implies the claim. For claim 3, note that if f is bounded from above by R then so is g. By lemma 2.2, $|g(\phi)| \geq \hat{r}$ for all ϕ, hence g is also $(R, \hat{r})$-bounded. □

Below we give a quantum algorithm for the hidden shift problem of $(R, \hat{r})$-bounded functions. We assume access to oracles for g and $\hat{f}$, and use additional unitary operators U_1 and U_2 that create states of the form (4.1). Specifically, we use the following quantum operators:

- Quantum oracles O_g and $O_{\hat{f}}$ to access g and $\hat{f}$ given by eqs. (1.1) and (1.2) (we ignore the second register as in section 3.2).
- The quantum operation $U_1 \in \mathrm{U}(\mathbb{C}^{2^n} \otimes \mathbb{C}^2)$ acting as $U_1 = \sum_w |w\rangle\langle w| \otimes U_1(w)$ where

$$U_1(w)|0\rangle = \frac{w}{R}|0\rangle + \sqrt{1 - |w/R|^2}|1\rangle \tag{4.2}$$

  and w ranges over some subset of $\mathbb{C}$ that contains both $\{g(x) : x \in G\}$ and $\{\hat{f}(\phi) : \phi \in \hat{G}\}$, and $|w\rangle$ denotes an n-bit encoding of w (we do not specify an explicit encoding).
- The quantum operation $U_2 \in \mathrm{U}(\mathbb{C}^{2^n} \otimes \mathbb{C}^2)$ acting as $U_2 = \sum_w |w\rangle\langle w| \otimes U_2(w)$ where[1]

$$U_2(w)^\dagger|0\rangle = \begin{cases} \frac{\hat{r}}{\bar{w}}|0\rangle + \sqrt{1 - |\hat{r}/\bar{w}|^2}|1\rangle & \text{if } w \neq 0, \\ |0\rangle & \text{if } w = 0. \end{cases} \tag{4.3}$$

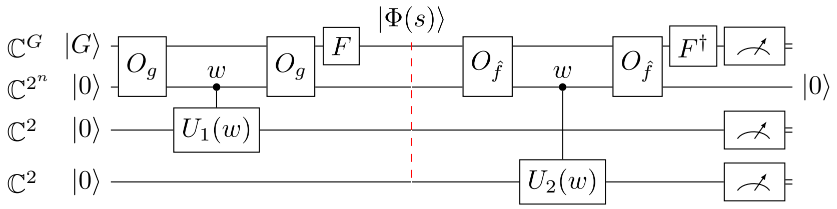
[1] We only need to know $U_2(w)^\dagger|0\rangle$ or $\langle 0|U_2(w)$ since we will analyze the second half of the circuit backwards.

- The quantum Fourier transform $F \in \mathrm{U}(\mathbb{C}^G)$ for G, see eq. (2.2).

The algorithm works on the space $\mathbb{C}^G \otimes \mathbb{C}^{2^n} \otimes \mathbb{C}^2 \otimes \mathbb{C}^2$, where the second register is extra n-qubit workspace that is returned to its original state.

Algorithm 4.2.

1. *Prepare the uniform superposition* $|G, 0, 0, 0\rangle = \frac{1}{|G|^{1/2}} \sum_{x \in G} |x, 0, 0, 0\rangle$.
2. *Perform the following quantum circuit:*



3. *If the measurements on the third and fourth register result in* 0*, output the value of the first register. Otherwise, output* FAIL.

Theorem 4.3. *For an* $(R, \hat{r})$*-bounded function* $f : G \to \mathbb{C}$*, the probability of finding the hidden shift* s *by algorithm 4.2 is*

$$p(s) = \left(\frac{\hat{r}}{R}\right)^2. \tag{4.4}$$

The algorithm uses two calls to each of the oracles O_g *and* $O_{\hat{f}}$*.*

Proof. Observe that the second register is always restored to $|0\rangle$ since each oracle is called twice and the state is not affected by controls. Hence, to successfully find the hidden shift s, we would like the final state before the measurement to be $|s, 0, 0, 0\rangle$. Let $|\Phi(s)\rangle$ denote the state at the marked time point, and similarly let $|\Psi(s)\rangle$ denote the state at the same time point but when running the circuit backwards from the desired target state $|s, 0, 0, 0\rangle$. Then the probability of finding the hidden shift s is

$$p(s) = |\langle\Psi(s)|\Phi(s)\rangle|^2 .$$

To compute this probability, we first analyze how the state changes when running the algorithm forwards till the marked time point. Using eqs. (1.1) and (4.2),

$$\begin{aligned}
\frac{1}{|G|^{1/2}} \sum_{x \in G} |x, 0, 0, 0\rangle \overset{O_g}{\longmapsto} & \frac{1}{|G|^{1/2}} \sum_{x \in G} |x, g(x), 0, 0\rangle \\
\overset{U_1}{\longmapsto} & \frac{1}{|G|^{1/2}} \sum_{x \in G} |x\rangle |g(x)\rangle \left(\frac{g(x)}{R} |0\rangle + \sqrt{1 - \left|\frac{g(x)}{R}\right|^2} |1\rangle \right) |0\rangle \\
\overset{O_g}{\longmapsto} & \frac{1}{|G|^{1/2}} \sum_{x \in G} |x\rangle |0\rangle \left(\frac{g(x)}{R} |0\rangle + \sqrt{1 - \left|\frac{g(x)}{R}\right|^2} |1\rangle \right) |0\rangle .
\end{aligned}$$

According to eqs. (2.3) and (2.4), the Fourier transform produces

$$|\Phi(s)\rangle = \frac{1}{|G|^{1/2}} \sum_{\phi \in \hat{G}} |\phi\rangle |0\rangle \left(\frac{\hat{g}(\phi)}{R} |0\rangle + \frac{1}{|G|^{1/2}} \sum_{x \in G} \phi(x) \sqrt{1 - \left|\frac{g(x)}{R}\right|^2} |1\rangle \right) |0\rangle .$$

Similarly, running the algorithm backwards from $|s, 0, 0, 0\rangle$ to the same time point we get

$$\begin{aligned}
|\Psi(s)\rangle &= O_{\hat{f}} U_2^\dagger O_{\hat{f}} F |s, 0, 0, 0\rangle \\
&= \frac{1}{|G|^{1/2}} \sum_{\phi \in \hat{G}} \phi(s) |\phi\rangle |0\rangle |0\rangle \left(\frac{\hat{r}}{\overline{\hat{f}(\phi)}} |0\rangle + \sqrt{1 - \left|\frac{\hat{r}}{\overline{\hat{f}(\phi)}}\right|^2} |1\rangle \right),
\end{aligned}$$

where we used eq. (4.3). Putting everything together, the probability of recovering s is

$$p(s) = |\langle\Psi(s)|\Phi(s)\rangle|^2 = \left|\frac{1}{|G|}\sum_{\phi\in\hat{G}} \overline{\phi(s)}\frac{\hat{g}(\phi)}{R}\frac{\hat{r}}{\hat{f}(\phi)}\right|^2 = \left|\frac{1}{|G|}\sum_{\phi\in\hat{G}} \phi(s)\overline{\phi(s)}\frac{\hat{r}}{R}\right|^2 = \left(\frac{\hat{r}}{R}\right)^2,$$

where the penultimate equality uses $\hat{g}(\phi) = \phi(s)\hat{f}(\phi)$ from lemma 2.2. □

4.2 Second generalization: more parameters

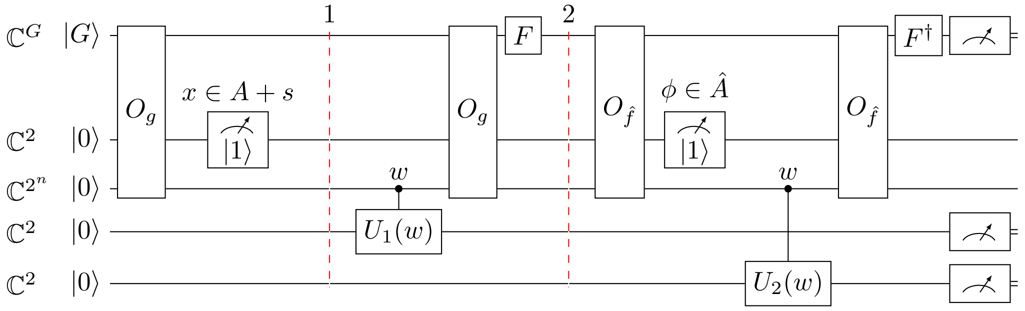
Recall that definition 1.7 requires an upper bound R on f and a lower bound $\hat{r}$ on $\hat{f}$. This is somewhat restrictive since, for example, no zeros are allowed in the Fourier spectrum $\hat{f}$. Moreover, even if f is very large or $\hat{f}$ very small at a single point, the ratio $\hat{r}/R$ and hence the success probability (4.4) is very small. This motivates a generalization to $(r, R, \hat{r}, \hat{R}, A, \hat{A})$-bounded functions as in definition 1.9.

In this section, we give a quantum algorithm for the hidden shift problem of such functions which is very similar to algorithm 4.2. The main difference is that we include the indicator registers of O_g and $O_{\hat{f}}$ (i.e., the second register in eqs. (1.1) and (1.2)) and post-select[1] them to $|1\rangle$. This has the effect of restricting the uniform superpositions over G and $\hat{G}$ to subsets $A + s$ and $\hat{A}$ so that $|g(x)| \in [r, R]$ and $|\hat{f}(\phi)| \in [\hat{r}, \hat{R}]$, respectively (see definition 1.9). The algorithm works on the space $\mathbb{C}^G \otimes \mathbb{C}^2 \otimes \mathbb{C}^{2^n} \otimes \mathbb{C}^2 \otimes \mathbb{C}^2$, where the second register is new compared to the previous section. Other than that here we use the same sequence of operations as algorithm 4.2.

Algorithm 4.4.

1. *Prepare the uniform superposition* $|G, 0, 0, 0, 0\rangle := \frac{1}{|G|^{1/2}}\sum_{x\in G}|x, 0, 0, 0, 0\rangle$.

2. *Perform the following quantum circuit:*



Output FAIL *if either of the two post-selections on the second register fail to produce* $|1\rangle$.

3. *Measure the first, fourth and fifth register in the standard basis. If the measurements on the last two registers result in* 0, *output the value of the first register. Otherwise output* FAIL.

Theorem 4.5. *For an* $(r, R, \hat{r}, \hat{R}, \alpha, \hat{\alpha})$*-bounded function* $f : G \to \mathbb{C}$, *the probability of finding the hidden shift with algorithm 4.4 is*

$$p(s) = \left(\frac{\hat{r}}{R}\right)^2 \left|\hat{\alpha} - \frac{1}{|G|^{3/2}}\sum_{\phi\in\hat{A}}\sum_{x\notin A+s}\phi(x)\overline{\phi(s)}\frac{g(x)}{\hat{f}(\phi)}\right|^2. \tag{4.5}$$

The algorithm uses two calls to each of the oracles O_g *and* $O_{\hat{f}}$.

Proof. Time point 1 is reached with probability α and leads to state

$$\frac{1}{|A|^{1/2}}\sum_{x\in A+s}|x, 1, g(x), 0, 0\rangle.$$

Next, at time point 2 the state changes to

[1] Post-selection means that we terminate the algorithm if the measurement does not produce the desired outcome. Both intermediate measurements in algorithm 4.4 post-select the qubit to state $|1\rangle$.

$$|\Phi(s)\rangle = \frac{1}{|A|^{1/2}|G|^{1/2}} \sum_{\phi\in\hat{G}} \sum_{x\in A+s} \phi(x)|\phi,0,0\rangle \left(\frac{g(x)}{R}|0\rangle + \sqrt{1-\left|\frac{g(x)}{R}\right|^2}|1\rangle \right)|0\rangle. \tag{4.6}$$

Similar to the proof of theorem 4.3, let us now analyze the circuit backwards. At the end of the circuit the second and third register are always restored to $|0\rangle$ because both oracles are called twice, so the final state should be $|s,0,0,0,0\rangle$ for the algorithm to successfully find the hidden shift s. Starting with this state and running the circuit backwards, time point 2 is reached with probability $\hat{\alpha}$ leading to state

$$|\Psi(s)\rangle = \frac{1}{|\hat{A}|^{1/2}} \sum_{\phi\in\hat{A}} \phi(s)|\phi,0,0,0\rangle \left(\frac{\hat{r}}{\overline{\hat{f}(\phi)}}|0\rangle + \sqrt{1-\left|\frac{\hat{r}}{\overline{\hat{f}(\phi)}}\right|^2}|1\rangle \right). \tag{4.7}$$

The total probability of finding the shift s can now be computed as

$$\begin{aligned} p(s) &= \alpha\hat{\alpha}\left|\langle\Psi(s)|\Phi(s)\rangle\right|^2 \\ &= \alpha\hat{\alpha}\left|\frac{1}{|\hat{A}|^{1/2}|A|^{1/2}|G|^{1/2}} \sum_{\phi\in\hat{A}} \sum_{x\in A+s} \phi(x)\overline{\phi(s)}\frac{g(x)\hat{r}}{\hat{f}(\phi)R}\right|^2 \\ &= \left(\frac{\hat{r}}{R}\right)^2 \left|\frac{1}{|G|^{3/2}} \sum_{\phi\in\hat{A}} \sum_{x\in A+s} \phi(x)\overline{\phi(s)}\frac{g(x)}{\hat{f}(\phi)}\right|^2. \end{aligned} \tag{4.8}$$

The second summation in eq. (4.8) can be split into two parts:

$$\begin{aligned} \frac{1}{|G|^{3/2}} \sum_{\phi\in\hat{A}} \sum_{x\in A+s} \phi(x)\overline{\phi(s)}\frac{g(x)}{\hat{f}(\phi)} &= \frac{1}{|G|^{3/2}} \sum_{\phi\in\hat{A}} \left(\sum_{x\in G} - \sum_{x\notin A+s} \right) \left(\phi(x)\overline{\phi(s)}\frac{g(x)}{\hat{f}(\phi)} \right) \\ &= \frac{1}{|G|} \sum_{\phi\in\hat{A}} \left(\overline{\phi(s)}\frac{\hat{g}(\phi)}{\hat{f}(\phi)} - \frac{1}{|G|^{1/2}} \sum_{x\notin A+s} \phi(x)\overline{\phi(s)}\frac{g(x)}{\hat{f}(\phi)} \right) \\ &= \hat{\alpha} - \frac{1}{|G|^{3/2}} \sum_{\phi\in\hat{A}} \sum_{x\notin A+s} \phi(x)\overline{\phi(s)}\frac{g(x)}{\hat{f}(\phi)}. \end{aligned}$$

Plugging this back into eq. (4.8) we get the desired expression. □

The expression in theorem 4.5 reduces to that of theorem 4.3 if we set $r=0$, $\hat{R}=\infty$ and require $A=G$, $\hat{A}=\hat{G}$. Then $\hat{\alpha}=1$ and the double summation in eq. (4.5) vanishes, giving us the probability in eq. (4.4) from the first generalization.

4.3 Example: characters

As an application of theorem 4.5, let us calculate the success probability of algorithm 4.4 for two examples: *primitive Dirichlet characters* and *finite field characters*.

Primitive Dirichlet characters. Let $G=\mathbb{Z}/n\mathbb{Z}$ be the additive group of integers modulo n and consider a multiplicative character $f:\mathbb{Z}/n\mathbb{Z}\to\mathbb{C}$. That is, f is a group homomorphism on the multiplicative group of units $(\mathbb{Z}/n\mathbb{Z})^*$, extended to $\mathbb{Z}/n\mathbb{Z}$ by defining it to be zero elsewhere. We call f *imprimitive* when there is a divisor $n_1\mid n$ and a multiplicative character $f_1:(\mathbb{Z}/n_1\mathbb{Z})^*\to\mathbb{C}$ such that

$$f(x)=f_1(x \bmod n_1) \text{ for all } x\in(\mathbb{Z}/n\mathbb{Z})^*,$$

otherwise we call f *primitive*.

To solve the hidden shift problem for a primitive character f we need to find the necessary parameters in definition 1.9. Setting $A=(\mathbb{Z}/n\mathbb{Z})^*$ we have

$$|f(x)| = \begin{cases} 1 & \text{if } x\in A, \\ 0 & \text{otherwise.} \end{cases}$$

It holds that $|A| = \varphi(n)$, the Euler's totient function, so we take $\alpha = \varphi(n)/n$ and $r = R = 1$. For the Fourier transform we make the identification

$$G \cong \hat{G},\ y \mapsto (\phi_y : x \mapsto e^{2\pi i x y/n}).$$

For any $y \in (\mathbb{Z}/n\mathbb{Z})^*$, using $\phi_y(x) = \phi_1(xy)$, $Ay = A$, and the multiplicativity of f we find that

$$\begin{aligned}\hat{f}(y) &= \frac{1}{|G|^{1/2}} \sum_{x \in A} f(x)\phi_y(x) \\ &= \frac{1}{|G|^{1/2}} \sum_{x' \in Ay} f(x'y^{-1})\phi_1(x') \\ &= \frac{1}{|G|^{1/2}} \overline{f(y)} \sum_{x' \in A} f(x')\phi_1(x') \\ &= \overline{f(y)}\hat{f}(1). \end{aligned} \tag{4.9}$$

Thus $|\hat{f}(y)| = |\hat{f}(1)| = 1$ whenever $\gcd(y, n) = 1$. By Parseval's identity (2.5) it follows that $\hat{f}(y) = 0$ otherwise. We can set $\hat{A} = (\mathbb{Z}/n\mathbb{Z})^*$ with size $|\hat{A}| = \varphi(n)$, making $\hat{\alpha} = \varphi(n)/n$ and $\hat{r} = \hat{R} = 1$. By theorem 4.5, the success probability of finding s is then

$$p(s) = \left(\frac{\hat{r}}{R}\right)^2 \left| \hat{\alpha} - \frac{1}{|G|^{3/2}} \sum_{y \in \hat{A}} \sum_{x \notin A+s} \phi_y(x)\overline{\phi_y(s)} \frac{g(x)}{\hat{f}(y)} \right|^2 = \hat{\alpha}^2 = \left(\frac{\varphi(n)}{n}\right)^2.$$

Note that the double sum vanishes because $g(x) = f(x - s) = 0$ whenever $x \notin A + s$.

Finite field characters. Let $q = p^k$ be a prime power and $\mathbb{F}_q$ be the finite field with q elements. Consider a multiplicative character $f : \mathbb{F}_q^* \to \mathbb{C}$ that is extended to $\mathbb{F}_q$ with $f(0) = 0$. To apply algorithm 4.4, we let $A = \mathbb{F}_q^*$ such that f is zero outside A and we can set $r = R = 1$.

The additive characters of $\mathbb{F}_q$ are given by $\phi_y : x \mapsto e^{2\pi i \operatorname{Tr}(xy)/p}$, where $\operatorname{Tr} : \mathbb{F}_q \to \mathbb{F}_p$ is the trace map given by $\operatorname{Tr}(x) = \sum_{j=0}^{k-1} x^{p^j}$. By the same argument as for Dirichlet characters we know that $\hat{f}(\phi_y) = \overline{f(y)}\hat{f}(1)$, meaning we can also take $\hat{A} = \mathbb{F}_q^*$ with $\hat{r} = \hat{R} = 1$ and $\hat{f}$ being zero outside $\hat{A}$. The probability of finding the hidden shift s by algorithm 4.4 is thus given by $p(s) = \hat{\alpha}^2 = (1 - \frac{1}{q})^2$. This is the same probability as obtained in [vDHI06].

4.4 Symmetry of the algorithm

The success probability in theorem 4.5 is not symmetric in the parameters $r, R, \hat{r}, \hat{R}, \alpha, \hat{\alpha}$. Here we explain why and give a mirrored version of algorithm 4.4.

Whenever g and f are not completely bent functions, we introduced parameters R and $\hat{r}$ to separate the superposition into the good and bad part, see eq. (4.1). To get the amplitude as close to $\hat{g}(\phi)/\hat{f}(\phi) = \hat{\delta}_s(\phi) = \phi(s)$ as possible (and thereby learn s) we used the approximation

$$\hat{\delta}_s(\phi) \approx \frac{\hat{g}(\phi)}{R} \frac{\hat{r}}{\hat{f}(\phi)}.$$

We hereby bounded g from above and $\hat{f}$ from below. One can instead use the inverse of this approximation given by

$$\hat{\delta}_s(\phi)^{-1} \approx \frac{r}{\hat{g}(\phi)} \frac{\hat{f}(\phi)}{\hat{R}},$$

where we need to bound g and $\hat{f}$ from the other side. To flip the roles of $\hat{f}$ and g and therefore to approximate $\phi(s)^{-1} = \overline{\phi(s)}$, we can run algorithm 4.4 with the following changes.

- We replace U_1 from eq. (4.2) with $U_1 = \sum_w |w\rangle\langle w| \otimes U_1(w)$, where

$$U_1(w)|0\rangle = \begin{cases} \frac{r}{w}|0\rangle + \sqrt{1 - |r/w|^2}|1\rangle & \text{if } w \neq 0, \\ |0\rangle & \text{if } w = 0. \end{cases}$$

- We replace U_2 from eq. (4.3) with $U_2 = \sum_w |w\rangle\langle w| \otimes U_2(w)$ where

$$U_2(w)^\dagger|0\rangle = \frac{\bar{w}}{\hat{R}}|0\rangle + \sqrt{1 - |w/\hat{R}|^2}|1\rangle.$$

- We apply on the first register between the two calls of O_g and also before the final measurement an extra operator $U_- \in \mathrm{U}(\mathbb{C}^G)$ acting via

$$U_-|x\rangle = |-x\rangle \text{ for all } x \in G.$$

  The rest of the algorithm stays the same.

We can then redo the proof of theorem 4.5 up to eq. (4.8) by making the changes

$$\frac{g(x)}{R} \mapsto \frac{r}{g(-x)}, \qquad \frac{\hat{r}}{\hat{f}(\phi)} \mapsto \frac{\hat{f}(\phi)}{\hat{R}}.$$

Afterwards we can use $\hat{A} \subset \hat{G}$ to split the sum and find that the the modified algorithm has success probability

$$p(s) = \left(\frac{r}{\hat{R}}\right)^2 \left| \alpha - \frac{1}{|G|^{3/2}} \sum_{\phi \notin \hat{A}} \sum_{-x \in A+s} \phi(x)\overline{\phi(-s)} \frac{\hat{f}(\phi)}{g(-x)} \right|^2.$$

Remark 4.6. This symmetry is part of the reason we include all parameters $(r, R, \hat{r}, \hat{R}, \alpha, \hat{\alpha})$, even though not all of them appear in the success probability. Another (slightly hidden) dependence of parameters in eq. (4.5) lies in the summations over $A + s$ and $\hat{A}$, which are determined by α and $\hat{\alpha}$ respectively. Lastly, all the parameters can be used to give an upper bound on the double summation in eq. (4.5).

5 Multidimensional bent functions

An even greater generalization of bent functions was proposed by Poinsot [Poi05] which includes functions from a finite abelian group to any hermitian space. We restrict ourselves to functions with codomain $\mathbb{C}^d$ for some dimension $d \geq 1$.

Definition 5.1 (Multidimensional Fourier transform)**.** Let $f : G \to \mathbb{C}^d$ be a multidimensional complex function. Writing the function coordinate-wise as $f(x) = (f_0(x), \ldots, f_{d-1}(x))$, the Fourier transform of f is given by

$$\hat{f}(\phi) = (\hat{f}_0(\phi), \ldots, \hat{f}_{d-1}(\phi)) \text{ for all } \phi \in \hat{G},$$

where each $\hat{f}_i$ is the one-dimensional Fourier transform from definition 2.1.

Definition 5.2 (Multidimensional bent function)**.** A multidimensional complex function $f : G \to \mathbb{C}^d$ is called *bent* if $\|f(x)\| = 1$ for all $x \in G$ and $\|\hat{f}(\phi)\| = 1$ for all $\phi \in \hat{G}$.

Example 5.3. While a multidimensional bent function has unit norm everywhere, this may fail for its one-dimensional parts. For example, writing $\omega = e^{2\pi i/3}$, this happens for the function $f = (f_0, f_1)$ where $f_i : \mathbb{Z}/3\mathbb{Z} \to \mathbb{C}$ are given by

	0	1	2
$f_0(x)$	1	$\frac{\omega+\omega^2}{2}$	1
$f_1(x)$	0	$\frac{\omega-\omega^2}{2}$	0

	0	1	2
$\hat{f}_0(\phi)$	$\frac{\sqrt{3}}{2}$	$-\frac{\sqrt{3}\omega}{2}$	$-\frac{\sqrt{3}\omega^2}{2}$
$\hat{f}_1(\phi)$	$\frac{\omega-\omega^2}{2\sqrt{3}}$	$\frac{\omega^2-1}{2\sqrt{3}}$	$\frac{1-\omega}{2\sqrt{3}}$

.

See appendix B for more discussion and examples.

5.1 Exact algorithm

Consider a multidimensional bent function $f : G \to \mathbb{C}^d$ on a finite abelian group G. A function $g : G \to \mathbb{C}^d$ hides a shift $s \in G$ if $g(x) = f(x - s)$ for all $x \in G$. We present an exact quantum algorithm, similar to algorithm 3.5, to solve the hidden shift problem, assuming that we have access to the quantum oracles O_g and $O_{\hat{f}}$ given by eqs. (1.1) and (1.2). One ingredient will be unitary operators $\tilde{O}_g \in \mathrm{U}(\mathbb{C}^G \otimes \mathbb{C}^d)$ and $\tilde{O}_{\hat{f}} \in \mathrm{U}(\mathbb{C}^{\hat{G}} \otimes \mathbb{C}^d)$ satisfying

$$\tilde{O}_g|x\rangle|0\rangle = |x\rangle \sum_{i=0}^{d-1} g_i(x)|i\rangle, \qquad \tilde{O}_{\hat{f}}|\phi\rangle|0\rangle = |\phi\rangle \sum_{i=0}^{d-1} \hat{f}_i(\phi)|i\rangle.$$

One can view $\tilde{O}_g$ and $\tilde{O}_{\hat{f}}^\dagger$ as multidimensional versions of the phase oracles defined in eq. (3.1) and used in algorithm 3.5.

Lemma 5.4. *Let $f : G \to \mathbb{C}^d$ be a multidimensional complex function such that $\|f(x)\| = 1$ for all $x \in G$. Given access to oracle $O_f \in \mathrm{U}(\mathbb{C}^G \otimes \mathbb{C}^{2^n})$ implementing the transformation $|x\rangle|b\rangle \mapsto |x\rangle|b \oplus f(x)\rangle$ for all $x \in G$ and $b \in \{0,1\}^n$, one can implement an operator $\tilde{O}_f \in \mathrm{U}(\mathbb{C}^G \otimes \mathbb{C}^d)$ satisfying $\tilde{O}_f|x\rangle|0\rangle = |x\rangle \sum_{i=0}^{d-1} f_i(x)|i\rangle$ using two calls to O_f.*

Proof. We start from the state $|x\rangle|0\rangle|0\rangle \in \mathbb{C}^G \otimes \mathbb{C}^{2^n} \otimes \mathbb{C}^d$, where the n-qubit register in the middle stores $f(x)$ as an n-bit string. Recall that we assumed in section 1.1 to have a perfect encoding of all values $f(x) \in \mathbb{C}^d$ using n qubits. We call O_f to obtain the state $|x\rangle|f(x)\rangle|0\rangle$. In a new auxiliary register, we then compute a classical description of a unitary operator $S \in \mathrm{U}(\mathbb{C}^d)$, expressed as a sequence of Givens rotations, satisfying $S|0\rangle = \sum_{i=0}^{d-1} f_i(x)|i\rangle$. Through a sequence of controlled operations, S is applied to the quantum state of the third register, after which we uncompute the data describing S in the auxiliary register. This results in the state $|x\rangle|f(x)\rangle \sum_{i=0}^{d-1} f_i(x)|i\rangle$. Finally, we call O_f again to uncompute the second register. □

Algorithm 5.5.

1. *Prepare the uniform superposition $|G\rangle|0\rangle = \frac{1}{|G|^{1/2}} \sum_{x\in G} |x\rangle|0\rangle$.*

2. *Apply the operator $(F^\dagger \otimes I)\tilde{O}_{\hat{f}}^\dagger(F \otimes I)\tilde{O}_g$ to transform the state as*

$$\begin{aligned}\frac{1}{|G|^{1/2}} \sum_{x\in G} |x\rangle|0\rangle &\xmapsto{\tilde{O}_g} \frac{1}{|G|^{1/2}} \sum_{x\in G} |x\rangle \otimes \left(\sum_{i=0}^{d-1} g_i(x)|i\rangle\right) \\ &\xmapsto{F\otimes I} \frac{1}{|G|^{1/2}} \sum_{\phi\in\hat{G}} |\phi\rangle \otimes \phi(s) \left(\sum_{i=0}^{d-1} \hat{f}_i(x)|i\rangle\right) \\ &\xmapsto{\tilde{O}_{\hat{f}}^\dagger} \frac{1}{|G|^{1/2}} \sum_{\phi\in\hat{G}} |\phi\rangle \otimes \phi(s)|0\rangle \xmapsto{F^\dagger\otimes I} |s\rangle|0\rangle.\end{aligned}$$

3. *Measure the first register in the standard basis.*

Theorem 5.6. *For a multidimensional bent function $f : G \to \mathbb{C}^d$, algorithm 5.5 determines the hidden shift s with certainty.*

Remark 5.7. The above algorithm also covers the one-dimensional case from algorithm 3.5. By taking $d = 1$ the values $g(x)$ and $\hat{f}(\phi)$ lie on the unit circle for all $x \in G, \phi \in \hat{G}$. Then the second register can be discarded, so $\tilde{O}_g$ and $\tilde{O}_{\hat{f}}^\dagger$ reduce to the phase oracles from eq. (3.1). The algorithm then works the same way and returns the hidden shift with certainty.

5.2 Approximate algorithm

We have generalized the one-dimensional algorithm 3.5 for bent functions in two different ways.

1. Algorithm 4.4 allows for function values outside the unit circle. This was done by renormalizing the function and preparing an ancillary qubit in a superposition of a 'good' part and a 'bad' part:

$$\underbrace{\frac{g(x)}{R}|0\rangle}_{\text{good}} + \underbrace{\sqrt{1 - \left|\frac{g(x)}{R}\right|^2}|1\rangle}_{\text{bad}}.$$

2. Algorithm 5.5 allows for multidimensional bent functions. Where algorithm 3.5 used $g(x)$ and $\hat{f}(\phi)$ as global phases, i.e. one-dimensional rotations, we now see the vector valued functions as rotations in $\mathbb{C}^d$.

We will now combine these two generalizations and describe an approximate algorithm for bent-like functions in higher dimensions. The approximate algorithm is therefore very similar to algorithm 4.4, but with an added rotation to reduce to the one-dimensional case. We assume the functions to satisfy definition 1.11.

We work with the state space $\mathbb{C}^G \otimes \mathbb{C}^2 \otimes \mathbb{C}^{2^n} \otimes \mathbb{C}^d \otimes \mathbb{C}^2$, where the third register acts as extra workspace to store the vector of function values and will be returned to its original state by the algorithm. We assume that we have access to the following quantum operators.

- Oracle access to the functions g and $\hat{f}$ via eqs. (1.1) and (1.2).
- For ease of notation we write $|w_0, \dots, w_{d-1}\rangle = |w\rangle \in \mathbb{C}^{2^n}$. Note the different dimensions of $w \in \mathbb{C}^d$ and $|w\rangle \in \mathbb{C}^{2^n}$, as we assumed in section 1.1 there is a perfect encoding of the d-dimensional values $g(x)$ in n qubits. We assume that we have three unitary operators

$$U_1, V_{\text{rot}} \in \mathrm{U}(\mathbb{C}^{2^n} \otimes \mathbb{C}^d), U_2 \in \mathrm{U}(\mathbb{C}^{2^n} \otimes \mathbb{C}^2)$$

given by

$$U_1 = \sum_w |w\rangle\langle w| \otimes U_1(w), \text{ where } U_1(w)|0\rangle = \sum_{i=0}^{d-1} \frac{w_i}{R}|i\rangle + \sqrt{1 - \frac{\|w\|^2}{R^2}}|d\rangle, \tag{5.1}$$

$$V_{\text{rot}} = \sum_w |w\rangle\langle w| \otimes V_{\text{rot}}(w), \text{ where } V_{\text{rot}}(w)^\dagger|0\rangle = \begin{cases} \frac{1}{\|w\|}\sum_{i=0}^{d-1} w_i|i\rangle & \text{if } w \neq 0, \\ |0\rangle & \text{if } w = 0, \end{cases} \tag{5.2}$$

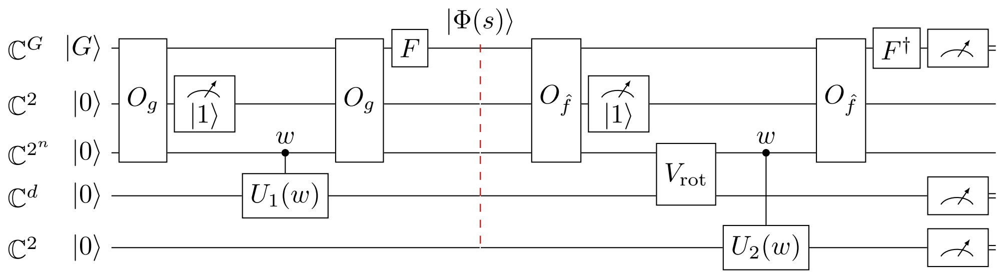
$$U_2 = \sum_w |w\rangle\langle w| \otimes U_2(w), \text{ where } U_2(w)^\dagger|0\rangle = \begin{cases} \frac{\hat{r}}{\|w\|}|0\rangle + \sqrt{1 - \frac{\hat{r}^2}{\|w\|^2}}|d\rangle & \text{if } w \neq 0, \\ |0\rangle & \text{if } w = 0. \end{cases} \tag{5.3}$$

- The Fourier transform on the register $\mathbb{C}^G$.

These operators have many similarities to those in section 4.2. The main difference is that the operator from eq. (4.2) is now split into U_1 and V_{rot} to account for the multiple dimensions. First we will use U_1, then apply the Fourier transform in each coordinate and then rotate backwards to put all the information in the first coordinate. This inverse rotation reduces it to a one-dimensional problem. The algorithm works on the space $\mathbb{C}^G \otimes \mathbb{C}^2 \otimes \mathbb{C}^{2^n} \otimes \mathbb{C}^d \otimes \mathbb{C}^2$ as follows.

Algorithm 5.8.

1. *Prepare the uniform superposition* $|G, 0, 0, 0, 0\rangle := \frac{1}{|G|^{1/2}} \sum_{x \in G} |x, 0, 0, 0, 0\rangle$.
2. *Perform the following quantum circuit.*


$|\Phi(s)\rangle$
$\mathbb{C}^G$ $|G\rangle$
$\mathbb{C}^2$ $|0\rangle$
$\mathbb{C}^{2^n}$ $|0\rangle$
$\mathbb{C}^d$ $|0\rangle$
$\mathbb{C}^2$ $|0\rangle$
O_g
$|1\rangle$
w
$U_1(w)$
O_g
F
$O_{\hat{f}}$
$|1\rangle$
V_{rot}
w
$U_2(w)$
$O_{\hat{f}}$
$F^\dagger$

Output FAIL *if either of the two intermediate measurements that post-select the second register to* $|1\rangle$ *fail.*

3. *Measure the first, third and fourth register in the standard basis. If the latter two measure* 0*, output the value of the first register. Otherwise output* FAIL.

Note the similarities with algorithm 4.4.

Theorem 5.9. *For an $(r, R, \hat{r}, \hat{R}, \alpha, \hat{\alpha})$-bounded function $f : G \to \mathbb{C}^d$, the probability of finding the hidden shift s by algorithm 5.8 is*

$$p(s) = \left(\frac{\hat{r}}{R}\right)^2 \left| \hat{\alpha} - \frac{1}{|G|^{3/2}} \sum_{\phi \in \hat{A}} \sum_{x \notin A+s} \phi(x) \sum_{i=0}^{d-1} \frac{g_i(x)\overline{\hat{f}_i(\phi)}}{\|\hat{f}(\phi)\|} \right|^2,$$

and it uses 2 calls to both the oracles O_g and $O_{\hat{f}}$.

Proof. The proof is very similar to the one of theorem 4.5. For ease of notation we denote

$$|g_0(1), \ldots, g_{d-1}\rangle = |g(x)\rangle, \text{ and } |\hat{f}_0(\phi), \ldots, \hat{f}_{d-1}(\phi)\rangle = |\hat{f}(\phi)\rangle.$$

To correctly identify the hidden shift s, we would like to observe the state $|s, 0, 0, 0, 0\rangle$ at the end. As we did for the proofs of theorems 4.3 and 4.5, we define $|\Phi(s)\rangle$ to be the state at the marked time point (assuming the intermediate measurement do not fail). Let $|\Psi(s)\rangle$ be the state at the same point obtained by running the circuit backwards starting with the target state $|s, 0, 0, 0, 0\rangle$ (also assuming the intermediate measurement does not fail). As this time point is reached with probability α and $\hat{\alpha}$ from the left and right respectively, the total probability is given by

$$p(s) = \alpha\hat{\alpha} \left|\langle\Psi(s)|\Phi(s)\rangle\right|^2.$$

Similarly to eq. (4.6) it holds that

$$|\Phi(s)\rangle = \frac{1}{|A|^{1/2}|G|^{1/2}} \sum_{\phi \in \hat{G}} \sum_{x \in A+s} \phi(x)|\phi, 0, 0\rangle \left(\sum_{i=0}^{d-1} \frac{g_i(x)}{R}|i\rangle + \sqrt{1 - \left|\frac{\|g(x)\|}{R}\right|^2}|d\rangle \right)|0\rangle.$$

Starting with the target state $|s, 0, 0, 0, 0\rangle$ and reading from the right, we obtain a state similar to eq. (4.7), only with an extra $V_{\mathrm{rot}}^\dagger$ applied to the fourth register (conditional on $\hat{f}(\phi) \neq 0$). Note that in theorem 4.5 this register is untouched by the right side of the circuit. The result is

$$|\Psi(s)\rangle = \frac{1}{|\hat{A}|^{1/2}} \sum_{\phi \in \hat{A}} \phi(s)|\phi\rangle|0, 0\rangle \left(\sum_{i=0}^{d-1} \frac{\hat{f}_i(\phi)}{\|\hat{f}(\phi)\|}|i\rangle \right) \left(\frac{\hat{r}}{\|\hat{f}(\phi)\|)}|0\rangle + \sqrt{1 - \left|\frac{\hat{r}}{\|\hat{f}\|}\right|^2}|1\rangle \right).$$

The total probability is thus given by

$$\begin{aligned}
p(s) &= \alpha\hat{\alpha} \left|\langle\Psi(s)|\Phi(s)\rangle\right|^2 \\
&= \alpha\hat{\alpha} \left| \frac{1}{|\hat{A}|^{1/2}|A|^{1/2}|G|^{1/2}} \sum_{\phi \in \hat{A}} \sum_{x \in A+} \left(\phi(x)\overline{\phi(s)} \sum_{i=0}^{d-1} \frac{g_i(x)\overline{\hat{f}_i(\phi)}}{\|\hat{f}(\phi)\|^2} \right) \right|^2 \\
&= \left(\frac{\hat{r}}{R}\right)^2 \left| \frac{1}{|G|^{3/2}} \sum_{\phi \in \hat{A}} \sum_{x \in G} \left(\phi(x)\overline{\phi(s)} \sum_{i=0}^{d-1} \frac{g_i(x)\overline{\hat{f}_i(\phi)}}{\|\hat{f}(\phi)\|^2} \right) - \right. \\
&\qquad\qquad \left. \frac{1}{|G|^{3/2}} \sum_{\phi \in \hat{A}} \sum_{x \notin A+s} \left(\phi(x)\overline{\phi(s)} \sum_{i=0}^{d-1} \frac{g_i(x)\overline{\hat{f}_i(\phi)}}{\|\hat{f}(\phi)\|^2} \right) \right|^2 \\
&= \left(\frac{\hat{r}}{R}\right)^2 \left| \frac{1}{|G|} \sum_{\phi \in \hat{A}} \left(\sum_{i=0}^{d-1} \frac{\overline{\phi(s)}\hat{g}_i(x)\overline{\hat{f}_i(\phi)}}{\|\hat{f}(\phi)\|^2} \right) - \frac{1}{|G|^{3/2}} \sum_{\phi \in \hat{A}} \sum_{x \notin A+s} \left(\overline{\phi(s)} \sum_{i=0}^{d-1} \frac{g_i(x)\overline{\hat{f}_i(\phi)}}{\|\hat{f}(\phi)\|^2} \right) \right|^2 \\
&= \left(\frac{\hat{r}}{R}\right)^2 \left| \hat{\alpha} - \frac{1}{|G|^{3/2}} \sum_{\phi \in \hat{A}} \sum_{x \notin A+s} \left(\overline{\phi(s)} \sum_{i=0}^{d-1} \frac{g_i(x)\overline{\hat{f}_i(\phi)}}{\|\hat{f}(\phi)\|^2} \right) \right|^2.
\end{aligned}$$

□

6 One-register approach

Recall from section 4.1 that at the end of algorithm 4.2 the measurement on both ancilla registers had to produce 0 for the algorithm to succeed. In this section, we consider a variation of this algorithm where we reduce the number of ancilla registers from two to one and include additional degrees of freedom within both oracles. Since both oracles now use the same ancilla register, this yields a richer interference pattern and also slightly reduces quantum memory. Here we investigate how this modification affects the overall success probability of the algorithm.

Recall that algorithm 4.2 from section 4.1 prepared states

$$\frac{g(x)}{R}|0\rangle + \sqrt{1-\left(\frac{g(x)}{R}\right)^2}|1\rangle \quad \text{and} \quad \frac{\hat{r}}{\hat{f}(\phi)}|0\rangle + \sqrt{1-\left(\frac{\hat{r}}{\hat{f}(\phi)}\right)^2}|1\rangle$$

on two different registers, and for both states we want as much amplitude as possible in the first basis state $|0\rangle$. However, we can try to improve the success probability of algorithm 4.2 by also making use of what was considered in eq. (4.1) as error or the 'bad part'. The modified algorithm works by letting the unitary operators U_1 and U_2 from eqs. (4.2) and (4.3) act on the same ancilla register, see algorithm 6.1. Previously these operators did not interact directly, so some of their degrees of freedom were not relevant, whereas now they could potentially be used to our advantage. We study the success probability of algorithm 6.1 in general and determine necessary and sufficient conditions for it to be equal to 1.

6.1 Oracles

Assume that f is an $(R,\hat{r})$-bounded function, and that we have quantum access to the oracles of g and $\hat{f}$ as in eqs. (1.1) and (1.2). Combining O_g with the unitary U_1 given in eq. (4.2), we obtain a unitary oracle $U_g \in \mathrm{U}(\mathbb{C}^G \otimes \mathbb{C}^2)$ that is given by

$$U_g = \sum_{x\in G} |x\rangle\langle x| \otimes \begin{pmatrix} b_0(x) & * \\ b_1(x) & * \end{pmatrix}, \qquad b_0(x) = \frac{g(x)}{R},\ b_1(x) = e^{i\theta(x)}\sqrt{1-\left|\frac{g(x)}{R}\right|^2}, \tag{6.1}$$

where we have introduced a phase function $\theta : G \to [0.2\pi)$ to take advantage of the additional degree of freedom. The remaining entries indicated by $*$ in eq. (6.1) can be chosen arbitrarily, so long as each 2×2 block is unitary. The actual choice (which involves an additional phase degree of freedom) will not matter to us. A similar construction can be done using $O_{\hat{f}}$ to obtain a unitary $U_{1/\hat{f}} \in \mathrm{U}(\mathbb{C}^{\hat{G}} \otimes \mathbb{C}^2)$ given by

$$U_{1/\hat{f}} = \sum_{\phi\in \hat{G}} |\phi\rangle\langle \phi| \otimes \begin{pmatrix} a_0(\phi) & a_1(\phi) \\ * & * \end{pmatrix}, \qquad a_0(\phi) = \frac{\hat{r}}{\hat{f}(\phi)},\ a_1(\phi) = e^{-i\chi(\phi)}\sqrt{1-\left|\frac{\hat{r}}{\hat{f}(\phi)}\right|^2}, \tag{6.2}$$

where we also introduced a phase function $\phi : \hat{G} \to [0,2\pi)$. For convenience, we let

$$\langle a(\phi)| = \begin{pmatrix} a_0(\phi) & a_1(\phi) \end{pmatrix}, \qquad\qquad |b(x)\rangle = \begin{pmatrix} b_0(x) \\ b_1(x) \end{pmatrix}.$$

Then the action of the oracles from eqs. (6.1) and (6.2) is captured by

$$\langle\phi|\langle 0| \cdot U_{1/\hat{f}} = \langle\phi|\langle a(\phi)| = \langle\phi|\left(\frac{\hat{r}}{\hat{f}(\phi)}\langle 0| + e^{-i\chi(\phi)}\sqrt{1-\left|\frac{\hat{r}}{\hat{f}(\phi)}\right|^2}\langle 1|\right) \qquad \text{for all } \phi \in \hat{G}, \tag{6.3}$$

$$U_g \cdot |x\rangle|0\rangle = |x\rangle|b(x)\rangle = |x\rangle\left(\frac{g(x)}{R}|0\rangle + e^{i\theta(x)}\sqrt{1-\left|\frac{g(x)}{R}\right|^2}|1\rangle\right) \qquad \text{for all } x \in G. \tag{6.4}$$

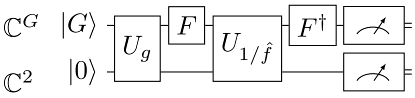
6.2 Approximate algorithm

The algorithm for solving the hidden shift problem for an $(R,\hat{r})$-bounded function f uses the oracles $U_{1/\hat{f}}$ and U_g defined above. The algorithm works on the space $\mathbb{C}^G \otimes \mathbb{C}^2$.

Algorithm 6.1.

1. *Prepare the state* $|G\rangle|0\rangle = \frac{1}{|G|^{1/2}}\sum_{x\in G}|x\rangle|0\rangle$.

2. *Apply the operator* $(F^\dagger \otimes I)U_{1/\hat{f}}(F\otimes I)U_g$*:*



3. *Measure both registers in the standard basis. If the second register contains* 0*, output the value of the first register. Otherwise, output* FAIL.

Theorem 6.2. *For an* $(R, \hat{r})$*-bounded function* $f : G \to \mathbb{C}$*, the probability of finding the hidden shift* s *by algorithm 6.1 is*

$$p(s) = \left| \frac{\hat{r}}{R} + \frac{1}{|G|^{3/2}} \sum_{x \in G} \sum_{\phi \in \hat{G}} \phi(x) e^{i\theta(x+s)} \sqrt{1 - \left|\frac{f(x)}{R}\right|^2} e^{-i\chi(\phi)} \sqrt{1 - \left|\frac{\hat{r}}{\hat{f}(\phi)}\right|^2} \right|^2. \tag{6.5}$$

Proof. We follow the proof of theorem 4.3. The success probability of algorithm 6.1 is given by

$$p(s) = \Big| \underbrace{\big(\langle s| \otimes \langle 0|\big)(F^\dagger \otimes I) U_{1/\hat{f}}}_{\langle \Psi(s)|} \underbrace{(F \otimes I) U_g \big(|G\rangle \otimes |0\rangle\big)}_{|\Phi(s)\rangle} \Big|^2,$$

which we will compute by separately evaluating $\langle \Psi(s)|$ and $|\Phi(s)\rangle$. By eqs. (6.3) and (6.4), we have

$$\langle \Psi(s)| = \big(\langle s|F^\dagger \otimes \langle 0|\big) U_{1/\hat{f}} = \frac{1}{|G|^{1/2}} \sum_{\phi \in \hat{G}} \overline{\phi(s)} \langle \phi| \otimes \langle a(\phi)|,$$

$$|\Phi(s)\rangle = (F \otimes I) U_g \left(\frac{1}{|G|^{1/2}} \sum_{x \in G} |x\rangle \otimes |0\rangle \right) = \frac{1}{|G|^{1/2}} \sum_{x \in G} F|x\rangle \otimes |b(x)\rangle.$$

Recall from eq. (2.2) that $\langle \phi|F|x\rangle = \phi(x)/|G|^{1/2}$. Combined with the expressions for $\langle a(\phi)|$ and $|b(x)\rangle$ from eqs. (6.1) and (6.2) the inner product between these two states is

$$\begin{aligned}
\langle \Psi(s)|\Phi(s)\rangle &= \frac{1}{|G|} \sum_{x \in G} \sum_{\phi \in \hat{G}} \overline{\phi(s)} \langle \phi|F|x\rangle \langle a(\phi)|b(x)\rangle = \frac{1}{|G|^{3/2}} \sum_{x \in G} \sum_{\phi \in \hat{G}} \overline{\phi(s)} \phi(x) \langle a(\phi)|b(x)\rangle \\
&= \frac{1}{|G|^{3/2}} \sum_{x \in G} \sum_{\phi \in \hat{G}} \overline{\phi(s)} \phi(x) \left(\frac{\hat{r}}{R} \frac{g(x)}{\hat{f}(\phi)} + e^{i\theta(x)} \sqrt{1 - \left|\frac{g(x)}{R}\right|^2} e^{-i\chi(\phi)} \sqrt{1 - \left|\frac{\hat{r}}{\hat{f}(\phi)}\right|^2} \right) \\
&= \frac{\hat{r}}{R} + \frac{1}{|G|^{3/2}} \sum_{x \in G} \sum_{\phi \in \hat{G}} \overline{\phi(s)} \phi(x) e^{i\theta(x)} \sqrt{1 - \left|\frac{g(x)}{R}\right|^2} e^{-i\chi(\phi)} \sqrt{1 - \left|\frac{\hat{r}}{\hat{f}(\phi)}\right|^2}.
\end{aligned}$$

The last equality follows from the proof of theorem 4.3. Note that the first term in this expression is the total probability obtained from algorithm 4.2. The desired formula follows by noting that $\overline{\phi(s)}\phi(x) = \phi(x - s)$ and substituting $x \mapsto x + s$. □

Next, we investigate conditions under which algorithm 6.1 gives the shift s with certainty. To this end, we introduce the function

$$\begin{aligned}
h_s &: G \to \mathbb{C} \\
x &\mapsto e^{i\theta(x+s)} \sqrt{1 - \left|\frac{f(x)}{R}\right|^2}
\end{aligned} \tag{6.6}$$

and its Fourier transform $\hat{h}_s : \hat{G} \to \mathbb{C}$. We can express eq. (6.5) in terms of $\hat{h}_s$ as

$$p(s) = \left| \frac{\hat{r}}{R} + Z_s \right|^2, \tag{6.7}$$

where

$$Z_s = \frac{1}{|G|} \sum_{\phi \in \hat{G}} e^{-i\chi(\phi)} \sqrt{1 - \left|\frac{\hat{r}}{\hat{f}(\phi)}\right|^2} \hat{h}_s(\phi). \tag{6.8}$$

For later use, note that by definition of h_s and a double application of Parseval's identity (2.5) we have

$$\sum_{\phi\in\hat{G}} |\hat{h}_s(\phi)|^2 = \sum_{x\in G}\left(1-\left|\frac{f(x)}{R}\right|^2\right) = |G| - \frac{1}{R^2}\sum_{\phi\in\hat{G}}\left|\hat{f}(\phi)\right|^2. \tag{6.9}$$

Theorem 6.3. *Suppose the hidden shift equals s. Then the probability that algorithm 6.1 outputs the correct shift (namely, s) equals 1 if and only if one of the following conditions holds:*

1. *$\hat{r} = R$;*
2. *$\hat{r} < R$, and it holds that*

$$|\hat{f}(\phi)| = \sqrt{\hat{r}R} \quad \textit{for all } \phi\in\hat{G}, \tag{6.10}$$

and

$$\hat{h}_s(\phi) = e^{i\chi(\phi)}\sqrt{1-\frac{\hat{r}}{R}} \quad \textit{for all } \phi\in\hat{G}. \tag{6.11}$$

Proof. By proposition 4.1(2) we have $\hat{r}\le R$. In case equality holds, proposition 4.1(2) gives $|f(x)| = R$ and $|\hat{f}(\phi)| = \hat{r}$ for all $x\in G$ and $\phi\in\hat{G}$. By eq. (6.5), we immediately deduce $p(s) = 1$. Now assume that we are in the case $\hat{r} < R$. If for all $\phi\in\hat{G}$ we have $|\hat{f}(\phi)| = \sqrt{\hat{r}R}$ and $\hat{h}_s(\phi) = e^{i\chi(\phi)}\sqrt{1-\frac{\hat{r}}{R}}$, then by eqs. (6.7) and (6.8) we obtain $Z_s = 1-\hat{r}/R$ and hence $p(s) = 1$.

Conversely, assume $p(s) = 1$, or equivalently

$$Z_s = 1 - \hat{r}/R. \tag{6.12}$$

The Cauchy–Schwarz inequality together with eq. (6.9) implies

$$\begin{aligned}|Z_s|^2 &\le \left(\frac{1}{|G|}\sum_{\phi\in\hat{G}}\left(1-\left|\frac{\hat{r}}{\hat{f}(\phi)}\right|^2\right)\right)\left(\frac{1}{|G|}\sum_{\phi\in\hat{G}}|\hat{h}_s(\phi)|^2\right)\\ &= \left(1-\frac{\hat{r}^2}{|G|}\sum_{\phi\in\hat{G}}\frac{1}{|\hat{f}(\phi)|^2}\right)\left(1-\frac{1}{|G|\cdot R^2}\sum_{\phi\in\hat{G}}\left|\hat{f}(\phi)\right|^2\right)\\ &= \left(1-\frac{\hat{r}^2}{H}\right)\left(1-\frac{A}{R^2}\right),\end{aligned}$$

where

$$A = \frac{\sum_{\phi\in\hat{G}}|\hat{f}(\phi)|^2}{|G|} \quad\text{and}\quad H = \frac{|G|}{\sum_{\phi\in\hat{G}}\frac{1}{|\hat{f}(\phi)|^2}}.$$

By the arithmetic-harmonic mean inequality one has $H\le A$, with equality if and only if $|\hat{f}(\phi)|^2$ is constant over all $\phi\in\hat{G}$, in which case $|\hat{f}(\phi)| = \sqrt{H}$ for all $\phi\in\hat{G}$. Furthermore, since $Z_s = 1-\hat{r}/R > 0$ and $H\ge\hat{r}^2$, we have $1-\hat{r}^2/H > 0$. Therefore it holds that

$$|Z_s|^2 \le \left(1-\frac{\hat{r}^2}{H}\right)\left(1-\frac{H}{R^2}\right) = 1-\left(\frac{\hat{r}^2}{H}+\frac{H}{R^2}\right)+\frac{\hat{r}^2}{R^2}.$$

By the arithmetic-geometric mean inequality we have

$$\frac{\hat{r}^2}{H}+\frac{H}{R^2}\ge 2\frac{\hat{r}}{R},$$

with equality if and only if $H = \hat{r}R$. We obtain

$$|Z_s|^2 \le 1 - 2\frac{\hat{r}}{R}+\frac{\hat{r}^2}{R^2} = \left(1-\frac{\hat{r}}{R}\right)^2.$$

In view of eq. (6.12), it follows that H indeed equals $\hat{r}R$, so eq. (6.10) holds.

It remains to prove eq. (6.11). To do this, we first note that by eqs. (6.8), (6.10) and (6.12) we have

$$\sqrt{1-\frac{\hat{r}}{R}} = \frac{1}{|G|}\sum_{\phi\in\hat{G}} e^{-i\chi(\phi)}\hat{h}_s(\phi).$$

This implies

$$\sqrt{1-\frac{\hat{r}}{R}} \leq \frac{1}{|G|}\sum_{\phi\in\hat{G}} |\hat{h}_s(\phi)|,$$

with equality if and only if $e^{-i\chi(\phi)}\hat{h}_s(\phi)$ is real and non-negative for all $\phi \in \hat{G}$. Next, the arithmetic-quadratic mean inequality together with eqs. (6.9) and (6.10) gives

$$1-\frac{\hat{r}}{R} \leq \frac{1}{|G|}\sum_{\phi\in\hat{G}} |\hat{h}_s(\phi)|^2 = 1-\frac{\hat{r}}{R},$$

with equality if and only if in addition all $|\hat{h}_s(\phi)|$ are equal, in which case the common value equals $\sqrt{1-\hat{r}/R}$. This proves eq. (6.11). □

Remark 6.4. In case 1 of theorem 6.3, the function f is bent by proposition 4.1(2), and algorithm 6.1 reduces to algorithm 3.5. The conditions in case 2 of theorem 6.3 are quite remarkable. Specifically, eq. (6.10) implies that f is not only $(R,\hat{r})$-bounded but even $(R,\sqrt{\hat{r}R})$-bounded. However, we need to treat f as an $(R,\hat{r})$-bounded function in order to attain succes probability 1 in algorithm 6.1.

The conditions of theorem 6.3 are dependent on the shift s. Since this shift is actually the value we want to determine, it is natural to ask under what conditions we can attain $p(s) = 1$ for *all* values of s.

Lemma 6.5. *Let $u : G \to \mathbb{R}_{\geq 0}$ be a function, and let $\hat{u} : \hat{G} \to \mathbb{C}$ be its Fourier transform. Then $|\hat{u}(\phi)|$ is constant for all $\phi \in \hat{G}$ if and only if there exists $x_0 \in G$ such that $u(x) = 0$ for all $x \neq x_0$. In this case we have*

$$\hat{u}(\phi) = \frac{1}{\sqrt{|G|}}u(x_0)\phi(x_0) \quad \textit{for all } \phi \in \hat{G}. \tag{6.13}$$

Proof. Suppose all $|\hat{u}(\phi)|$ are equal. Since the claim clearly holds if u is the zero function, we assume that u is not the zero function. We write

$$U = \sum_{x\in G} u(x) > 0.$$

Let $\phi \in \hat{G}$. By the definition of the Fourier transform, the triangle inequality and the assumption that $u(x) \geq 0$ for all $x \in G$ we can write

$$|\hat{u}(\phi)| = \frac{1}{|G|}\left|\sum_{x\in G} u(x)\phi(x)\right| \leq \frac{U}{|G|} = |\hat{u}(\phi_0)|. \tag{6.14}$$

By assumption, the inequality in eq. (6.14) is in fact an equality. This implies

$$\left|\sum_{x\in G} u(x)\phi(x)\right| = U,$$

so we have a convex combination of $|G|$ points on the unit circle giving a point on the unit circle, namely

$$\left|\sum_{x\in G} \frac{u(x)}{U}\phi(x)\right| = 1.$$

This can only hold for all $\phi \in \hat{G}$ if there exists $x_0 \in G$ such that $u(x) = 0$ for all $x \neq x_0$. Conversely, if there is $x_0 \in G$ such that $u(x) = 0$ for all $x \neq x_0$, then eq. (6.13) holds, and hence we have $|\hat{u}(\phi)| = |u(x_0)|/\sqrt{|G|}$ for all $\phi \in \hat{G}$. □

Theorem 6.6. *We have $p(s) = 1$ for all $s \in G$ if and only if one of the following conditions holds:*

1. *$\hat{r} = R$;*

2. $1 - \frac{1}{|G|} \leq \frac{\hat{r}}{R} < 1$, *and there exist* $x_0 \in G$ *and* $\alpha \in \mathbb{R}$ *such that for all* $x \in G$ *and* $\phi \in \hat{G}$ *it holds that*

$$\begin{aligned} |\hat{f}(\phi)| &= \sqrt{\hat{r}R}, \\ |f(x)| &= \begin{cases} R\sqrt{1 - |G|\left(1 - \frac{\hat{r}}{R}\right)} & \textit{if } x = x_0, \\ R & \textit{if } x \neq x_0, \end{cases} \\ \theta(x) &= \alpha, \\ \chi(\phi) &= \alpha + \arg(\phi(x_0)). \end{aligned}$$

Proof. In the case $\hat{r} = R$, the claim follows directly from theorem 6.3. Now suppose that we are in the case $\hat{r} < R$. First suppose that the conditions in item 2 hold. Then we compute

$$h_s(x) = \begin{cases} e^{i\alpha}\sqrt{|G|}\sqrt{1 - \frac{\hat{r}}{R}} & \text{if } x = x_0, \\ 0 & \text{if } x \neq x_0. \end{cases}$$

This implies that for all $\phi \in \hat{G}$ we have

$$\hat{h}_s(\phi) = \phi(x_0)e^{i\alpha}\sqrt{1 - \frac{\hat{r}}{R}} = e^{i\chi(\phi)}\sqrt{1 - \frac{\hat{r}}{R}}.$$

By theorem 6.3, it follows that $p(s)$ equals 1 for all $s \in G$.

Conversely, suppose $\hat{r} < R$ and $p(s) = 1$ for all $s \in G$. By theorem 6.3, eqs. (6.10) and (6.11) are satisfied for all $s \in G$, and we need to show that the conditions in item 2 hold. By eq. (6.11), the function $\hat{h}_s$ is non-zero and independent of s, and therefore the same holds for h_s. It follows that θ is constant, say

$$\theta(x) = \alpha \quad \text{for all } x \in G.$$

Thus we have

$$h_s(x) = e^{i\alpha}u(x) \quad \text{with} \quad u(x) = \sqrt{1 - \left|\frac{f(x)}{R}\right|^2}.$$

By eq. (6.11) the functions $\hat{h}_s$ and therefore also $\hat{u}$ have constant absolute value, so lemma 6.5 implies that there exists $x_0 \in G$ such that

$$|f(x)| = R \quad \text{for all } x \neq x_0.$$

It then follows from Parseval's identity (2.5) and eq. (6.10) that $(|G| - 1)R^2 \geq |G|\hat{r}R$, or equivalently

$$\frac{\hat{r}}{R} \geq 1 - \frac{1}{|G|},$$

and

$$|f(x_0)| = R\sqrt{1 - |G|\left(1 - \frac{\hat{r}}{R}\right)}.$$

We then compute

$$h_s(x) = \begin{cases} e^{i\alpha}\sqrt{|G|}\sqrt{1 - \frac{\hat{r}}{R}} & \text{if } x = x_0, \\ 0 & \text{if } x \neq x_0, \end{cases}$$

and subsequently

$$\hat{h}_s(\phi) = \frac{1}{\sqrt{|G|}}\phi(x_0)h(x_0) = \phi(x_0)e^{i\alpha}\sqrt{1 - \frac{\hat{r}}{R}}.$$

Comparing this to eq. (6.11), we obtain

$$\chi(\phi) = \alpha + \arg(\phi(x_0)) \quad \text{for all } \phi \in \hat{G},$$

which concludes the proof. □

Example 6.7. We give explicit examples of families of functions satisfying the conditions of theorem 6.6 for the groups $G = \mathbb{Z}/n\mathbb{Z}$ with $n \in \{2, 3\}$. We identify $\hat{G}$ with G via the isomorphism $\phi : \mathbb{Z}/n\mathbb{Z} \to \hat{G}$ defined by $\phi(a)(x) = \exp(2\pi i a x/n)$.

For $n = 2$, we fix $\eta \in \mathbb{C}$ with $|\eta| = 1$ and $\operatorname{Re} \eta \le 0$. We consider the Fourier transform pair

x	0	1
$f(x)$	$\frac{1+\eta}{\sqrt{2}}$	$\frac{1-\eta}{\sqrt{2}}$
$\hat{f}(x)$	1	η

This satisfies the conditions of theorem 6.6 with $R = 1/\hat{r} = |1 - \eta|/\sqrt{2}$.

For $n = 3$, we fix $\eta \in \mathbb{C}$ with $|\eta| = 1$ and $\operatorname{Re} \eta \le 1/2$. We consider the Fourier transform pair

x	0	1	2
$f(x)$	$\frac{1+2\eta}{\sqrt{3}}$	$\frac{1-\eta}{\sqrt{3}}$	$\frac{1-\eta}{\sqrt{3}}$
$\hat{f}(x)$	1	η	η

This satisfies the conditions of theorem 6.6 with $R = 1/\hat{r} = |1 - \eta|/\sqrt{3}$.

6.3 Lower bounds for the success probability

We have shown that algorithm 6.1 can only attain success probability 1 under the conditions of theorems 6.3 and 6.6. Since these conditions are rather strict, and those of theorem 6.3 moreover depend on the unknown shift s, it is natural to ask for lower bounds on the success probability $p(s)$ given by eq. (6.5), using the freedom that we have in the choice of the functions θ and χ.

Theorem 6.8. *Consider algorithm 6.1 with $\theta = 0$. For any value of the hidden shift s, the choice of χ that maximizes the success probability of measuring s is $\chi(\phi) = \arg(\hat{h}(\phi))$ for all $\phi \in \hat{G}$ with $\hat{h}(\phi) \neq 0$ (and $\chi(\phi)$ is arbitrary for $\hat{h}(\phi) = 0$). The success probability in this case equals*

$$p = \left(\frac{\hat{r}}{R} + |G|^{-1} \sum_{\phi} \sqrt{1 - \left| \frac{\hat{r}}{\hat{f}(\phi)} \right|^2} \, |\hat{h}(\phi)| \right)^2$$

with $h(x) = \sqrt{1 - |f(x)/R|^2}$.

Proof. Substituting $\theta = 0$, we find that eqs. (6.6) to (6.8) simplify to

$$p(s) = \left| \frac{\hat{r}}{R} + \frac{1}{|G|} \sum_{\phi \in \hat{G}} e^{-i\chi(\phi)} \sqrt{1 - \left| \frac{\hat{r}}{\hat{f}(\phi)} \right|^2} \, \hat{h}(\phi) \right|^2 .$$

This expression is independent of s and is maximal when each term in the sum is real and positive, which happens for the choice of χ described in the theorem. □

Note that theorem 6.8 only gives the optimal choice for χ when we fix $\theta = 0$; we do not claim that these choices are optimal among all choices of θ and χ. Furthermore, since this choice for χ seems difficult to compute efficiently, we will next explore the case where $\theta = 0$ and where we let χ take uniformly random values in $[0, 2\pi)$.

Lemma 6.9. *For any two complex numbers a and b, we have*

$$\frac{1}{2\pi} \int_0^{2\pi} \left| a + b e^{i\alpha} \right|^2 d\alpha = |a|^2 + |b|^2.$$

Proof. We have

$$\left| a + b e^{i\alpha} \right|^2 = |a|^2 + \bar{a} b e^{i\alpha} + a \bar{b} e^{-i\alpha} + |b|^2.$$

Integrating the right-hand side over $[0, 2\pi]$ gives $2\pi(|a|^2 + |b|^2)$. □

Lemma 6.9 admits the following higher-dimensional generalisation.

Lemma 6.10. *Let a and $b_1, \dots, b_n$ (with $n \geq 1$) be complex numbers. One has*

$$\frac{1}{(2\pi)^n}\int_0^{2\pi}\cdots\int_0^{2\pi}\left|a + b_1 e^{i\alpha_1} + \cdots + b_n e^{i\alpha_n}\right|^2 d\alpha_1 \cdots d\alpha_n = |a|^2 + |b_1|^2 + \cdots + |b_n|^2.$$

Proof. We use induction on n; the case $n = 0$ is immediate and the induction step follows from lemma 6.9. □

Theorem 6.11. *Assume that $\theta = 0$ and the images $\chi(\phi)$ of the phase function χ are chosen independently and uniformly random in the interval $[0, 2\pi)$. Then the probability of obtaining the shift s is independent of s and is given by*

$$p = \left(\frac{\hat{r}}{R}\right)^2 + \frac{1}{|G|^2}\sum_{\phi \in \hat{G}}\left(1 - \left|\frac{\hat{r}}{\hat{f}(\phi)}\right|^2\right)\left|\hat{h}(\phi)\right|^2.$$

Proof. For all $\phi \in \hat{G}$, we set

$$F(\phi) = \frac{1}{|G|^{3/2}}\sum_{x \in G}\sqrt{1 - \left|\frac{\hat{r}}{\hat{f}(\phi)}\right|^2}\sqrt{1 - \left|\frac{f(x)}{R}\right|^2}\phi(x).$$

By theorem 6.2, the conditional probability of obtaining the shift s given χ equals

$$P(|s\rangle|0\rangle \mid \chi) = \left|\frac{\hat{r}}{R} + \sum_{\phi \in \hat{G}} F(\phi)e^{i\chi(\phi)}\right|^2.$$

Using the law of total probability, the probability $P(|s\rangle|0\rangle)$ writes

$$P(|s\rangle|0\rangle) = \int_\chi P(|s\rangle|0\rangle \mid \chi)d\chi.$$

We evaluate the latter using lemma 6.10 to get

$$\begin{aligned}
P(|s\rangle|0\rangle) &= \left(\frac{\hat{r}}{R}\right)^2 + \sum_{\phi \in \hat{G}}|F(\phi)|^2 \\
&= \left(\frac{\hat{r}}{R}\right)^2 + \frac{1}{|G|^2}\sum_{\phi \in \hat{G}}\left(1 - \left|\frac{\hat{r}}{\hat{f}(\phi)}\right|^2\right)\left|\frac{1}{|G|^{1/2}}\sum_{x \in G}\sqrt{1 - \left|\frac{f(x)}{R}\right|^2}\phi(x)\right|^2 \\
&= \left(\frac{\hat{r}}{R}\right)^2 + \frac{1}{|G|^2}\sum_{\phi \in \hat{G}}\left(1 - \left|\frac{\hat{r}}{\hat{f}(\phi)}\right|^2\right)\left|\hat{h}(\phi)\right|^2.
\end{aligned}$$

□

Finally, we consider the case where both the images $\theta(x)$ of the phase function θ and the images $\chi(\phi)$ of the phase function χ are chosen independently and uniformly randomly in the interval $[0, 2\pi)$.

Theorem 6.12. *Assume that the images $\chi(\phi)$ and $\theta(x)$ of the respective phase functions χ and θ are chosen independently and uniformly randomly in the interval $[0, 2\pi)$. Then the probability of obtaining the shift s is independent of s and is given by*

$$p = \left(\frac{\hat{r}}{R}\right)^2 + \frac{1}{|G|^3}\sum_{(\phi,x) \in (\hat{G},G)}\left(1 - \left|\frac{\hat{r}}{\hat{f}(\phi)}\right|^2\right)\left(1 - \left|\frac{f(x)}{R}\right|^2\right).$$

Proof. Note that one can generalize lemma 6.10 as follows: if a and $\{b_{j,k}\}_{j,k=1}^n$ are also complex numbers, then

$$\frac{1}{(2\pi)^{2n}}\int_0^{2\pi}\cdots\int_0^{2\pi}\int_0^{2\pi}\cdots\int_0^{2\pi}\left|a + \sum_{j,k} b_{j,k}e^{i\alpha_j}e^{i\beta_k}\right|^2 d\alpha_1 \cdots d\alpha_n d\beta_1 \cdots d\beta_n = |a|^2 + \sum_{j,k}|b_{j,k}|^2. \tag{6.15}$$

By the law of total probability, the probability $P(|s\rangle|0\rangle)$ is given by

$$P(|s\rangle|0\rangle) = \int_\theta \int_\chi P(|s\rangle|0\rangle \mid (\chi,\theta)) d\chi d\theta.$$

By proceeding as before and using eq. (6.15), we can infer that

$$P(|s\rangle|0\rangle) = \left(\frac{\hat{r}}{R}\right)^2 + \sum_{(\phi,x)\in(\hat{G},G)} |F(\phi,x)|^2$$

with

$$F(\phi,x) = \frac{1}{|G|^{3/2}} \sqrt{1 - \left|\frac{\hat{r}}{\hat{f}(\phi)}\right|^2} \sqrt{1 - \left|\frac{f(x)}{R}\right|^2}. \qquad \square$$

7 Discussion

Improving the success probability via amplitude amplification

We can use amplitude amplification [BHMT02] to increase the 'good' part of the output state in our approximate algorithms 4.2, 4.4, 5.8 and 6.1, and thus the success probability of finding the hidden shift. To do so, we assume that we are in one of the cases corresponding to theorems 4.3, 4.5, 5.9, 6.8, 6.11 and 6.12, where the success probability p is known. In the case of algorithms 4.4 and 5.8, the algorithms need to be adapted slightly by storing the output of the checks for $x \in A+s$ and $\phi \in \hat{A}$ (after the first application of O_g and $O_{\hat{f}}$, respectively) in separate qubits. Instead of measuring these qubits and the last two registers, we use them to define the 'good' states for the purpose of amplitude amplification.

We denote by $\mathcal{A}$ the unitary operator corresponding to the whole algorithm, excluding measurements. Let S_χ be the operator that sends every good basis state $|\psi\rangle$ to $-|\psi\rangle$ and fixes the bad basis states. Let S_0 be the operator that sends the initial state $|0\rangle$ to $-|0\rangle$ and fixes all other basis states. The amplification operator is then given by

$$Q = -\mathcal{A}S_0\mathcal{A}^\dagger S_\chi.$$

Taking θ such that $\sin^2(\theta) = p$, we need $\lfloor \pi/4\theta \rfloor = \Theta\left(\frac{1}{\sqrt{p}}\right)$ applications of the amplification operator Q, and thus of the operators $\mathcal{A}$ and $\mathcal{A}^\dagger$, starting with the state $\mathcal{A}|0\rangle$.

Oracle for the Fourier transform

A major weakness of our approach is that in problem 1.13 we require oracle access to the Fourier transform $\hat{f}$ of f. Indeed, in practice it might be expensive to compute $\hat{f}$ from a classical description of f, and thus difficult to implement the oracle $O_{\hat{f}}$ for $\hat{f}$. Nevertheless, in some cases the function f and its Fourier transform are related in a simple way, allowing for an efficient implementation of $O_{\hat{f}}$ from access to O_f. For example, if f is a primitive Dirichlet character or a multiplicative character of a finite field, then $\hat{f}(y) = \overline{f(y)}\hat{f}(1)$, see eq. (4.9). It would be nice if one could find ways of overcoming this limitation more generally.

Results on multidimensional bent functions

In the one-dimensional case, bent functions have been studied in various settings and under various names such as biunimodular vectors, see remark 1.5. However, not much is known about multidimensional bent functions [Poi05] defined in definition 5.2 and discussed further in appendix B. In particular, while biunimodular vectors have been classified for small $|G|$ [FR15], including $|G| = 13$ [GS02], no classification or even non-trivial constructions of multidimensional bent functions are known.

Classification of bent functions for small abelian groups can indicate the 'density' of bent functions among all complex functions. The closer a given function is to a bent function, the higher the success probability of our algorithms. The more bent functions there are, the more functions are *close* to one, thus having formal results on bent function density would inform us on how well our algorithms work in general.

Acknowledgments

SA thanks the Quantum Software Consortium programme for research stay at Leiden University. PB was partially supported by the Dutch Research Council (NWO/OCW) as part of the Quantum Software Consortium programme (project number 024.003.037). MO was partially supported by the Dutch Research Council (NWO) Vidi grant (No.VI.Vidi.192.109). JS was supported by the Dutch Research Council (NWO) as part of the Quantum Software Consortium programme (project number 024.003.037) and the Quantum Delta NL programme (project number NGF.1623.23.023).

References

[AA15] Scott Aaronson and Andris Ambainis. Forrelation: A problem that optimally separates quantum from classical computing. In *Proceedings of ACM STOC'2015*, pages 307–316, 2015. doi:10.1145/2746539.2746547.

[AS24] Matthew Amy and Lucas Shigeru Stinchcombe. Polynomial-time classical simulation of hidden shift circuits via confluent rewriting of symbolic sums, 2024. arXiv:2408.02778.

[BCM19] John J. Benedetto, Katherine Cordwell, and Mark Magsino. CAZAC sequences and Haagerup's characterization of cyclic n-roots. In Akram Aldroubi, Carlos Cabrelli, Stéphane Jaffard, and Ursula Molter, editors, *New Trends in Applied Harmonic Analysis, Volume 2: Harmonic Analysis, Geometric Measure Theory, and Applications*, pages 1–43. Springer International Publishing, Cham, 2019. doi:10.1007/978-3-030-32353-0_1.

[BHMT02] Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. In *Quantum computation and information (Washington, DC, 2000)*, volume 305 of *Contemp. Math.*, pages 53–74. Amer. Math. Soc., Providence, RI, 2002. doi:10.1090/conm/305/05215.

[CKOR13] Andrew M. Childs, Robin Kothari, Maris Ozols, and Martin Roetteler. Easy and hard functions for the boolean hidden shift problem. In *8th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2013)*, volume 22 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 50–79, Dagstuhl, Germany, 2013. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.TQC.2013.50.

[CvD10] Andrew M. Childs and Wim van Dam. Quantum algorithms for algebraic problems. *Reviews of Modern Physics*, 82(1):1–52, January 2010. doi:10.1103/revmodphys.82.1.

[DM24] Suman Dutta and Subhamoy Maitra. Introducing nega-forrelation: quantum algorithms in analyzing nega-hadamard and nega-crosscorrelation spectra. *Designs, Codes and Cryptography*, 92(3):863–883, 2024. doi:10.1007/s10623-023-01346-x.

[FIM$^+$14] Katalin Friedl, Gábor Ivanyos, Frédéric Magniez, Miklos Santha, and Pranab Sen. Hidden translation and translating coset in quantum computing. *SIAM Journal on Computing*, 43(1):1–24, 2014. doi:10.1137/130907203.

[FR15] Hartmut Führ and Ziemowit Rzeszotnik. On biunimodular vectors for unitary matrices. *Linear Algebra and its Applications*, 484:86–129, 2015. doi:10.1016/j.laa.2015.06.019.

[Gro96] Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, 1996. doi:10.1145/237814.237866.

[GRR11] Dmitry Gavinsky, Martin Roetteler, and Jérémie Roland. Quantum algorithm for the boolean hidden shift problem. In *Computing and Combinatorics: 17th Annual International Conference, COCOON 2011, Dallas, TX, USA, August 14-16, 2011. Proceedings 17*, pages 158–167. Springer, 2011. doi:10.1007/978-3-642-22685-4_14.

[GS02] Ernst M. Gabidulin and Vitaly V. Shorin. New sequences with zero autocorrelation. *Problems of Information Transmission*, 38:255–267, 2002. doi:10.1023/A:1022093728009.

[HJ12] Roger A. Horn and Charles R. Johnson. *Matrix analysis*. Cambridge university press, 2 edition, 2012.

[Kit95] A. Yu. Kitaev. Quantum measurements and the abelian stabilizer problem, 1995. arXiv:quant-ph/9511026.

[Kob87] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, 48(177):203–209, January 1987. doi:10.2307/2007884.

[Kup05] Greg Kuperberg. A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. *SIAM Journal on Computing*, 35(1):170–188, 2005. doi:10.1137/S0097539703436345.

[Kup13] Greg Kuperberg. Another subexponential-time quantum algorithm for the dihedral hidden subgroup problem. In *8th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2013)*, volume 22 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 20–34, Dagstuhl, Germany, 2013. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.TQC.2013.20.

[Mes16] Sihem Mesnager. *Bent Functions: Fundamentals and Results*. Springer, 2016. doi:10.1007/978-3-319-32595-8.

[Mil85] Victor S. Miller. Use of elliptic curves in cryptography. In *Advances in Cryptology (CRYPTO 1985)*, pages 417–426, Berlin, Heidelberg, 1985. Springer Berlin Heidelberg. doi:10.1007/3-540-39799-X_31.

[NC12] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, June 2012. doi:10.1017/cbo9780511976667.

[Pei20] Chris Peikert. He Gives C-Sieves on the CSIDH. In *Advances in Cryptology (EUROCRYPT 2020)*, page 463–492. Springer, 2020. doi:10.1007/978-3-030-45724-2_16.

[Poi05] Laurent Poinsot. Multidimensional bent functions. *GESTS International Transactions on Computer Science and Engeneering*, 18(1):185–195, October 2005. URL: https://hal.science/hal-00460339.

[Reg02] Oded Regev. Quantum computation and lattice problems. In *The 43rd Annual IEEE Symposium on Foundations of Computer Science*, SFCS-02, page 520–529. IEEE Comput. Soc, 2002. doi:10.1109/sfcs.2002.1181976.

[Reg04] Oded Regev. A subexponential time algorithm for the dihedral hidden subgroup problem with polynomial space. Preprint, 2004. arXiv:quant-ph/0406151.

[Röt09] Martin Rötteler. Quantum algorithms to solve the hidden shift problem for quadratics and for functions of large Gowers norm. In *Mathematical Foundations of Computer Science 2009*, pages 663–674. Springer Berlin Heidelberg, 2009. doi:10.1007/978-3-642-03816-7_56.

[Röt10] Martin Rötteler. Quantum algorithms for highly non-linear Boolean functions. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 448–457. SIAM, January 2010. doi:10.1137/1.9781611973075.37.

[RSA83] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 26(1):96–99, January 1983. doi:10.1145/357980.358017.

[Sho94] Peter W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, SFCS-94, page 124–134. IEEE Comput. Soc. Press, 1994. doi:10.1109/sfcs.1994.365700.

[Sho99] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review*, 41(2):303–332, January 1999. doi:10.1137/s0036144598347011.

[Sim97] Daniel R. Simon. On the power of quantum computation. *SIAM Journal on Computing*, 26(5):1474–1483, October 1997. doi:10.1137/s0097539796298637.

[Tok15] Natalia Tokareva. *Bent Functions: Results and Applications to Cryptography*. Academic Press, 2015. URL: https://archive.org/details/bentfunctionsres0000toka/, doi:10.1016/C2014-0-02922-X.

[vDHI06] Wim van Dam, Sean Hallgren, and Lawrence Ip. Quantum algorithms for some hidden shift problems. *SIAM Journal on Computing*, 36(3):763–778, January 2006. doi:10.1137/s009753970343141x.

A Error analysis

Throughout the article, it was assumed that the oracles

$$O_g : |x\rangle|0\rangle \to |x\rangle|g(x)\rangle,\ O_{\hat{f}} : |\phi\rangle|0\rangle \mapsto |\phi\rangle|\hat{f}(\phi)\rangle$$

give access to the complex function value $g(x)$ without any approximation needed. In this section, we calculate the error of algorithm 4.2 when we take the approximation into account. Specifically, we assume that any number $y \in \mathbb{C}$ can be represented with $2n$ bits by $(\arg(y), |y|)$, where both real numbers are approximated up to precision $\delta = \frac{1}{2^n}$.

Proposition A.1. *Let $f : G \to \mathbb{C}$ be an $(r, R, \hat{r}, \hat{R}, A, \hat{A})$-bounded function and assume complex numbers can be stored up to precision $\delta = \frac{1}{2^n}$ in their argument and modulus. Assume that $n \gg \log(|G|)$. Then there exists a positive real number C such that algorithm 4.4 with error finds the hidden shift with probability*

$$p(s) - C|G|^{1/2}\delta \leq p_e(s) \leq p(s) + C|G|^{1/2}\delta,$$

where $p(s)$ is the probability without approximation given in eq. (4.5).

Proof. We follow the notation from the proof of theorem 4.5. By considering oracle access to the functions g/R and $\hat{r}/\hat{f}$, we may assume that all complex numbers have a modulus between 0 and 1. This means that invoking the oracle O_g, applying the unitary U_1 from equation eq. (4.2) and then applying O_g again introduces an error via

$$|\Phi(s)\rangle = \frac{1}{|A|^{1/2}|G|^{1/2}} \sum_{\phi \in \hat{G}} \sum_{x \in A+s} \phi(x)|\phi, 0, 0\rangle \left(\left(\frac{g(x)}{R} + \varepsilon(x) \right) |0\rangle + \left(\sqrt{1 - \left| \frac{g(x)}{R} \right|^2} + \varepsilon'(x) \right) |1\rangle \right) |0\rangle.$$

A similar error holds for $O_{\hat{f}}$ in combination with U_2 from eq. (4.3), which we will denote by $\hat{\varepsilon}(\phi)$. Reading the circuit of algorithm 4.4 from the right gives the state

$$|\Psi(s)\rangle = \frac{1}{|\hat{A}|^{1/2}} \sum_{\phi \in \hat{A}} \phi(s)|\phi, 0, 0, 0\rangle \left(\left(\frac{\hat{r}}{\overline{\hat{f}(\phi)}} + \hat{\varepsilon}(\phi) \right) |0\rangle + \left(\sqrt{1 - \left| \frac{\hat{r}}{\overline{\hat{f}(\phi)}} \right|^2} + \hat{\varepsilon}'(\phi) \right) |1\rangle \right).$$

The errors $\varepsilon(x)$ and $\hat{\varepsilon}(\phi)$ are complex number with a small norm. In particular, there is a constant $C_1 > 0$ such that $|\varepsilon(x)| < C_1\delta$ and $|\hat{\varepsilon}(\phi)| < C_1$ for all $x \in G, \phi \in \hat{G}$. Following the proof of theorem 4.5 the inner product becomes

$$\begin{aligned}
\langle\Psi(s)|\Phi(s)\rangle &= \frac{1}{|\hat{A}|^{1/2}|A|^{1/2}|G|^{1/2}} \sum_{\phi\in\hat{A}} \sum_{x\in A+s} \phi(x)\overline{\phi(s)} \left(\frac{g(x)}{R} + \varepsilon(x)\right) \left(\frac{\hat{r}}{\hat{f}(\phi)} + \hat{\varepsilon}(\phi)\right) \\
&= \frac{1}{|\hat{A}|^{1/2}|A|^{1/2}|G|^{1/2}} \sum_{\phi\in\hat{A}} \sum_{x\in A+s} \phi(x)\overline{\phi(s)} \left(\frac{g(x)\hat{r}}{\hat{f}(\phi)R} + \hat{\varepsilon}(\phi)\frac{g}{R} + \varepsilon(x)\frac{\hat{r}}{\hat{f}(\phi)} + \varepsilon(x)\hat{\varepsilon}(\phi)\right) \\
&= \frac{1}{|\hat{A}|^{1/2}|A|^{1/2}|G|^{1/2}} \sum_{\phi\in\hat{A}} \sum_{x\in A+s} \phi(x)\overline{\phi(s)} \frac{g(x)\hat{r}}{\hat{f}(\phi)R} + \frac{|\hat{A}|^{1/2}|A|^{1/2}}{|G|^{1/2}} O(\varepsilon). \\
&= S + \frac{1}{|\hat{A}|^{1/2}|A|^{1/2}|G|^{1/2}} \sum_{\phi\in\hat{A}} \sum_{x\in A+s} \phi(x)\overline{\phi(s)} \left(\hat{\varepsilon}(\phi)\frac{g}{R} + \varepsilon(x)\frac{\hat{r}}{\hat{f}(\phi)} + \varepsilon(x)\hat{\varepsilon}(\phi)\right) \\
&= S + E,
\end{aligned}$$

where S is the sum in eq. (4.8) obtained in the version without approximation. The double sum denoted by E is a complex number of which we can estimate the norm. Using the triangle inequality and the fact that $|A|, |\hat{A}| \leq |G|$ as well as $\left|\frac{g(x)}{R}\right|, \left|\frac{\hat{r}}{\hat{f}(\phi)}\right| \leq 1$ we obtain

$$\begin{aligned}
|E| &= \left|\frac{1}{|\hat{A}|^{1/2}|A|^{1/2}|G|^{1/2}} \sum_{\phi\in\hat{A}} \sum_{x\in A+s} \phi(x)\overline{\phi(s)} \left(\hat{\varepsilon}(\phi)\frac{g}{R} + \varepsilon(x)\frac{\hat{r}}{\hat{f}(\phi)} + \varepsilon(x)\hat{\varepsilon}(\phi)\right)\right| \\
&\leq \frac{1}{|A|^{1/2}|\hat{A}|^{1/2}|G|^{1/2}} \sum_{\phi\in\hat{A}} \sum_{x\in A+s} (|\varepsilon(x)| + |\hat{\varepsilon}(\phi)| + |\varepsilon(x)||\hat{\varepsilon}(\phi)|) \\
&\leq |G|^{1/2}(2C_1\delta + C_1^2\delta^2) \leq |G|^{1/2}C_2\delta,
\end{aligned}$$

for some other constant $C_2 > 0$ under the assumption that $\delta = \frac{1}{2^n}$ is small enough. The total probability to find the hidden shift s with approximation errors is given by

$$p_e(s) = \alpha\hat{\alpha}\left|S + E\right|^2.$$

Using the (inverse) triangle inequality we find that

$$\begin{aligned}
p_e(s) &\geq \alpha\hat{\alpha}(|S| - |E|)^2 = p(s) - 2p(s)^{1/2}(\alpha\hat{\alpha})^{1/2}|E| + \alpha\hat{\alpha}|E|^2 \geq p(s) - C|G|^{1/2}\delta, \\
p_e(s) &\leq \alpha\hat{\alpha}(|S| + |E|)^2 = p(s) + 2p(s)^{1/2}(\alpha\hat{\alpha})^{1/2}|E| + \alpha\hat{\alpha}|E|^2 \leq p(s) + C|G|^{1/2}\delta,
\end{aligned}$$

for some new constant $C > 0$. □

B Multidimensional bent functions

In section 5 we presented a quantum algorithm for the hidden shift problem of multidimensional complex bent functions. However, very little is known about the existence and classification of these functions. In this section, we present some examples of multidimensional bent functions and discuss their properties.

The only reference on multidimensional bent functions known to us [Poi05] provides two constructions:

1. The *concatenation construction*. Let $f_i : G \to \mathbb{C}$ with $1 \leq i \leq d$ be a family of 1-dimensional bent functions. Then for any vector $u \in \mathbb{C}^d$ of norm 1 the function

$$\begin{aligned}
f : G &\to \mathbb{C}^d, \\
x &\mapsto (u_1 f_1(x), \ldots, u_d f_d(x))
\end{aligned}$$

   is d-dimensionally bent.

2. The *disjoint support construction*. The function $f : G \to \mathbb{C}^{|G|}$ given by $f(g) = e_g$ is a $|G|$-dimensional bent function.

The concatenation construction can be extended to arbitrary dimensions.

Lemma B.1 (Concatenation construction)**.** *Let $f^{(i)} : G \to \mathbb{C}^{d_i}$, $1 \leq i \leq n$ be a collection of bent functions and let $u \in \mathbb{C}^n$ be a vector such that $\|u\|^2 = 1$. Then*

$$\begin{aligned} f : G &\to \mathbb{C}^{d_1+\cdots+d_n}, \\ x &\mapsto u_1 f^{(1)}(x) \oplus \cdots \oplus u_n f^{(n)}(x) \end{aligned}$$

is an $(d_1 + \cdots + d_n)$-dimensional bent function.

Proof. The norm of $f(x)$ is given by

$$\|f(x)\|^2 = \sum_{i=1}^{n} u_i^2 \|f^{(i)}\|^2 = \sum_{i=1}^{n} u_i^2 = 1.$$

As the Fourier transform is performed coordinate-wise, the norm of $\hat{f}(\phi)$ is given by

$$\|\hat{f}(\phi)\|^2 = \sum_{i=1}^{n} u_i^2 \|\hat{f}^{(i)}\|^2 = \sum_{i=1}^{n} u_i^2 = 1,$$

from which we can conclude that f is bent. □

When having the hidden shift problem in mind, the concatenation construction does not give any interesting bent functions. These functions are already bent (up to a scalar) in the first coordinate. The hidden shift can then be found using the one-dimensional algorithm. In fact, the hidden shift problem for many multidimensional bent functions can be reduced to the one-dimensional case.

Lemma B.2. *Let G be a finite abelian group. The action of $\mathrm{U}(\mathbb{C}^d)$ on the space of d-dimensional bent functions $f : G \to \mathbb{C}^d$ given by*

$$(Uf)(x) := U \begin{pmatrix} f_1(x) \\ \vdots \\ f_d(x) \end{pmatrix} \in \mathbb{C}^d \text{ for all } U \in \mathrm{U}(\mathbb{C}^d)$$

defines an equivalence relation.

Proof. For a unitary matrix U, the norm $\|Uf(x)\|$ equals 1 for all $x \in G$. Because

$$\widehat{Uf}(\phi) = \frac{1}{|G|^{1/2}} \sum_{x \in G} \phi(x)(Uf)(x) = U \frac{1}{|G|^{1/2}} \sum_{x \in G} \phi(x) f(x) = U\hat{f}(\phi)$$

it holds that $\|\widehat{Uf}(\phi)\| = 1$ for all $\phi \in \hat{G}$, making Uf a bent function. We relate f and g if there is a $U \in \mathrm{U}(\mathbb{C}^d)$ such that $f = Ug$. This is an equivalence relation because $\mathrm{U}(\mathbb{C}^d)$ contains the identity matrix and multiplicative inverses, and is closed under multiplication. □

Definition B.3 (Equivalence of multidimensional bent functions)**.** Let (G, e_G) be a finite abelian group and $d \geq 1$ an integer. Define $B_d(G)$ to be the set of equivalence classes of d-dimensional bent functions on G.

Definition B.4 (Concatenated multidimensional bent function)**.** We call an element $f \in B_d(G)$ *concatenated* if it is equivalent to a bent function obtained from the concatenation construction.

Example B.5. Let $G = \mathbb{Z}/3\mathbb{Z}$ and recall the 2-dimensional bent function f given in example 5.3. This function is equivalent to

$$f' := Uf, \text{ with } U = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}, \quad \begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline f_1'(x) & \frac{1}{\sqrt{2}} & \frac{\omega}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ f_2'(x) & \frac{1}{\sqrt{2}} & \frac{\omega^2}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{array}.$$

The new function f' is bent in both coordinates (scaled with $\frac{1}{\sqrt{2}}$) and thus a concatenated bent function.

The above example implies that the quantum algorithms in section 5 are not necessary for the hidden shift problem of this function, because applying the correct unitary reduces it to the 1-dimensional case. The following question then arises: are all multidimensional bent functions concatenated?

B.1 Gram matrices

In order to answer the question posed above, we need to study bent functions up to a suitable notion of equivalence. As the action of unitary matrices can be seen as a basis change on $\mathbb{C}^d$, it will be easier to study the Gram matrix of a bent function. We first give a key property of multidimensional bent functions and then make an identification with Gram matrices.

Lemma B.6 ([Poi05, Proposition 2])**.** *Let $f : G \to \mathbb{C}^d$ be a complex function for which $\|f(x)\| = 1$ for all $x \in G$. Then f is bent if and only if $\sum_{x\in G} f(x)\overline{f(x+a)} = 0$ for all $a \neq e_G$.*

Definition B.7. Define $C_d(G)$ to be the set of $|G| \times |G|$ matrices M that satisfy

1. M is a hermitian and positive semidefinite matrix of rank $\leq d$,
2. $M_{x,x} = 1$ for all $x \in G$,
3. M satisfies $\sum_{x\in G} M_{x,x+a} = 0$ when $a \neq e_G$.

Proposition B.8. *For a finite abelian group G and an integer $d \geq 1$, there is a one-to-one correspondence between $B_d(G)$ and $C_d(G)$.*

Proof. Let $f : G \to \mathbb{C}^d$ be a bent function. Consider the Gram matrix M obtained from the d vectors $(f_i(x))_{x\in G} \in \mathbb{C}^{|G|}$. That is,

$$M = F^\dagger F, \text{ where } F = \begin{pmatrix} \dots & f_1(x) & \dots \\ \dots & f_2(x) & \dots \\ & \vdots & \\ \dots & f_d(x) & \dots \end{pmatrix}.$$

By construction, M is a Gram matrix of a set of d row vectors and thus hermitian and positive semidefinite. The entries can be written as $M_{xy} = \sum_{i=1}^d f_i(x)\overline{f_i(y)}$. Because f is a bent function, it follows that $M_{x,x} = 1$ for all $x \in G$. By lemma B.6, the last criterion is also satisfied so that $M \in C_d(G)$. The action of a unitary matrix $U \in \mathrm{U}(\mathbb{C}^d)$ on f changes the column vectors of F to those of UF. The Gram matrix then remains

$$(UF)^\dagger UF = F^\dagger U^\dagger UF = F^\dagger F = M.$$

We conclude that the map

$$\phi : B_d(G) \to C_d(G) : \quad f \mapsto F^\dagger F$$

is well defined.

Conversely, any hermitian positive semidefinite matrix of rank $\leq d$ is the Gram matrix $M = F^\dagger F$ of a set of d vectors in $\mathbb{C}^{|G|}$ (if the rank is lower than d we can add zeros as rows). We can now construct a function by defining f as the column vectors of F. The obtained function f is bent precisely when M satisfies the conditions of definition B.7. If there is another matrix G consisting of d vectors in $\mathbb{C}^{|G|}$ such that $M = G^\dagger G$, then it must hold that $F = UG$ for some unitary matrix U [HJ12, Theorem 7.3.11]. It follows that the map

$$\psi : C_d(G) \to B_d(G), \quad M = F^\dagger F \mapsto f$$

is well-defined and the inverse of ϕ, giving the desired bijection. □

In terms of Gram matrices, concatenation as in lemma B.1 means the following.

Corollary B.9. *A bent function $f \in B_{n+m}(G)$ is a concatenation of functions in $B_n(G)$ and $B_m(G)$ if and only if the corresponding Gram matrix M can be written as $M = tM_1 + (1-t)M_2$ for some $M_1 \in C_n(G), M_2 \in C_m(G)$ and $t \in [0,1]$. In this case we call M* concatenated.

B.2 The case $G = \mathbb{Z}/3\mathbb{Z}$

For the small cyclic group $\mathbb{Z}/3\mathbb{Z}$ we will show some examples of bent functions in dimension $d = 1, 2$. First, we classify all one-dimensional bent functions and then we show that not all 2-dimensional ones are equivalent to concatenation constructions. Since $\mathrm{U}(\mathbb{C}) = \{u \in \mathbb{C} : |u| = 1\}$

we can rephrase

$$\begin{aligned} B_1(\mathbb{Z}/3\mathbb{Z}) &= \{f : \mathbb{Z}/3\mathbb{Z} \to C : f \text{ bent and } f(0) = 1\} \\ &\cong \{f = (1, z_1, z_2) \in \mathbb{C}^3 : |z_1| = |z_2| = 1, z_1 + \overline{z_2} + \overline{z_1} z_2 = 0)\} && \text{(lemma B.6)} \\ &= \{f = (1, x_1 + iy_1, x_2 + iy_2) : x_1^2 + y_1^2 = x_2^2 + y_2^2 = 1, \\ &\qquad x_1 + x_2 + x_1 x_2 + y_1 y_2 = y_1 - y_2 + x_1 y_2 - x_2 y_1 = 0\} \\ &= \{(1, 1, \omega), (1, 1, \omega^2), (1, \omega, 1), (1, \omega, \omega^2), (1, \omega^2, 1), (1, \omega^2, \omega)\}. && (\omega = e^{2\pi i/3}) \end{aligned}$$

Equivalently,

$$C_1(\mathbb{Z}/3\mathbb{Z}) = \left\{ \begin{pmatrix} 1 & a & b \\ \bar{a} & 1 & \bar{a}b \\ \bar{b} & a\bar{b} & 1 \end{pmatrix} : (1, a, b) \in B_1(\mathbb{Z}/3\mathbb{Z}) \right\}$$

and all concatenated matrices in $C_2(G)$ are of the form $tM_1 + (1 - t)M_2$ with $t \in [0, 1]$ and $M_i \in C_1(\mathbb{Z}/3\mathbb{Z})$.

Example B.10. Consider the matrix

$$M := \begin{pmatrix} 1 & \frac{e^{2\pi i a}}{\sqrt{2}} & -\frac{e^{-2\pi i a}}{\sqrt{2}} \\ \frac{e^{-2\pi i a}}{\sqrt{2}} & 1 & 0 \\ -\frac{e^{2\pi i a}}{\sqrt{2}} & 0 & 1 \end{pmatrix}$$

with $a \in \mathbb{R}$. It can be readily checked that M lies in $C_2(\mathbb{Z}/3\mathbb{Z})$ and it has rank 2 with eigenvalues $0, 1, 2$. The corresponding bent function (up to equivalence) is given by

$$f : \mathbb{Z}/3\mathbb{Z} \to \mathbb{C}^2, \qquad \begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline f_1(x) & 1 & \frac{e^{-2\pi i a}}{\sqrt{2}} & -\frac{e^{2\pi i a}}{\sqrt{2}} \\ f_2(x) & 0 & \frac{e^{-2\pi i a}}{\sqrt{2}} & \frac{e^{2\pi i a}}{\sqrt{2}} \end{array} .$$

Using a computer program like *Mathematica* reveals that M is not equal to any combination of two points in $C_1(\mathbb{Z}/3\mathbb{Z})$, making f not concatenated.

C Forrelation for abelian groups

The forrelation problem for boolean functions was introduced by Aaronson and Ambainis [AA15] as an example of optimal separation between quantum and classical query complexity. We explain the concept of forrelation and extend it to functions on general abelian groups.

For two boolean functions $f, g : \mathbb{F}_2^n \to \{-1, 1\}$, the *forrelation* (or Fourier correlation) is given by

$$\Phi_{f,g} := \frac{1}{2^n} \sum_{x \in \mathbb{F}_2^n} f(x) W_g(x) = \frac{1}{2^{3n/2}} \sum_{x,y \in \mathbb{F}_2^n} f(x)(-1)^{x \cdot y} g(y), \tag{C.1}$$

where W_g denotes the Walsh-Hadamard transform of g, also denoted by $\hat{g}$. The forrelation problem is then to determine whether $|\Phi_{f,g}| \leq \frac{1}{100}$ or $|\Phi_{f,g}| \geq \frac{3}{5}$, assuming one of the two holds. The relation with boolean bent functions is given by the following lemma.

Proposition C.1 ([DM24, Proposition 5.2])**.** *Given bent functions $f, g : \mathbb{F}_2^n \to \{-1, 1\}$ such that $g(x) = f(x \oplus u)$ for all $x \in \mathbb{F}_2^n$, then*

$$\Phi_{g,\hat{f}} = \begin{cases} 1 & \textit{if } u = 0^n, \\ 0 & \textit{if } u \neq 0^n. \end{cases} \tag{C.2}$$

For boolean functions, taking the Fourier transform twice gives the identity. For general abelian groups this is not the case anymore. By the correspondence between a group G and its double dual

$$G \to \hat{\hat{G}}, \quad x \mapsto \{\psi_x : \phi \mapsto \phi(x) \text{ for all } \phi \in \hat{G}\}, \tag{C.3}$$

one can easily check that $\hat{\hat{f}}(\psi_x) = f(-x)$. Instead, we make use of the inverse Fourier transform.

Definition C.2 (Inverse Fourier transform)**.** For a function $h : \hat{G} \to \mathbb{C}$, the *inverse Fourier transform* is given by

$$\check{h} : G \to \mathbb{C}, \quad \check{h}(x) = \frac{1}{|G|^{1/2}} \sum_{\phi \in \hat{G}} \overline{\phi(x)} h(\phi). \tag{C.4}$$

For a multidimensional function the inverse Fourier transform is applied coordinate-wise.

Implied by its name, the inverse Fourier transform is the inverse of the Fourier transform. We use it to define forrelation for multidimensional functions on abelian groups.

Definition C.3 (Multidimensional forrelation)**.** Let $g : G \to \mathbb{C}^d$ and $h : \hat{G} \to \mathbb{C}^d$ be two functions. The *forrelation* between g and h is given by

$$\Phi_{g,h} = \frac{1}{|G|} \sum_{x \in G} \sum_{i=1}^{d} g_i(x) \overline{\check{h}_i(x)}. \tag{C.5}$$

The forrelation between two bent functions that are shifted relative to each other has the following property.

Proposition C.4. *Let $f, g : G \to \mathbb{C}^d$ be bent functions hiding a shift s. Then*

$$\Phi_{g,\hat{f}} = \begin{cases} 1 & \textit{if } s = 0, \\ 0 & \textit{if } s \neq 0. \end{cases} \tag{C.6}$$

Proof. Writing out the forrelation definition we obtain

$$\Phi_{g,\hat{f}} = \frac{1}{|G|} \sum_{x \in G} \sum_{i=1}^{d} g_i(x) \overline{\check{\hat{f}}_i(x)} = \frac{1}{|G|} \sum_{x \in G} \sum_{i=1}^{d} f_i(x-s) \overline{f_i(x)}.$$

By [Poi05, Proposition 5.2] this expression is zero for all $s \neq 0$, from which the result follows. □