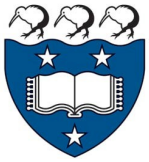
The Sparse Fourier Transform



Joel Laity

Department of Mathematics
The University of Auckland

Supervisor: Dr Steven Galbraith

A thesis submitted in fulfilment of the requirements for the degree of MSc in Mathematics, The University of Auckland, 2016.

Abstract

Some functions can be well approximated by taking their Fourier transforms and discarding the terms that have small Fourier coefficients. The sparse Fourier transform is an algorithm that computes such an approximation more efficiently than computing the entire Fourier transform.

The sparse Fourier transform has many applications to problems in mathematics and engineering. For example, in mathematics the sparse Fourier transform can be used to solve the *chosen multiplier hidden number problem*. In engineering, the sparse Fourier transform can be used to compress audio or video data.

In Chapter 3 we present an algorithm that computes the sparse Fourier transform. This algorithm generalises and unifies the sparse fast Fourier transforms in [19] and [21]. These algorithms are of particular importance as they are the earliest algorithms for computing the sparse Fourier transform.

The final chapter develops a method for reducing the problem of calculating the sparse Fourier transform over $\mathbb{Z}_n$ to calculating it over $\mathbb{Z}_{2^k}$ where k is the smallest integer such that $n \leq 2^k$, provided the function has certain special properties. This method is based on ideas from Shor's algorithm for factoring integers.

Contents

Chapter 1

Introduction

1.1 The sparse Fourier transform

Fourier analysis was originally developed to study the vibration of strings [8, p. 267]. A vibrating string can be viewed as a function on an interval, the red graph in Figure 1.1. This complicated function can be decomposed into simple, periodic functions, the blue graphs in Figure 1.1, called pure tones.

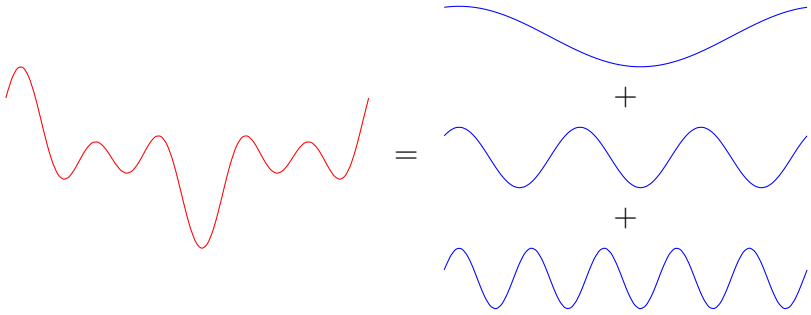


Figure 1.1: A complicated function, in red, as the sum of three simple, periodic functions, in blue.

In practice, the vibration of a string is measured by dividing the interval into evenly spaced segments and measuring the displacement of the string at each segment as in Figure 1.2.

More formally, the red function in Figure 1.2 is a mapping $f : \mathbb{Z}_n \to \mathbb{R}$ where n is the number of samples. And the pure tones are a family of functions $g_k : \mathbb{Z}_n \to \mathbb{R}$ by $g_k(x) = \cos\left(\frac{\pi}{n}(x + 1/2)(k + 1/2)\right)$ where $k = 0, 1, \ldots, n-1$. The

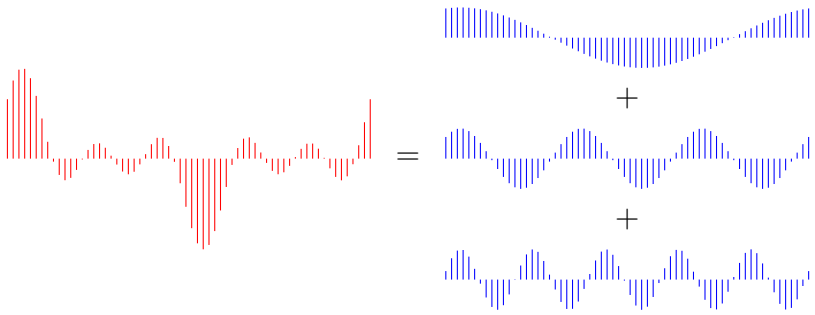


Figure 1.2: A discrete version of figure 1.1.

g_k are periodic and form a basis for the space of all functions $\{f : \mathbb{Z}_n \to \mathbb{R}\}$. Thus any $f : \mathbb{Z}_n \to \mathbb{R}$ can be written as a linear combination, $f = \sum_{k=0}^{n-1} c_k g_k$, where the $c_k \in \mathbb{R}$ are the amplitudes of the pure tones.

In this thesis, we will be interested in complex valued functions. We therefore consider $f : \mathbb{Z}_n \to \mathbb{C}$ (more generally, $f : G \to \mathbb{C}$ where G is a finite abelian group). The complex analogue of the pure tones is the family of functions $\chi_\alpha : \mathbb{Z}_n \to \mathbb{C}$ defined by $\chi_\alpha(x) = \exp(2\pi i x\alpha/n)$ for $\alpha = 0, 1, \ldots, n-1$. The χ_α form a basis for the space of all functions $\{f : \mathbb{Z}_n \to \mathbb{C}\}$. Thus any $f : \mathbb{Z}_n \to \mathbb{C}$ can be written as a linear combination, $f = \sum_{\alpha=0}^{n-1} \widehat{f}(\alpha)\chi_\alpha$ where $\widehat{f}(\alpha) \in \mathbb{C}$. The complex numbers $\widehat{f}(\alpha)$ are called the *Fourier coefficients*.

We now have two ways to describe a function $f : \mathbb{Z}_n \to \mathbb{C}$. We can either list the function values $f(0), f(1), \ldots, f(n-1)$ or we can list the Fourier coefficients $\widehat{f}(0), \widehat{f}(1), \ldots, \widehat{f}(n-1)$. The *discrete Fourier transform* is the mapping $f \mapsto \widehat{f}$.

The *fast Fourier Transform* is an algorithm which computes the discrete Fourier transform of a function $f : \mathbb{Z}_n \to \mathbb{C}$ where n is an integer power of 2. That is, the fast Fourier transform takes the list of complex numbers $f(0), f(1), \ldots, f(n-1)$ as input and outputs $\widehat{f}(0), \widehat{f}(1), \ldots, \widehat{f}(n-1)$. The running time of the fast Fourier transform is $O(n \log n)$. This is "fast" because the naive algorithm takes $O(n^2)$ time. Moreover, since the output for the discrete Fourier transform has size n, the best running time we can hope for is a linear time algorithm, i.e., $O(n)$.

For many applications (see the next section) it is only necessary to find the $\alpha \in \mathbb{Z}_n$ for which $|\widehat{f}(\alpha)|$ is "large". One way of finding these coefficients would be to use the fast Fourier transform to calculate all the Fourier coefficients and then discard those with small magnitude. Of course, this seems wasteful if there

are only a small number of large coefficients. Thus there is interest in developing algorithms which can improve on this naive approach. A (possibly probabilistic) algorithm which can identify the $\alpha \in \mathbb{Z}_n$ for which $|\widehat{f}(\alpha)|$ is large in sublinear time is called a *sparse Fourier transform*.

1.2 Applications

The Fourier transform has applications in analysing signals where it is expected that $\widehat{f}(\alpha)$ is large for only a few $\alpha \in \mathbb{Z}_n$. It turns out that many signals which are relevant to engineering satisfy this property. For example, in a typical 8×8 video only 11% of the Fourier coefficients are non-negligible [14]. There is a research group at MIT which has developed many sparse Fourier transform algorithms and has even built a chip which is optimised to calculate sparse Fourier transforms [1].

Sparse fast Fourier transforms also have applications to a problem called the *hidden number problem*. The hidden number problem was introduced in [6] by Boneh and Venkatesan to study the bit security of the Diffie-Hellman key exchange protocol.

Definition 1 (Hidden number problem)**.** *Let $f : \mathbb{Z}_p \to \mathbb{C}$ where p is prime. Let $s \in \mathbb{Z}_p^*$. Given a set of pairs $\{(x_i, f(sx_i))\}_i$ and access to the function f, the hidden number problem is to find s.*

We now explain the motivation for this definition. In the Diffie-Hellman key exchange protocol two parties, Alice and Bob, with private keys a and b compute g^a and g^b where $g \in \mathbb{Z}_p^*$ is a primitive element and send each other these values. They then compute the shared key g^{ab}. The goal for an attacker is, given g, g^a and g^b, to find $s = g^{ab}$. This is known as the *Diffie-Hellman problem*.

The computational assumption is that finding s hard. This necessarily means that it is difficult to compute some of the bits of s; if all the bits of s were easy to compute we would have the binary representation for s. If we want to show a particular bit is hard to compute, say the most significant bit, we first suppose the opposite, i.e., there exists an algorithm A which given g, g^a and g^b can compute in polynomial time the most significant bit of g^{ab} for any integers a, b. We then show that if Eve has access to the algorithm A then she can calculate s in polynomial time. This contradicts the computational assumption that the Diffie-Hellman problem is hard and proves that the problem of computing the most significant bit of s is a hard problem.

Define $f : \mathbb{Z}_p \to \mathbb{C}$ by $f(h) = (-1)^{\mathrm{MSB}(h)}$ and define $f_s : \mathbb{Z}_p \to \mathbb{C}$ by $f(h) = f(sh)$. Eve can choose various numbers k and calculate $g^a g^k = g^{a+k}$. If we suppose she has access to the oracle A she can calculate $A(g, g^{a+k}, g^b) = \mathrm{MSB}((g^{a+k})^b) = \mathrm{MSB}(sg^{kb})$. This means Eve can calculate pairs $(g^{kb}, f_s(g^{kb}))$. Since the goal is to find s, the problem is now reduced to a specific case of the hidden number problem, in particular the case where $f : \mathbb{Z}_p \to \{-1, 1\}$.

There is a variant of the hidden number problem called the chosen multiplier hidden number problem where, instead of being given the pairs $(x_i, f(sx_i))$ and being asked to find s, one is given oracle access to the function $f_s : \mathbb{Z}_p \to \mathbb{C}$ defined by $f_s(x) = f(sx)$.

Definition 2 (Chosen multiplier hidden number problem)**.** *Let $f : \mathbb{Z}_p \to \mathbb{C}$ where p is prime. Let $s \in \mathbb{Z}_p^*$. The goal is to find s given oracle access to the function $f_s : \mathbb{Z}_p \to \{1, -1\}$ defined by $f_s(x) = f(sx)$.*

Note that a solution to the chosen multiplier hidden number problem does not give a solution to the hidden number problem. That is, the chosen multiplier hidden number problem is weaker than the hidden number problem.

Nevertheless, the chosen multiplier hidden number problem is in itself interesting. In fact, it can be reduced the problem of identifying the large Fourier coefficients by the next theorem.

Theorem 1. *(Informal) Let A be an algorithm that learns the large Fourier coefficients of a function $f : \mathbb{Z}_p \to \mathbb{C}$. Then there exists an algorithm which solves the chosen multiplier hidden number problem for f.*

Proof. (Sketch) Let $f, f_s : \mathbb{Z}_p \to \mathbb{C}$ be the functions from the hidden number problem. The Fourier transform has the property that $\widehat{f_s}(x) = \widehat{f}(s^{-1}x)$ see Proposition 8 . Thus the Fourier coefficients of f_s are simply the Fourier coefficients of f permuted by s^{-1}. We use the algorithm, A to compute the (short) list of large Fourier coefficients of f and f_s and then match them up to deduce the permuting element s^{-1}. □

1.3 Work on the problem

The first sparse fast Fourier transform algorithm was published by Kushilevitz and Mansour in [19] who built on work by Goldreich and Levin in [11]. Kushilevitz and Mansour showed how their algorithm could be used to learn decision trees.

Their algorithm works for functions of the form $f : \mathbb{Z}_2^n \to \{-1, 1\}$. This was soon followed by another paper by Mansour [21] which showed how to identify the heavy Fourier coefficients of functions of the form $f : \mathbb{Z}_{2^n} \to \mathbb{C}$. Since then, work on sparse Fourier transforms has flourished and many algorithms have been proposed, including [9], [5], [10], [4], [18], [14], [13], [20], [7], [22], [17], [16]. See [16] for a table of algorithms and their complexity.

To solve the hidden number problem we need to compute the sparse Fourier transform over $\mathbb{Z}_p$. Most of the algorithms cited above work over $\mathbb{Z}_{2^n}$, where n is some positive integer, as this is the most common case for engineering applications and lends itself to divide and conquer style algorithms. A sparse Fourier transform over $\mathbb{Z}_p$ was first outlined in [5] and further details were given in [2] and [3].

1.4 Organisation of thesis

The main part of this thesis is divided into four chapters. Chapter 2 develops the theory surrounding the Fourier transform in a rigorous and self contained way. The goal is to present (mostly) standard knowledge with consistent notation and in full generality.

In Chapter 3 we present an algorithm that computes the sparse Fourier transform over any finite abelian group; it is efficient when the prime factors of $|G|$ are small. This algorithm generalises and unifies the sparse fast Fourier transforms in [19] and [21]. These algorithms are of particular importance as they are the earliest algorithms for computing the sparse Fourier transform.

Chapter 4 contains a discussion of the algorithm in [5] which works for functions of the form $f : \mathbb{Z}_n \to \mathbb{C}$ where n is any positive integer. We show how the ideas used in this algorithm are related to the algorithms in [19] and [21].

The final chapter develops a method for reducing the problem of calculating the sparse Fourier transform over $\mathbb{Z}_n$ to calculating it over $\mathbb{Z}_{2^k}$, where k is the smallest integer such that $n \leq 2^k$, provided the function has certain special properties. This method is based on ideas from Shor's algorithm for factoring integers.

Chapter 2

Preliminaries for Fourier analysis

A good reference for the mathematics of the discrete Fourier transform is [24, Chapter 7]. It contains a version of many of the theorems we will present here. However, as the book is written with engineering applications in mind, the approach is less group-theoretic and less general.

The author highly recommends the textbook [26] by Terras. This book describes an interesting array of applications of discrete Fourier analysis. Most of the material in this chapter is covered in this textbook, although, as the book is quite expositional and gives many applications, these theorems are spread throughout the book and many of the proofs are left as exercises.

In this chapter we will develop the theory surrounding the Fourier transform in a rigorous and self contained way. The goal is to present (mostly) standard knowledge with consistent notation and in full generality.

2.1 The space $L^2(G)$

The group $\mathbb{Z}_n = (\mathbb{Z}_n, +)$ is the set of integers $\{0, 1, \ldots, n-1\}$ under the group operation of addition modulo n. For $x \in \mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \cdots \times \mathbb{Z}_{m_r}$ we write x as an r-tuple $x = (x_1, x_2, \ldots, x_r)$ where $x_i \in \mathbb{Z}_{m_i}$. We use $\mathbb{C}^* = (\mathbb{C} \setminus \{0\}, \cdot)$ to denote the group of nonzero complex numbers under multiplication.

Definition 3. *Let G be a finite, abelian group. Define $L^2(G)$ to be the set of all functions $f : G \to \mathbb{C}$. In symbols,*

$$L^2(G) = \{f : G \to \mathbb{C}\}.$$

The set $L^2(G)$ is a vector space over $\mathbb{C}$ with the usual addition and scalar multiplication of functions. It comes with an inner product

$$\langle f, g\rangle = \frac{1}{|G|}\sum_{x\in G} f(x)\overline{g(x)}.$$

In fact, $L^2(G)$ is a Hilbert space; i.e., the inner product satisfies the following properties:

- $\langle f, g\rangle = \overline{\langle g, f\rangle}$,
- $\langle f, g\rangle$ is linear in f and conjugate linear in g,
- $\langle f, f\rangle \geq 0$,
- $\langle f, f\rangle = 0$ if and only if $f = 0$,

and the space is complete with respect to the norm induced by the inner product

$$\|f\|_2 = \sqrt{\langle f, f\rangle}.$$

Proposition 1. *Let G be a finite, abelian group. Then*

$$\dim L^2(G) = |G|.$$

Proof. For all $x \in G$ define $\delta_x : G \to \mathbb{C}$ by $\delta_x(y) = 1$ if $x = y$ and $\delta_x(y) = 0$ if $x \neq y$. Then the δ_x are linearly independent. For any $f \in L^2(G)$ we have $f = \sum_{x\in G} f(x)\delta_x$ thus the δ_x span $L^2(G)$. It follows that $\dim L^2(G) = |G|$. □

Definition 4. *Let G be a finite, abelian group. Let $f, g \in L^2(G)$. Then the convolution of f and g is the function $f * g : G \to \mathbb{C}$ defined by*

$$(f * g)(x) = \frac{1}{|G|}\sum_{y\in G} f(y)g(x - y).$$

Proposition 2. *Let G be a finite, abelian group. Let $f, g, h \in L^2(G)$. Then*

1. $f * g = g * f$,
2. $f * (g * h) = (f * g) * h$,
3. $f * (g + h) = f * g + f * h$.

Proposition 2 shows that the vector space $L^2(G)$ is an associative algebra where $*$ is the algebra multiplication. In fact, if we instead define convolution by $(f * g)(x) = \sum_{y \in G} f(y)g(x-y)$ then the space $L^2(G)$ is isomorphic to the group algebra of G where the multiplication in the group algebra is convolution in $L^2(G)$. We will use Definition 4 as this seems to be the standard definition in the cryptography literature.

2.2 Characters

Definition 5. *Let G be a finite, abelian group. Define the dual of G, denoted $\widehat{G}$, to be the set of all group homomorphisms from G to the group of non-zero complex numbers $\mathbb{C}^*$. In symbols,*

$$\widehat{G} = \{\chi : G \to \mathbb{C}^* \mid \chi \text{ is a group homomorphism}\}.$$

The members of $\widehat{G}$ are called characters.

Remark 1. For any $\chi \in \widehat{G}$ and any $x \in G$ we have $\chi(x)^{|G|} = \chi(|G|x) = \chi(0) = 1$. So $\chi(x)$ is a $|G|$-th root of unity and, in particular, $|\chi(x)| = 1$.

Definition 6. *Let $\chi \in \widehat{G}$ be a character of G. Define the conjugate character $\overline{\chi}(x) : G \to \mathbb{C}$ by $\overline{\chi}(x) = \overline{\chi(x)}$.*

Proposition 3. *The set $\widehat{G}$ from Definition 5 forms an abelian group where the group operation is pointwise multiplication, i.e. $(\chi\psi)(x) = \chi(x)\psi(x)$ for any $\chi, \psi \in \widehat{G}$ and any $x \in G$.*

Proof. The group operation is clearly associative. Let $\chi, \psi \in \widehat{G}$. Then $(\chi\psi)(x+y) = \chi(x+y)\psi(x+y) = \chi(x)\chi(y)\psi(x)\psi(y) = (\chi\psi)(x)(\chi\psi)(y)$. So $\widehat{G}$ is closed under multiplication. Finally the inverse of a character χ in the group $\widehat{G}$ is the conjugate character $\overline{\chi}$ as defined in Definition 6. The function $\overline{\chi}$ is clearly a homomorphism and it is the inverse of χ since $(\chi\overline{\chi})(x) = \chi(x)\overline{\chi(x)} = |\chi(x)|^2 = 1$, where the last equality follows by Remark 6. □

Definition 7. *Let $G = \mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \cdots \times \mathbb{Z}_{m_r}$. Then for every $x = (x_1, x_2, \ldots, x_r) \in G$ define $\chi_x : G \to \mathbb{C}^*$ by*

$$\chi_x(y) = \prod_{j=1}^{r} \exp\left(2\pi i \frac{x_j y_j}{m_j}\right),$$

where $i = \sqrt{-1}$.

Remark 2. Note that $\chi_x(y) = \chi_y(x)$ for all $x, y \in G$.

It is easy to verify that χ_x is a character of G for all $x \in G$. The next proposition shows that all the characters are of this form.

Proposition 4. *Let $G = \mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \cdots \times \mathbb{Z}_{m_r}$. Then $G \cong \widehat{G}$ under the isomorphism $\phi : G \to \widehat{G}$ defined by $\phi(x) = \chi_x$.*

Proof. The map ϕ is a homomorphism since

$$\begin{aligned}\chi_{x+y}(z) &= \prod_{j=1}^{r} \exp\left(2\pi i \frac{(x_j + y_j)z_j}{m_j}\right) \\ &= \prod_{j=1}^{r} \exp\left(2\pi i \frac{x_j z_j}{m_j}\right) \exp\left(2\pi i \frac{y_j z_j}{m_j}\right) \\ &= \chi_x(z)\chi_y(z).\end{aligned}$$

For surjectivity, define $e_j = (0, \ldots, 0, 1, 0 \ldots, 0) \in G$ to be the tuple with 1 in the j-th position and 0 everywhere else. Let $\chi : G \to \mathbb{C}^*$ be a homomorphism. Then $\chi(e_j)$ is an m_j-th root of unity (see Remark 1) so $\chi(e_j) = \exp(2\pi i x_j/m_j)$ for some $x_j \in \{0, 1, \ldots, m_j - 1\}$. Let $x = (x_1, \ldots, x_{m_j})$ then $\chi_x(e_j) = \chi(e_j)$ for all $1 \leq j \leq r$ and, since the e_j form a generating set for G, it follows that $\chi = \chi_x$. □

The fundamental theorem of finite abelian groups, when combined with the proposition above, tells us that any finite abelian group is isomorphic to its dual.

Proposition 5. *Any finite abelian group G is isomorphic to its dual $\widehat{G}$.*

Note that there is no *canonical* isomorphism between G and $\widehat{G}$. For example, when G is cyclic we must first identify it with $\mathbb{Z}_n$ to invoke Proposition 4. To do this we must first choose a generator for G.[1] A change in the choice of generator changes the isomorphism. Despite this, we will find it very useful later on (see Definition 6) to index the elements of $\widehat{G}$ with elements of G using the χ_x notation.

Since G is isomorphic to its dual it is obvious that G is isomorphic to its double dual, $G^{\widehat{}\,\widehat{}}$. Even though there is no canonical isomorphism between G and $\widehat{G}$ there *is* a canonical isomorphism between G and $G^{\widehat{}\,\widehat{}}$. The isomorphism is $\phi : G \to G^{\widehat{}\,\widehat{}}$ where, for each $g \in G$ we define $\phi(g) : \widehat{G} \to \mathbb{C}^*$ by $[\phi(g)](\chi) = \chi(g)$.

[1] It is possible to define precisely what it means to say no isomorphism between G and $\widehat{G}$ is canonical using category theory. We will not discuss it here.

Theorem 2. *Let G be a finite, abelian group. The group, $\widehat{G}$, of characters of G satisfies the following orthogonality relations:*

$$\sum_{x\in G}\chi(x)=\begin{cases}|G| & \text{if } \chi \text{ is the identity in } \widehat{G},\\ 0 & \text{otherwise,}\end{cases} \qquad \sum_{\chi\in\hat{G}}\chi(x)=\begin{cases}|G| & \text{if } x=0,\\ 0 & \text{otherwise.}\end{cases}$$

Proof. Let $S=\sum_{x\in G}\chi(x)$. If χ is the identity in $\widehat{G}$ then clearly $\chi(x)=1$ for all $x\in G$ so $S=|G|$. If not, then there exists some $y\in G$ such that $\chi(y)\neq 1$. Then

$$\chi(y)S=\sum_{x\in G}\chi(x+y)=S.$$

Hence $S=0$.

For the second part let $T=\sum_{\chi\in\hat{G}}\chi(x)$. If $x=0$ then $T=|G|$. If $x\neq 0$ then, by using Definition 7 it is easy to see there exists some $\psi\in\widehat{G}$ such that $\psi(x)\neq 1$. Then

$$\psi(x)T=\sum_{\chi\in\hat{G}}(\psi\chi)(x)=T.$$

Hence $T=0$. □

Corollary 1. *Let $\chi,\psi\in\widehat{G}$. Then*

$$\langle\chi,\psi\rangle=\begin{cases}1 & \text{if } \chi=\psi,\\ 0 & \text{if } \chi\neq\psi.\end{cases}$$

Thus the set $\widehat{G}$ forms an orthonormal basis for $L^2(G)$.

Proof. By definition

$$\langle\chi,\psi\rangle=\frac{1}{|G|}\sum_{x\in G}\chi(x)\overline{\psi(x)}=\frac{1}{|G|}\sum_{x\in G}(\chi\overline{\psi})(x).$$

Recall from the proof of Proposition 3 that $\overline{\psi}=\psi^{-1}$. The corollary now follows by substituting $\chi\psi^{-1}$ for χ in the first equation of the theorem. Since the characters are orthogonal they are linearly independent in $L^2(G)$. Using Propositions 4 and 1 we know $|\widehat{G}|=|G|=\dim L^2(G)$ hence the characters form a spanning set. □

2.3 The Fourier transform

Since the characters form an orthonormal basis for $L^2(G)$ we know that any $f \in L^2(G)$ can be written as $f = \sum_{x \in G} c_x \chi_x$, where $c_x = \langle f, \chi_x \rangle$. The Fourier transform of a function is a mapping which takes a character as input and outputs the coefficient, c_x, of that character.

Definition 8. *The Fourier transform is a function $\mathscr{F} : L^2(G) \to L^2(\widehat{G})$ defined by*

$$(\mathscr{F} f)(\chi) = \langle f, \chi \rangle$$

for any $f \in L^2(G)$.

Since the inner product on $L^2(G)$ is linear in the first slot it follows that $\mathscr{F}$ is a linear map. The kernel of $\mathscr{F}$ is trivial because if $\langle f, \chi \rangle = 0$ for all $\chi \in \widehat{G}$ then $f = \sum_{\chi \in \widehat{G}} \langle f, \chi \rangle \chi = 0$. This means that $\mathscr{F}$ is an injective linear map and, since $L^2(G)$ has the same dimension as $L^2(\widehat{G})$, we conclude that $\mathscr{F}$ is in fact a bijection. In other words $\mathscr{F}$ is a vector space isomorphism. We formalise this in the next proposition.

Proposition 6. *The map $\mathscr{F} : L^2(G) \to L^2(\widehat{G})$ is bijective and linear.*

Unfortunately the notation $(\mathscr{F} f)(\chi)$ is a little cumbersome. We will define a different notation which uses the fact that we can index the characters of $\mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \cdots \times \mathbb{Z}_{m_r}$ by the elements of $\mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \cdots \times \mathbb{Z}_{m_r}$.

Definition 9. *Let $G = \mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \cdots \times \mathbb{Z}_{m_r}$. For every $f : G \to \mathbb{C}$ define $\widehat{f} : G \to \mathbb{C}$ by $\widehat{f}(x) = (\mathscr{F} f)(\chi_x)$.*

Note that $\widehat{f}(x)$ is not well defined if the domain of f is an arbitrary abelian group. This is because there is no canonical isomorphism between G and $\widehat{G}$ and therefore no God-given choice for what the character χ_x should be for a given $x \in G$.

The next proposition gives us a formula for writing f as a linear combination of the characters of $\mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \cdots \times \mathbb{Z}_{m_r}$.

Proposition 7. *Let $G = \mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \cdots \times \mathbb{Z}_{m_r}$. Then*

$$f = \sum_{x \in G} \widehat{f}(x) \chi_x$$

for all $f \in L^2(G)$.

Proof. Note by definition $\widehat{f}(x) = \langle f, \chi_x \rangle$. The proposition now follows since the χ_x form an orthonormal basis. □

Remark 3. In the special case where the group G is the n-roots of unity, i.e., $G = \{\omega^j \mid j \in \{0, \ldots, n-1\}$ where $\omega = \exp(2\pi i/n)$ and $i = \sqrt{-1}$. It is easy to see that all the homomorphisms from G to $\mathbb{C}^*$ are of the from $x \mapsto x^j$ where $j \in \{0, 1, \ldots, n-1\}$. This means that the space $L^2(G)$ is equal to $\text{span}_{\mathbb{C}}\{1, x, x^2, \ldots, x^{n-1}\}$. In other words, $L^2(G)$ is the set of all polynomials with complex coefficients of degree less than n. The Fourier coefficients are the coefficients of the polynomial.

We have seen in Proposition 6 that the Fourier transform respects the addition and scalar multiplication in $L^2(G)$. Propositions 8 through 11 prove some of the fundamental properties of the Fourier transform. The first of these propositions shows us how the Fourier transform interacts with the convolution operator.

Proposition 8. *Let $G = \mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \cdots \times \mathbb{Z}_{m_r}$. Then*

$$\widehat{(f * g)}(x) = \widehat{f}(x)\widehat{g}(x)$$

for every $f, g \in L^2(G)$ and every $x \in G$.

Proof. Let $f, g \in L^2(G)$. Let $x \in G$. Then

$$\begin{aligned}
\widehat{(f * g)}(x) &= \langle f * g, \chi_x \rangle \\
&= \frac{1}{|G|} \sum_{y \in G} (f * g)(y) \overline{\chi_x(y)} \\
&= \frac{1}{|G|} \sum_{y \in G} \left(\frac{1}{|G|} \sum_{z \in G} f(z) g(y - z) \right) \overline{\chi_x(y)} \\
&= \frac{1}{|G|^2} \sum_{y, z \in G} f(z) g(y - z) \overline{\chi_x(y)}
\end{aligned}$$

using the change of variables $w = y - z$ we get

$$\begin{aligned}
&= \frac{1}{|G|^2} \sum_{w,z \in G} f(z) g(w) \overline{\chi_x(w+z)} \\
&= \frac{1}{|G|^2} \sum_{w,z \in G} f(z) g(w) \overline{\chi_x(w)\chi_x(z)} \\
&= \frac{1}{|G|} \sum_{z \in G} f(z) \overline{\chi_x(z)} \frac{1}{|G|} \sum_{w \in G} g(w) \overline{\chi_x(w)} \\
&= \langle f, \chi_x \rangle \langle g, \chi_x \rangle \\
&= \widehat{f}(x) \widehat{g}(x).
\end{aligned}$$

□

Proposition 9 (Parseval's Identity)**.** *Let $G = \mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \cdots \times \mathbb{Z}_{m_r}$. Let $f \in L^2(G)$. Then*

$$\frac{1}{|G|} \sum_{x \in G} |f(x)|^2 = \sum_{x \in G} |\widehat{f}(x)|^2.$$

Proof. We have

$$\begin{aligned}
\frac{1}{|G|} \sum_{x \in G} |f(x)|^2 &= \langle f, f \rangle \\
&= \langle \sum_{x \in G} \widehat{f}(x) \chi_x, f \rangle \\
&= \sum_{x \in G} \widehat{f}(x) \langle \chi_x, f \rangle \\
&= \sum_{x \in G} \widehat{f}(x) \overline{\langle f, \chi_x \rangle} \\
&= \sum_{x \in G} \widehat{f}(x) \overline{\widehat{f}(x)} \\
&= \sum_{x \in G} |\widehat{f}(x)|^2.
\end{aligned}$$

□

There are two other useful forms of Parseval's theorem which we state in the corollaries below for easy reference later.

Corollary 2. *Let G and f be as above. Then $\frac{1}{|G|}\langle f, f\rangle = \left\langle \widehat{f}, \widehat{f} \right\rangle$.*

Corollary 3. *Let G and f be as above. Then $\frac{1}{|G|}\|f\|_2^2 = \|\widehat{f}\|_2^2$.*

Proposition 10. *Let $s \in G$. Let $f \in L^2(G)$. Define $g : G \to \mathbb{C}$ by $g(x) = f(s+x)$. Then $\widehat{g}(x) = \chi_x(s)\widehat{f}(x)$.*

Proof. We have

$$\begin{aligned}
\widehat{g}(x) &= \langle g, \chi_x \rangle \\
&= \frac{1}{|G|}\sum_{y\in G} g(y)\overline{\chi_x(y)} \\
&= \frac{1}{|G|}\sum_{y\in G} f(s+y)\overline{\chi_x(y)} \\
&= \frac{1}{|G|}\sum_{w\in G} f(w)\overline{\chi_x(w-s)} && \text{substituting } w = y+s \\
&= \frac{1}{|G|}\sum_{w\in G} f(w)\overline{\chi_x(w)\chi_x(-s)} \\
&= \frac{1}{|G|}\sum_{w\in G} f(w)\overline{\chi_x(w)\chi_{-s}(x)} \\
&= \frac{1}{|G|}\sum_{w\in G} f(w)\overline{\chi_x(w)}\chi_x(s) \\
&= \chi_x(s)\frac{1}{|G|}\sum_{w\in G} f(w)\overline{\chi_x(w)}.
\end{aligned}$$

□

This next proposition is used in the proof of Theorem 1 which shows the equivalence between finding heavy Fourier coefficients and solving the chosen multiplier hidden number problem.

Proposition 11. *Let $s \in \mathbb{Z}_n^*$. Let $f \in L^2(\mathbb{Z}_n)$. Define $g : \mathbb{Z}_n \to \mathbb{C}$ by $g(x) = f(sx)$. Then $\widehat{g}(x) = \widehat{f}(s^{-1}x)$.*

Proof. We have

$$\begin{aligned}\widehat{g}(x) &= \langle g, \chi_x \rangle \\ &= \frac{1}{n} \sum_{y \in \mathbb{Z}_n} g(y) \overline{\chi_x(y)} \\ &= \frac{1}{n} \sum_{y \in \mathbb{Z}_n} f(sy) \overline{\chi_x(y)}\end{aligned}$$

We use the change of variables $z = sy$. Since $s \in \mathbb{Z}_n^*$ we know that z ranges over $\mathbb{Z}_n$ as y ranges over $\mathbb{Z}_n$.

$$\begin{aligned}&= \frac{1}{n} \sum_{y \in \mathbb{Z}_n} f(sy) \overline{\chi_x(y)} \\ &= \frac{1}{n} \sum_{z \in \mathbb{Z}_n} f(z) \overline{\chi_x(s^{-1}z)} \\ &= \frac{1}{n} \sum_{z \in \mathbb{Z}_n} f(y) \overline{\chi_{s^{-1}x}(z)} \\ &= \langle f, \chi_{s^{-1}x} \rangle \\ &= \widehat{f}(s^{-1}x).\end{aligned}$$

□

2.4 Quotient groups and the Poisson summation formula

Sparse Fourier transform algorithms were given by Kushilevitz and Mansour for functions of the form $f : \mathbb{Z}_2^n \to \mathbb{C}$ in [19] and for $f : \mathbb{Z}_{2^n} \to \mathbb{C}$ by Mansour in [21]. The generalisation and unification of these two algorithms will be the subject of the next chapter, however we will describe their general approach here as Proposition 15 was inspired by these two papers.

Both the algorithms in [19] and [21] use a divide and conquer approach. They begin by considering a function $f = \sum_{\alpha \in G} \widehat{f}(\alpha)\chi_\alpha$ and then divide G into two disjoint subsets $G = A \cup B$. They then consider the functions $f_A, f_B : G \to \mathbb{C}$ defined by $f_A = \sum_{\alpha \in A} \widehat{f}(\alpha)\chi_\alpha$ and $f_B = \sum_{\alpha \in B} \widehat{f}(\alpha)\chi_\alpha$. Both algorithms

recursively find the large Fourier coefficients of f_A and f_B and then combine the results to get the large Fourier coefficients of f. In both papers the sets A and B are cosets of some subgroup.

The central idea in both algorithms is to have some way of calculating $f_A(x)$ when given access to f. In [19] a formula for $f_A(x)$ is given in Lemma 3.2 and in [21] a formula for $f_A(x)$ is given in Lemma 3.3. The origin of the formulas for $f_A(x)$ are unclear; in fact, the proof of Lemma 3.2 in [21] begins by saying "The proof is somewhat technical. We basically transform the right-hand side to the left-hand side."

Proposition 15 gives a generalisation of the discrete Poisson summation formula which, as far as the author is aware, is novel. The lemmas in [19], [21] are special cases of the proposition. The author hopes that this will demystify the formulas in [19] and [21] and provide some context.

Before we prove Proposition 15 we turn our attention to classifying the duals of quotient groups. This will give us the results we need to prove the finite analogue of the Poisson summation formula. It turns out we can identify the duals of all the quotients of G with subgroups of $\widehat{G}$.

Definition 10. *Let G be a finite, abelian group and let $H \leq G$ be a subgroup of G. Define*

$$H^{\#} = \{\chi \in \widehat{G} \mid \chi(h) = 1 \text{ for all } h \in H\}.$$

In the next proposition we use the notation $\overline{\chi}$ to refer to a character defined on a quotient of G. Everywhere else in this thesis the notation $\overline{\chi}$ refers to the conjugate character defined by $\overline{\chi}(x) = \overline{\chi(x)}$ as in Definition 6.

Proposition 12. *Let G be a finite, abelian group and let $H \leq G$ be a subgroup of G. For each $\chi \in H^{\#}$ define $\overline{\chi} : G/H \to \mathbb{C}^*$ by $\overline{\chi}(g + H) = \chi(g)$. Then $\overline{\chi}$ is well defined and*

$$H^{\#} \cong \widehat{G/H}$$

under the map $\chi \mapsto \overline{\chi}$.

Proof. Define $\phi : H^{\#} \to \widehat{G/H}$ by $\phi(\chi) = \overline{\chi}$. Let $\chi \in H^{\#}$. Since $\chi(h) = 1$ for all $h \in H$ the function $\overline{\chi}$ is well defined. The map ϕ is a homomorphism since $\overline{\chi\psi}(g + H) = \chi\psi(g) = \chi(g)\psi(g) = \overline{\chi}(g + H)\overline{\psi}(g + H)$. Define $\theta : \widehat{G/H} \to H^{\#}$ by $\theta(\psi)(g) = \psi(g + H)$. Then $\theta(\psi)$ is identically 1 on H, so the map is well defined. It is easy to show that θ is the inverse of ϕ, thus ϕ is bijective. This completes the proof. □

Definition 11. *Let $G = \mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \cdots \times \mathbb{Z}_{m_r}$. Let $H \leq G$. Define*

$$H^\perp = \{x \in G \mid \chi_x \in H^\#\}.$$

Let $G = \mathbb{Z}_n$. Then any $H \leq G$ is generated by a single element $H = \langle a \rangle$. If we assume a is minimal then a divides n and $H^\perp = \langle n/a \rangle$. Moreover, if K is a subgroup of some group J then $(H \times K)^\perp = H^\perp \times K^\perp \leq G \times J$, so given generators for any subgroup of $\mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \cdots \times \mathbb{Z}_{m_r}$ it is easy to find $H^\perp$.

Now that we have defined $H^\perp$ we can prove a generalisation of the orthogonality relations in Proposition 2.

Proposition 13. *Let $G = \mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \cdots \times \mathbb{Z}_{m_r}$. Let $H \leq G$. Then*

$$\sum_{h \in H} \chi_h(x) = \begin{cases} |H|, & \text{if } x \in H^\perp, \\ 0, & \text{otherwise.} \end{cases}$$

Proof. If $x \in H^\perp$ then, by definition of $H^\perp$, we have $\chi_x(h) = 1$ for all $h \in H$. Hence by Remark 2 we have $\chi_h(x) = 1$ for all $h \in H$. If $x \notin H^\perp$ then, by definition of $H^\perp$, there exists some $h' \in H$ for which $\chi_x(h') \neq 1$. Then

$$\sum_{h \in H} \chi_h(x) = \sum_{h \in H} \chi_{h+h'}(x) = \chi_{h'}(x) \sum_{h \in H} \chi_h(x) = \chi_x(h') \sum_{h \in H} \chi_h(x)$$

which implies that $\sum_{h \in H} \chi_h(x) = 0$.

□

Proposition 14 (Poisson summation formula)**.** *Let $G = \mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \cdots \times \mathbb{Z}_{m_r}$. Let $f : G \to \mathbb{C}$. Let $H \leq G$. Then*

$$\sum_{h \in H} \widehat{f}(h) \chi_h(x) = \frac{1}{|G : H|} \sum_{y \in H^\perp} f(x - y).$$

We will actually prove a slightly more general version of the Poisson summation formula where the sum on the left hand side of the equation is a sum over a coset instead of a subgroup.

Proposition 15. *Let $f : G \to \mathbb{C}$. Let $H \leq G$. Let $\alpha \in G$. Then*

$$\sum_{h \in H} \widehat{f}(\alpha + h) \chi_{\alpha+h}(x) = \frac{1}{|G : H|} \sum_{y \in H^\perp} f(x - y) \chi_\alpha(y).$$

Proof. Define $g : G \to \mathbb{C}$ by $g(x) = \sum_{h\in H} \chi_{\alpha+h}(x)$. Then Proposition 8 tells us that $\widehat{(f * g)}(\alpha) = \widehat{f}(\alpha)\widehat{g}(\alpha)$ so

$$\sum_{h\in H} \widehat{f}(\alpha + h)\chi_{\alpha+h}(x) = (f * g)(x).$$

Now since the mapping $x \mapsto \chi_x$ is an isomorphism we know $\chi_{\alpha+h}(x) = \chi_\alpha(x)\chi_h(x)$. Hence $g(x) = \chi_\alpha(x) \sum_{h\in H} \chi_h(x)$. It follows from Proposition 13 that

$$g(x) = \begin{cases} \chi_\alpha(x) \cdot |H|, & \text{if } x \in H^\perp, \\ 0 & \text{otherwise.} \end{cases}$$

Putting this all together we get

$$(f * g)(x) = (g * f)(x) = \frac{1}{|G|} \sum_{y\in G} g(y)f(x - y) = \frac{|H|}{|G|} \sum_{y\in H^\perp} \chi_\alpha(y)f(x - y)$$

which is the right hand side of the equation. □

As described in the introduction to this section, Proposition 15 is a general version of Lemma 3.2 in [19] and Lemma 3.3 in [21].

The Poisson summation formula is now an immediate consequence of Proposition 15 by substituting $\alpha = 0$.

Chapter 3

A sparse Fourier transform algorithm

3.1 Introduction

The first sparse Fourier transform algorithm was published by Kushilevitz and Mansour in [19] who built on work by Goldreich and Levin in [11]. Their algorithm is probabilistic and works for *boolean functions on the hypercube*, i.e., functions of the form $f : \mathbb{Z}_2^n \to \{-1, 1\}$. They were interested in functions of this form was because of their application to learning decision trees (see their paper [19] for details).

Two years later, another paper was published by Mansour [21] which gave a probabilistic algorithm for finding the heavy Fourier coefficients of functions of the form $f : \mathbb{Z}_{2^n} \to \mathbb{C}$. Although the paper was written in the language of polynomials (the paper's title is "Randomized interpolation and approximation of sparse polynomials") this is really the same as computing the Fourier coefficients of a function, see Remark 3.

The goal of this chapter is to generalise the ideas from these two papers to construct an algorithm which finds the heavy Fourier coefficients of $f : G \to \mathbb{C}$ where G is a finite abelian group. The algorithms in [19] and [21] were developed for specific applications: learning decision trees and interpolating polynomials respectively. Our approach is more group-theoretic and many of the proofs are substantially different. We hope this approach will lay bare the ideas behind these algorithms.

3.2 Results

The following theorem is the main result of this chapter.

Theorem 3. *There exists an algorithm A which has the following input:*

1. *An abelian group G,*
2. *Oracle access to a function $f : G \to \mathbb{C}$,*
3. *M, an upper bound for $\max_{x \in G} |f(x)|^2$,*
4. *A threshold value θ,*
5. *A failure probability δ.*

The output of A is a list $\alpha_1, \alpha_2, \ldots, \alpha_r \in G$ such that with probability at least $1 - \delta$ the list contains all $\alpha \in G$ such that $|\widehat{f}(\alpha)|^2 \geq \theta$ and does not contain any $\alpha \in G$ for which $|\widehat{f}(\alpha)|^2 < \theta/2$.

The algorithm, A, runs in time $\mathrm{poly}(p, \log|G|, M/\theta, \log(1/\delta))$, where p is the largest prime dividing $|G|$.

The running time depends on the ratio M/θ, in order for the algorithm to be efficient this ratio can not be too large. The time complexity of the algorithms in[19] and [21] also depends on this ratio.

Before we present the algorithm we will develop some terminology and notation.

Definition 12. *Let $f : G \to \mathbb{C}$. We say that the Fourier coefficient $\widehat{f}(\alpha)$ is θ-heavy if $|\widehat{f}(\alpha)|^2 > \theta$.*

Definition 13 (Expected Value Notation)**.** *Let $A = \{a_i \mid i \in I\} \subseteq \mathbb{C}$ be a finite set of complex numbers indexed by I. We define*

$$\mathop{\mathbb{E}}_{i \in I}[a_i] = \frac{1}{|I|} \sum_{i \in I} a_i.$$

Note that $\|f\|_2^2 = \frac{1}{|G|} \sum_{x \in G} |f(x)|^2 = \mathbb{E}_{x \in G} |f(x)|^2$. We will use expected value notation to signal to the reader that the quantity can be approximated by taking random samples and computing the mean. For example we could approximate $\|f\|_2^2$ by choosing m samples $x_i \in G$ at random and calculating $\frac{1}{m} \sum_{i=1}^{m} |f(x_i)|^2$. The Chernoff-Hoeffding bounds tell us how large the sample size m needs to be in order for the estimation $\frac{1}{m} \sum_{i=1}^{m} |f(x_i)|^2$ to be close to the true value $\|f\|_2^2$.

Theorem 4 (Chernoff-Hoeffding)**.** *Let $A = \{a_i \mid i \in I\}$ be a finite set of complex numbers such that $|x| \leq M$ for all $x \in A$. Let $\varepsilon > 0$. Let $m \geq \frac{2M^2}{\varepsilon^2} \log(2/\delta)$. Let $x_j \in A$ be chosen randomly and uniformly from A where $1 \leq j \leq m$. Then*

$$\Pr\left[\left|\mathop{\mathbb{E}}_{i \in I}[a_i] - \frac{1}{m}\sum_{j=1}^{m} x_j\right| > \varepsilon\right] \leq \delta$$

Proof. See [15, Theorem 2]. □

Definition 14. *Let X be a finite set. Let $f : X \to \mathbb{C}$. The support of f is the subset of X, denoted by* $\operatorname{supp} f$*, given by:*

$$\operatorname{supp} f = \{x \in X \mid f(x) \neq 0\}.$$

3.3 The Algorithm

3.3.1 Overview

At a high level the algorithm is relatively simple. It is a recursive algorithm that proceeds as follows.

- Partition G into disjoint subsets, say $G = A \cup B$. Define $f_A, f_B : G \to \mathbb{C}$ by $f_A(x) = \sum_{\alpha \in A} \widehat{f}(\alpha)\chi_\alpha(x)$ and $f_B(x) = \sum_{\alpha \in B} \widehat{f}(\alpha)\chi_\alpha(x)$.
- Estimate the values $\|f_A\|_2^2$ and $\|f_B\|_2^2$.
- 1. If $\|f_A\|_2^2 < \theta$ then no Fourier coefficient $\widehat{f}(\beta)$ can be θ-heavy for any $\beta \in A$ since by Parseval's identity $|\widehat{f}(\beta)|^2 \leq \sum_{\alpha \in A} |\widehat{f}(\alpha)|^2 = \|f_A\|_2^2 < \theta$. Thus we can exclude the set A from our search.
  2. If $\|f_A\|_2^2 \geq \theta$ then recursively partition A and continue the search.
  3. Do steps 1 and 2 for $\|f_B\|_2^2$.
- This continues until the sets are singletons.

It is entirely non-obvious how to do the second step of the algorithm and we will spend most of this section developing a technique for estimating $\|f_A\|_2^2$ and $\|f_B\|_2^2$. Instead of choosing arbitrary sets, we will ensure A and B are both cosets of some subgroup of G. This will allow us to take advantage of the group structure of G when estimating $\|f_A\|_2^2$ and $\|f_B\|_2^2$.

3.3.2 Pseudocode

We will now formally describe the algorithm in pseudocode. The algorithm make calls to the subroutine EstNormSq which estimates $\|f_A\|_2^2$ and $\|f_B\|_2^2$. How the subroutine works will be described later.

Let G be a finite abelian group with a chain of subgroups $0 = H_1 \leq H_2 \leq \cdots \leq H_n = G$ where $|H_{i+1} : H_i|$ is prime for all i. Algorithm 1 defines a recursive function, Heavy, that can be used to compute that θ-heavy coefficients of f. The θ-heavy coefficients of f can be determined by calling Heavy(G).

Algorithm 1: Heavy

Input: A coset $z + H_i$ where $z \in G$ and $1 \leq i \leq n$.
if $i = 1$ **then**
  return z
else
  Let Y be a set of coset representatives for H_{i-1} in H_i
  Let $Y' = \{y \in Y \mid \text{EstNormSq}(f_{(z+y)+H_{i-1}}) \geq 3\theta/4\}$
  return $\cup_{y \in Y'} \text{Heavy}((z + y) + H_{i-1})$

Figure 3.1 on the next page gives an example of how the algorithm would work when the group is $G = \mathbb{Z}_8$ and the subgroups are $0 \leq 4\mathbb{Z}_8 \leq 2\mathbb{Z}_8 \leq \mathbb{Z}_8 = G$.

The graphs in the figure are of the Fourier coefficients of the functions f_A for various $A \subseteq \mathbb{Z}_8$. The algorithm begins by splitting $\mathbb{Z}_8$ into two disjoint sets, $0 + 2\mathbb{Z}_8$ and $1 + 2\mathbb{Z}_8$. It then estimates $\|f_{0+2\mathbb{Z}_8}\|_2^2 = \sum_{\alpha \in 0+2\mathbb{Z}_8} |\widehat{f}(\alpha)|^2$, determines that this is less than the threshold θ and concludes there are no θ-heavy Fourier coefficients in this set. That branch of the tree is therefore not explored any further and is greyed out in the diagram. The algorithm then determines that $\|f_{1+2\mathbb{Z}_8}\|_2^2$ is above the threshold, thus it recursively divides this set in two and explores both subsets. This continues until the sets under consideration are singletons.

3.3.3 The subroutine EstNormSq and filtering

Algorithm 1 relies on calls to the subroutine EstNormSq which estimates $\|f_A\|_2^2$ efficiently. In this section we describe how this subroutine works.

The idea is to use Parseval's identity, which says that $\|f_A\|_2^2 = \mathbb{E}_{x \in G} |f_A(x)|^2$. If we could evaluate $f_A(x)$ for any $x \in G$, then we could estimate $\|f_A\|_2^2$ by

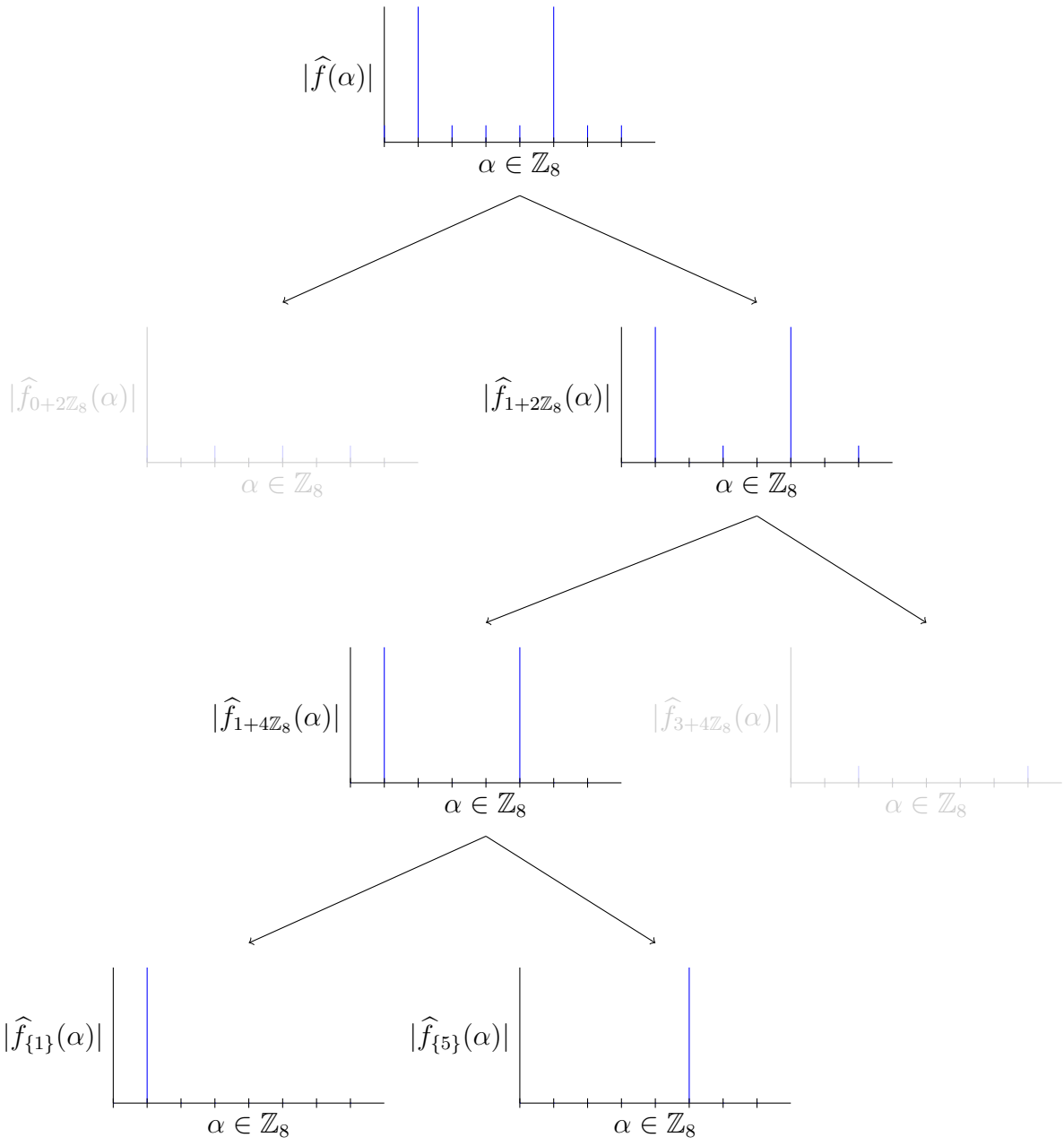


Figure 3.1: An example of the recursion tree of Algorithm 1 for a function $f : \mathbb{Z}_8 \to \mathbb{C}$.

choosing m sufficiently large (as determined by the Chernoff-Hoeffding bounds), randomly choosing $x_i \in G$ where $1 \leq i \leq m$ and calculating

$$\frac{1}{m}\sum_{i=1}^{m} |f_A(x_i)|^2.$$

This reduces the problem of estimating $\|f_A\|_2^2$ to the problem of calculating $f_A(x)$. Unfortunately, for a given $x \in G$, calculating $f_A(x)$ directly from the definition of f_A would require us to know $\widehat{f}(\alpha)$ for all $\alpha \in A$ (and would take $|A|$ additions).

We will therefore estimate $f_A(x)$, instead of computing it exactly. To accomplish this we use a technique called *filtering*. We define a function $h_A : G \to \mathbb{C}$ by

$$h_A(x) = \sum_{\alpha \in A} \chi_\alpha(x). \tag{3.1}$$

We choose this function because it has the property that $\widehat{h}_A(\alpha) = 1$ if $\alpha \in A$ and $\widehat{h}_A(\alpha) = 0$ otherwise. We can now use Proposition 8 which says that $\widehat{f * h_A}(\alpha) = \widehat{f}(\alpha)\widehat{h}_A(\alpha)$ so

$$\widehat{f * h_A}(\alpha) = \begin{cases} \widehat{f}(\alpha), & \text{if } \alpha \in A, \\ 0, & \text{otherwise.} \end{cases}$$

This implies that $f * h_A = f_A$. The function h_A is called a *filter function* because it is defined in such a way that when convoluted with f, it filters out all the coefficients except those in A. Figure 3.2 gives a graphical demonstration of this.

We can estimate $f_A(x)$ by using the fact that

$$f_A(x) = (f * h_A)(x) = \frac{1}{|G|}\sum_{y \in G} f(y)h_A(x-y) = \mathop{\mathbb{E}}_{y \in G} f(y)h_A(x-y). \tag{3.2}$$

We would like to estimate $f_A(x)$ by choosing sufficiently many random samples $y \in G$ and estimating the sum on the right hand side of equation (3.2). This does not require us to know any Fourier coefficients and only requires access to the function f.

In order to be able to efficiently estimate $f_A(x)$ using this method, there are two hurdles that need to be overcome:

1. We need to be able to calculate $h_A(x-y)$ for various $y \in G$.

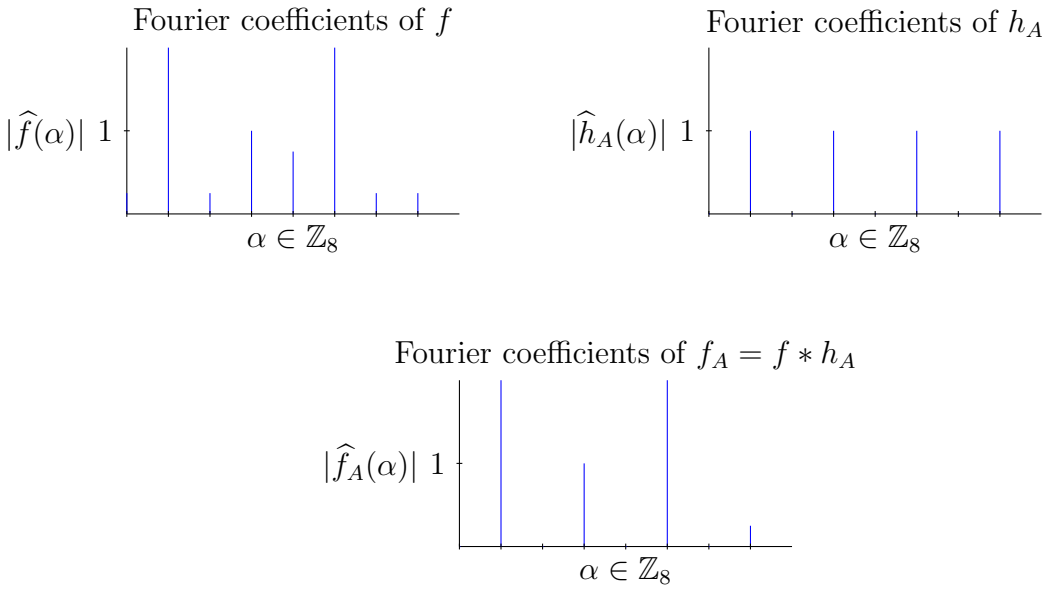


Figure 3.2: The effect of the filter function h_A when convoluted with $f : \mathbb{Z}_8 \to \mathbb{C}$. Here $A = \{1, 3, 5, 7\}$.

2. We require that the number of samples to get an accurate estimate is not too large.

This is where the structure of the set A is crucial; if A is a coset then $A = z + H$ for some $z \in G$ and $H \leq G$. Then

$$h_{z+H}(x) = \sum_{h \in H} \chi_{z+h}(x) = \sum_{h \in H} \chi_z(x)\chi_h(x) = \chi_z(x) \sum_{h \in H} \chi_h(x).$$

Recall the definition of $H^\perp$ in Definition 11. Proposition 13 now tells us that

$$h_{z+H}(x) = \begin{cases} \chi_z(x) \cdot |H|, & \text{if } x \in H^\perp, \\ 0, & \text{otherwise.} \end{cases} \tag{3.3}$$

This solves the first problem: we can easily calculate $h_A(x)$ by simply checking if $x \in H^\perp$ and applying the formula.

For the second problem the Chernoff-Hoeffding bounds tell us that the number of samples m must be proportional to $M = \max_{y \in G} |f(y)h_A(x - y)|$. Unfortunately $h_A(x - y)$ could possibly be as large as $|A|$ (when, for example $x - y = 0$).

This is where we use the fact that $h_A(x)$ has small support:

$$\begin{aligned} f_A(x) &= \frac{1}{|G|}\sum_{y\in G} f(y)h_A(x-y) \\ &= \frac{1}{|G|}\sum_{y\in H^\perp} f(y)\chi_z(x-y)\cdot|H| \\ &= \mathop{\mathbb{E}}_{y\in H^\perp}[f(y)\chi_z(x-y)]. \end{aligned}$$

We can now estimate $\mathbb{E}_{y\in H^\perp}[f(y)\chi_z(x-y)]$ using the Chernoff-Hoeffding bounds. The number of samples only needs to be proportional to $\max_{y\in H^\perp}|f(y)\chi_z(x-y)| = \max_{y\in H^\perp}|f(y)|$.

Remark 4. In general a filter function should:

- Be easy to calculate.
- Have the property that $|\widehat{h}(x)| \approx 1$ if $x \in A$ and $|\widehat{h}(x)| \approx 0$ if $x \notin A$. (In this particular case $|\widehat{h}(x)| = 0$ for $x \notin A$ but this is not necessary, as we will see in the next chapter.)
- Have small support.

For more information on filter functions see [25, Section 4.4].

To summarise, we can write $\|f_A\|_2^2$ using an expected value formula.

$$\begin{aligned} \|f_A\|_2^2 &= \|(f*h_A)\|_2^2 \\ &= \mathop{\mathbb{E}}_{x\in G}|(f*h_A)(x)|^2 \\ &= \mathop{\mathbb{E}}_{x\in G}|\mathop{\mathbb{E}}_{y\in G}[f(y)h_A(x-y)]|^2 \\ &= \mathop{\mathbb{E}}_{x\in G}\left|\mathop{\mathbb{E}}_{y\in H^\perp}[f(y)\chi_z(x-y)]\right|^2 \qquad \text{(using Equation (3.3))} \end{aligned}$$

This allows us to approximate $\|f_A\|_2^2$ by choosing m_1, m_2 sufficiently large, randomly choosing $x_i \in G$ where $1 \le i \le m_1$, randomly choosing $y_{ij} \in H^\perp$ for each i where $1 \le j \le m_2$ and calculating

$$\text{EstNormSq}(f_A) = \frac{1}{m_1}\sum_{i=1}^{m_1}\left|\frac{1}{m_2}\sum_{j=1}^{m_2} f(y_{ij})\chi_z(x_i-y_{ij})\right|^2. \tag{3.4}$$

In the next section, where we prove the running time of the algorithm, we will determine how large m_1 and m_2 need to be.

Algorithm 2: EstNormSq

Input: $f_{z+H} : G \to \mathbb{C}$.
Choose $x_i \in G$ at random for $1 \leq i \leq m_1$
for *each* x_i **do**
  Choose $y_{ij} \in H^\perp$ at random $1 \leq j \leq m_2$
  Calculate $\frac{1}{m_2} \sum_{j=1}^{m_2} f(y_{ij}) \chi_z(x_i - y_{ij})$
return $\frac{1}{m_1} \sum_{i=1}^{m_1} \left| \frac{1}{m_2} \sum_{j=1}^{m_2} f(y_{ij}) \chi_z(x_i - y_{ij}) \right|^2$

3.4 Analysis of algorithm

We will analyse the running time of Algorithm 1 by first bounding the running time of the subroutine EstNormSq in Subsection 3.4.1 and then bounding the number of calls Algorithm 1 makes to the subroutine.

3.4.1 The running time of EstNormSq

The running time of the subroutine EstNormSq is $m_1 m_2$ (we assume all addition and multiplications take constant time to perform). We therefore begin our analysis of Algorithm 1 by determining how large the integers m_1 and m_2 in Equation (3.4) need to be. We will use the Chernoff-Hoeffding bounds, although their application is not straightforward. We define two random variables. The first is

$$A_{m_2}(x_i, z + H) = \frac{1}{m_2} \sum_{j=1}^{m_2} f(y_{ij}) \chi_z(x_i - y_{ij}),$$

and the second is

$$B_{m_1,m_2}(z + H) = \frac{1}{m_1} \sum_{i=1}^{m_1} |A_{m_1}(x_i, z + H)|^2.$$

Note that $B_{m_1,m_2}(z + H)$ is the right hand side of equation 3.4.

We begin with a technical lemma, the purpose of which will become clear in the proof of Theorem 5.

Lemma 1. *Let $L > 0$. Let $A = \{a_i \mid 1 \leq i \leq m\} \subseteq \mathbb{C}$ and $B = \{b_i \mid 1 \leq i \leq m\} \subseteq \mathbb{C}$ be sets of complex numbers such that $\max_{1 \leq i \leq m} |a_i| \leq L$ and $\max_{1 \leq i \leq m} |b_i| \leq L$. Suppose*

$$|a_i - b_i| \leq \varepsilon / 2L$$

for all $i \in \{1, \ldots, m\}$. Then

$$\left| \frac{1}{m} \sum_{i=1}^{m} |a_i|^2 - \frac{1}{m} \sum_{i=1}^{m} |b_i|^2 \right| < \varepsilon.$$

Proof. For all $i \in \{1, \ldots, m\}$ we have

$$\begin{aligned} \left| |a_i|^2 - |b_i|^2 \right| &= |a_i + b_i||a_i - b_i| \\ &\leq (|a_i| + |b_i|) \frac{\varepsilon}{2L} \\ &\leq 2L \frac{\varepsilon}{2L} \\ &= \varepsilon. \end{aligned}$$

Thus

$$\begin{aligned} \left| \frac{1}{m} \sum_{i=1}^{m} |a_i|^2 - \frac{1}{m} \sum_{i=1}^{m} |b_i|^2 \right| &\leq \frac{1}{m} \sum_{i=1}^{m} \left| |a_i|^2 - |b_i|^2 \right| \\ &\leq \frac{1}{m} m\varepsilon \\ &\leq \varepsilon. \end{aligned}$$

□

We now prove the main result for this subsection.

Theorem 5. *Let $f : G \to \mathbb{C}$. Then there exists constants c_1 and c_2 such that if $m_1 \geq c_1 \frac{M^2}{\theta^2} \log(4/\delta)$ and $m_2 \geq c_2 \frac{M^2}{\theta^2} \log(4m_1/\delta)$ then*

$$\Pr \left[\left| B_{m_1, m_2}(z + H) - \|f_{z+H}\|_2^2 \right| > \frac{\theta}{4} \right] < \delta.$$

Proof. Since

$$\left|B_{m_1,m_2}(z+H) - \|f_{z+H}\|_2^2\right| \leq \left|B_{m_1,m_2}(z+H) - \frac{1}{m_1}\sum_{i=1}^{m_1}|f_{z+H}(x_i)|^2\right| + \left|\frac{1}{m_1}\sum_{i=1}^{m_1}|f_{z+H}(x_i)|^2 - \|f_{z+H}\|_2^2\right|$$

it suffices to show that each of the summands on the RHS is at most $\theta/8$ with high probability.

Recall that $\|f_{z+H}\|_2^2 = \mathbb{E}_{x\in G}|f_{z+H}(x_i)|^2$. To apply the Chernoff-Hoeffding bounds to the second summand, we need an upper bound for $|f_{z+H}(x_i)|^2$. Since $f_{z+H}(x_i) = \mathbb{E}_{y\in H^\perp} f(y)\chi_z(x-y)$ we know that $|f_{z+H}(x_i)| \leq \max_{y\in G}|f(y)| \leq \sqrt{M}$. Thus $|f_{z+H}(x_i)|^2 \leq M$. Choose $m_1 \geq \frac{2M^2}{(\theta/8)^2}\log(4/\delta)$ the Chernoff-Hoeffding bounds tell us that

$$\left|\frac{1}{m_1}\sum_{i=1}|f_{z+H}(x_i)|^2 + \|f_{z+H}\|_2^2\right| < \frac{\theta}{8}$$

with probability at least $1-\delta/2$. Choose $m_2 \geq \frac{2M}{(\theta/(16\sqrt{M}))^2}\log(4m_1/\delta)$ we know that, for each i,

$$|A_{m_2}(x_i, z+H) - f_{z+H}(x_i)| \leq \frac{\theta}{16\sqrt{M}}$$

with probability at least $1-\delta/(2m_1)$. We now apply Lemma 1 which tells us that

$$\left||A_{m_2}(x_i,z+H)|^2 - |f_{z+H}(x_i)|^2\right| \leq \frac{\theta}{8}.$$

It follows that the above equation holds for every i simultaneously with probability at least $1-\delta/2$. Hence

$$\begin{aligned}\left|B_{m_1,m_2}(z+H) - \frac{1}{m_1}\sum_{i=1}^{m_1}|f_{z+H}(x_i)|^2\right| &= \left|\frac{1}{m_1}\sum_{i=1}^{m_1}|A_{m_2}(x_i,z+H)|^2 - \frac{1}{m_1}\sum_{i=1}^{m_1}|f_{z+H}(x_i)|^2\right| \\ &\leq \frac{1}{m_1}\sum_{i=1}^{m_1}\left||A_{m_2}(x_i,z+H)|^2 - |f_{z+H}(x_i)|^2\right| \\ &\leq \frac{\theta}{8}.\end{aligned}$$

□

3.4.2 The running time of the algorithm

We have shown that we can estimate $\|f_{z+H}\|_2^2$ with an error of at most $\theta/4$. Thus if $\|f_{z+H}\|_2^2 > \theta$ then we know with probability at least $1-\delta$ that $|B_{m_1,m_2}(z+H)| > 3\theta/4$. And if $\|f_{z+H}\|_2^2 < \theta/2$ then $|B_{m_1,m_2}(z+H)| < 3\theta/4$.

This shows that the algorithm performs the right recursive calls with high probability. We now bound the number of recursive calls at each level in the recursion tree.

Lemma 2. *Let $f : G \to \mathbb{C}$.*

- *There are at most $2\|f\|_2^2/\theta$ values of α such that $|\widehat{f}(\alpha)|^2 \geq \theta/2$.*
- *Let $H \leq G$. There are at most $2\|f\|_2^2/\theta$ cosets $z+H$ such that $\|f_{z+H}\|_2^2 \geq \theta/2$.*

Proof. We prove the second claim first. Suppose it is not true. Let Z be a set of coset representatives for H in G. Then $\|f\|_2^2 = \sum_{z\in Z} \|f_{z+H}\|_2^2 > \frac{2\|f\|_2^2}{\theta}\frac{\theta}{2} = \|f\|_2^2$, a contradiction. The first claim is just a special case of the second when $H = 0$. □

We finally prove the running time of the algorithm.

Theorem 6. *The running time of Algorithm 1 is polynomial in p, M/θ, $\log|G|$ and $\log(1/\delta)$.*

Proof. From Lemma 2 we know for each $i = 1, \ldots, n$ the number of cosets, $z+H_i$, such that $\|f_{z+H_i}\|_2^2 > \theta/2$ is at most $2\|f\|_2^2/\theta$. We can bound the norm of f by $\|f\|_2^2 = \frac{1}{|G|}\sum_{x\in G}|f(x)|^2 \leq M$ so we have $2\|f\|_2^2/\theta \leq 2M/\theta$. Since $|H_i : H_{i-1}| \leq p$ this bounds the number of calls to EstNormSq by $2Mpn/\theta$ (recall that n is the length of the chain of subgroups $H_1 \leq H_2 \leq \cdots H_n$).

Since there are at most $2Mpn/\theta$ calls to EstNormSq, in order to guarantee a total error probability of no more than δ we need to choose m_1 and m_2 so that the probability of error on any particular call is no more than $\delta\theta/(2Mpn)$.

Hence the running time is

$$O\left(\frac{pnM}{\theta}m_1m_2\right) = O\left(\frac{pnM}{\theta}\frac{M^2}{\theta^2}\log\left(\frac{4}{\delta'}\right)\frac{M^2}{\theta^2}\log\left(\frac{4m_1}{\delta'}\right)\right),$$

where $\delta' = \delta\theta/(2Mpn)$. This formula is polynomial in p, n, M/θ and $\log(1/\delta)$. The result now follows since $n \leq \lfloor \log_2|G| \rfloor + 1$. □

3.5 Estimating the heavy coefficients once they have been identified

Algorithm 1 outputs a list of $\alpha_1, \ldots, \alpha_r \in G$ which contains all $\alpha \in G$ such that $|\widehat{f}(\alpha)|^2 > \theta$ and does not contain any $\alpha \in G$ for which $|\widehat{f}(\alpha)|^2 < \theta/2$. Once we have this list we can estimate the values $\widehat{f}(\alpha_i)$ by using the fact that $\widehat{f}(\alpha_i) = \langle f, \chi_{\alpha_i} \rangle = \frac{1}{|G|} \sum_{y \in G} f(y)\overline{\chi_{\alpha_i}(y)}$. By the Chernoff-Hoeffding bounds if we choose $m \geq \frac{2M}{\varepsilon^2} \log(2/\delta)$ and randomly select $y_j \in G$ for $1 \leq j \leq m$, then with probability $1 - \delta$ we have

$$\left| \widehat{f}(\alpha_i) - \frac{1}{m} \sum_{j=1}^{m} f(y_j)\overline{\chi_{\alpha_i}(y_j)} \right| < \varepsilon.$$

This allows us to estimate the coefficients with arbitrary precision.

Chapter 4

The AGS algorithm

Algorithm 1 from the previous chapter uses a divide and conquer approach to determine the heavy Fourier coefficients of $f : G \to \mathbb{C}$. The group G is divided into disjoints sets and the algorithm determines which sets contain no heavy coefficients and discards them. The remaining sets are then recursively searched to find the heavy coefficients. In Algorithm 1 we require that the disjoint sets are cosets of some subgroup of G. This was the approach in [19] and [21] (although, they did not use the language of group theory).

If $G = \mathbb{Z}_p$, where p is a large prime, then the only subgroups of G are the trivial subgroup and $\mathbb{Z}_p$ itself. This means that the algorithm will divide G into the cosets of the trivial subgroup: $\{0\}, \{1\}, \{2\}, \ldots, \{p-1\}$ and search each one. This will take $O(p)$ time, whereas we want the algorithm to run in time polynomial in $\log p$.

In [5] Akavia, Goldwasser and Shafra give an algorithm which finds the heavy Fourier coefficients of a function $f : \mathbb{Z}_n \to \{-1, 1\}$ for any $n \in \mathbb{N}$ (our convention is that $\mathbb{N}$ does not include 0) and the running time is polynomial in $\log n$. We will call this the AGS algorithm. A patent application for the AGS algorithm was filed in 2005 [12].

The paper [5] is relatively brief. Akavia's PhD thesis [2, Chapter 3] goes into more detail and generalises the algorithm slightly. The precise statement of the theorem, as in [2], is given below.

Theorem 7. *There exists an algorithm which, when given oracle access to the function $f : G \to \mathbb{C}$, finds all the θ-heavy Fourier coefficients of f. The running time is polynomial in $\log |G|$, $1/\theta$ and $\|f\|_\infty = \max_{x \in G} |f(x)|$.*

The goal of this chapter is to give a sketch of the algorithm and show how it

is related to the algorithms in [19] and [21]. Although the algorithm works for any $f : G \to \mathbb{C}$ we will focus on the case where $G = \mathbb{Z}_p$ where p is prime. We refer the reader to [2, Chapter 3] for more details.

4.1 The AGS algorithm

At a high level the algorithm works the same as Algorithm 1 (see Section 3.3.1 for a high level description of Algorithm 1). However, since we can not filter out cosets of $\mathbb{Z}_p$ as described in Section 3.3.3 we will instead filter out intervals. That is, we define $A = \{0, 1, \ldots \lfloor p/2 \rfloor\}$ and $B = \{\lfloor p/2 \rfloor + 1, \lfloor p/2 \rfloor + 2, \ldots, p-1\}$ and then estimate the norm of f_A and f_B. By Parseval's identity $\|f_A\|_2^2 = \sum_{z \in A} |\widehat{f}(z)|^2$ so if A contains a θ-heavy coefficient then $\|f_A\|_2^2 \geq \theta$. If $\|f_A\|_2^2 \geq \theta$ then the algorithm recursively divides the interval in half. If $\|f_A\|_2^2 < \theta$ then the interval A is discarded and the search continues in B. The pseudocode is given below. To determine the θ-heavy coefficients of f, you simply call AGS($\mathbb{Z}_p$).

Algorithm 3: AGS

Input: $A = \{a, a+1, \ldots, b\} \subseteq \mathbb{Z}_p$.
if $a = b$ **then**
  return $\{a\}$
else
  Let $c = \lfloor (a+b)/2 \rfloor$
  Let $A_1 = \{a, a+1, \ldots, c\}$
  Let $A_2 = \{c, c+1, \ldots, b\}$
  Let $I = \{i \in \{1, 2\} \mid \text{AGSEstNormSq}(f_{A_i}) \geq \theta\}$
  return $\cup_{i \in I} \text{Heavy}(A_i)$

As in Algorithm 1 the difficult part is the estimation of $\|f_A\|_2^2$ and $\|f_B\|_2^2$. In Algorithm 1 we relied on the fact that A and B were cosets to prove that $\|f_A\|_2^2$ and $\|f_B\|_2^2$ can be estimated efficiently. Since A, B are now intervals, a different approach is used which we describe in the next section.

4.2 The subroutine AGSEstNormSq

Let $A = \{a, a+1, \ldots, b\}$. We will now demonstrate how to efficiently estimate $\|f_A\|_2^2$. As in Section 3.3.3 the idea is that $\|f_A\|_2^2 = \mathbb{E}_{x \in G} |f_A(x)|^2$ so if we can

estimate $f_A(x)$ for particular values of x then we can estimate $\|f_A\|_2^2$ using the Chernoff-Hoeffding bounds. We could naively try to estimate $f_A(x)$ by using the filter function $h_A(x) = \sum_{\alpha=a}^{b} \chi_\alpha(x)$, which is the $\mathbb{Z}_p$ analogue to equation (3.1). Then $f_A(x) = (f * h_A)(x)$ as in Section 3.3.3.

However, this naive approach does not work. We would like to approximate $(f*h_A)(x)$ by choosing m random elements $y_i \in G$ and calculating $\frac{1}{m}\sum_{i=1}^{m} f(y_i)h_A(x-y_i)$. Unfortunately, the Chernoff-Hoeffding bounds require that the number of samples, m, is polynomial in L, where L is an upper bound for $|h_A(x)f(x-y)|$. Thus, m would need to be too large since $|h_A(x)|$ could possibly be as large as $b-a$ (when, for example, $x=0$). This means that when searching the interval $\{0,1,\ldots,\lfloor p/2 \rfloor\}$ we would need m to be proportional to $p/2$. In Section 3.3.3 we got around this by using the fact that if A is a coset then h_A has small support. However, in general h_A does not have small support. Therefore a different filter function is needed.

We now present the filter function defined in [2, Definition 3.14]. This filter function can be seen as an adaptation of equation (3.3) to the $\mathbb{Z}_p$ case. Let $a, b \in \mathbb{Z}_p$ so $0 \leq a < b < p$. Let $c = \lfloor (a+b)/2 \rfloor$. Let $t = \left\lfloor \frac{N}{2(b-a)} \right\rfloor$. Define

$$h_{a,b}(x) = \begin{cases} \frac{p}{t}\chi_{-c}(x), & \text{if } 0 \leq x \leq t-1, \\ 0, & \text{otherwise.} \end{cases}$$

The idea is that this function satisfies the properties in Remark 4. In particular, it has Fourier coefficients which are close to 1 inside the interval $[a,b]$ and close to zero outside the interval. The set of Fourier coefficients inside the interval $[a,b]$ is called the passband. Figure 4.1 is a plot of the Fourier coefficients of $h_{a,b}$ and the passband is labelled on the diagram.

This means that when $h_{a,b}$ is convoluted with the function f the coefficients of f outside the interval $[a,b]$ are filtered out.

In [2, Chapter 4, Section 4] Akavia proves that the filter function $h_{a,b}$ has the desired properties. In particular, Akavia proves the following Lemma.

Lemma 3. *For all $x \in \mathbb{Z}_p$ define* $\mathrm{abs}(x) = \min(x, n-x)$. *If $b-a \leq p/2$ then $h_{a,b}$ satisfies the following properties:*

- *Pass Band: For all $\alpha \in \mathbb{Z}_p$ and $\gamma \in [0,1]$, if* $\mathrm{abs}(\alpha - \frac{a+b}{2}) \leq \gamma\frac{b-a}{2}$, *then* $|\widehat{h}_{a,b}(\alpha)|^2 > 1 - \frac{5}{6}\gamma^2$.
- *Fast decreasing: For all $\alpha \in \mathbb{Z}_p$, we have* $|\widehat{h}_{a,b}(\alpha)|^2 < \left(\frac{2(b-a)}{\mathrm{abs}\left(\alpha - \frac{a+b}{2}\right)}\right)$.

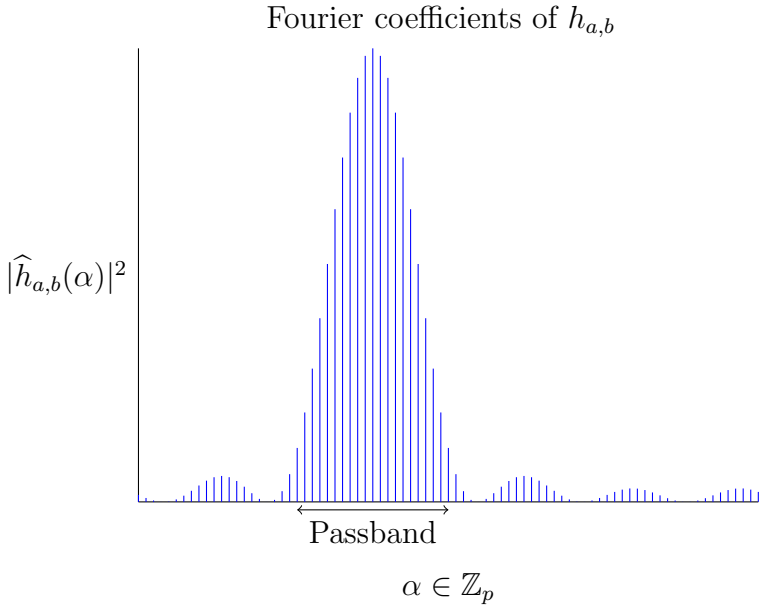


Figure 4.1: The Fourier coefficients of the filter function $h_{a,b}$. The passband is the interval $[a, b]$. Here $p = 83$, $a = 21$, $b = 41$.

Proof. See [2, Proposition 3.31]. □

Lemma 3 shows that $|\widehat{h}_{a,b}(\alpha)| \approx 1$ if $a \leq \alpha \leq b$ and $|\widehat{h}_{a,b}(\alpha)| \approx 0$ for α outside this interval. Since $\widehat{(f * h_{a,b})}(\alpha) = \widehat{f}(\alpha)\widehat{h}_{a,b}(\alpha)$, if $A = \{a, a+1, \ldots, b\}$ then $\widehat{f}_A(\alpha)$ is approximately equal to $\widehat{(f * h_{a,b})}(\alpha)$. Hence

$$\|f_A\|_2^2 = \sum_{a \leq \alpha \leq b} |\widehat{f}(\alpha)|^2 \approx \sum_{\alpha \in \mathbb{Z}_p} \widehat{(f * h_{a,b})}(\alpha) = \|f * h_{a,b}\|_2^2.$$

We also have

$$\begin{aligned} \|(f * h_{a,b})(x)\|_2^2 &= \mathop{\mathbb{E}}_{x \in \mathbb{Z}_p} \left| \mathop{\mathbb{E}}_{y \in \mathbb{Z}_p} [f(y) h_{a,b}(x-y)] \right|^2 \\ &= \mathop{\mathbb{E}}_{x \in \mathbb{Z}_p} \left| \mathop{\mathbb{E}}_{0 \leq y \leq t-1} [f(y) \chi_{-c}(x-y)] \right|^2, \end{aligned}$$

where $c = \lfloor (a+b)/2 \rfloor$.

So we can approximate $\|f_A\|_2^2$ as before by choosing m_1, m_2 sufficiently large, randomly choosing $x_i \in \mathbb{Z}_p$ where $1 \leq i \leq m_1$, randomly choosing $y_{ij} \in \{0, 1, \ldots, t-1\}$ for each i where $1 \leq j \leq m_2$ and calculating

$$\text{AGSEstNormSq}(f_A) = \frac{1}{m_1} \sum_{i=1}^{m_1} \left| \frac{1}{m_2} \sum_{j=1}^{m_2} f(y_{ij}) \chi_{-c}(x_i - y_{ij}) \right|^2 .$$

Algorithm 4: AGSEstNormSq

Input: $f_A : G \to \mathbb{C}$ where $A = \{a, a+1, \ldots, b\} \subseteq \mathbb{Z}_p$.
Choose $x_i \in \mathbb{Z}_p$ where $1 \leq i \leq m_1$
Let $t = \left\lfloor \frac{N}{2(b-a)} \right\rfloor$
For each i, choose $y_{ij} \in \{0, 1, \ldots, t-1\}$ where $1 \leq j \leq m_2$
Let $c = \lfloor (a+b)/2 \rfloor$
return $\frac{1}{m_1} \sum_{i=1}^{m_1} \left| \frac{1}{m_2} \sum_{j=1}^{m_2} f(y_{ij}) \chi_{-c}(x_i - y_{ij}) \right|^2$

The number of required samples m_1, m_2 is given in [2, Chapter 3]. Both m_1 and m_2 are polynomial in $1/\theta$, $\log |G|$ and $\|f\|_\infty = \max_{x \in G} |f(x)|$. This means that the running time of the algorithm is polynomial in $1/\theta$, $\log |G|$ and $\|f\|_\infty$.

Chapter 5

Modulus switching

In the literature, most algorithms for finding the heavy Fourier coefficients of $f : G \to \mathbb{C}$ are designed to work when $|G|$ is an integer power of 2. This is the most common case, especially in engineering applications. Since the community has invested serious effort to optimise sparse Fourier transform algorithms for this case it would be useful to take advantage of this work when computing sparse Fourier transforms where, for example $|G|$ is prime. While some of these algorithms could probably individually be modified to handle the case where $|G|$ is prime this would be quite difficult as many of the proofs become considerably more messy. We wish to develop a more general approach.

In this section we will explore a way of finding the heavy Fourier coefficients of $f : \mathbb{Z}_p \to \mathbb{C}$ when given access to an algorithm which finds the heavy Fourier coefficients of functions of the form $g : \mathbb{Z}_{2^k} \to \mathbb{C}$, for $k \in \mathbb{N}$. We therefore reduce the problem of finding the heavy Fourier coefficients of $f : \mathbb{Z}_n \to \mathbb{C}$ to the case where n is an integer power of 2. We call this *modulus switching*. We show how this technique can be used in the case where the heavy Fourier coefficients of f are sufficiently far apart from each other.

Throughout this chapter we will suppose that f is mapping from $\mathbb{Z}_p$ where p is some large prime. It is not essential that p is prime, although this is the application we have in mind. Let k be the smallest integer such that $p < 2^k$. Define $\widetilde{f} : \mathbb{Z}_{2^k} \to \mathbb{C}$ by $\widetilde{f}(x) = f(x)$ for $x \in \{0, 1, \ldots, p-1\}$ and $\widetilde{f}(x) = 0$ for $x \in \{p, p+1, \ldots, 2^k - 1\}$. If we compare the discrete Fourier transform of f and $\widetilde{f}$ in Figures 5.1 and 5.2 it appears that the heavy Fourier coefficients of $\widetilde{f}$ are somehow related to the heavy Fourier coefficients of f. The idea is to show that if we can find the heavy Fourier coefficients of $\widetilde{f}$ (using one of the many

algorithms which work when the domain size is a power of 2) then we can recover the coefficients of f.

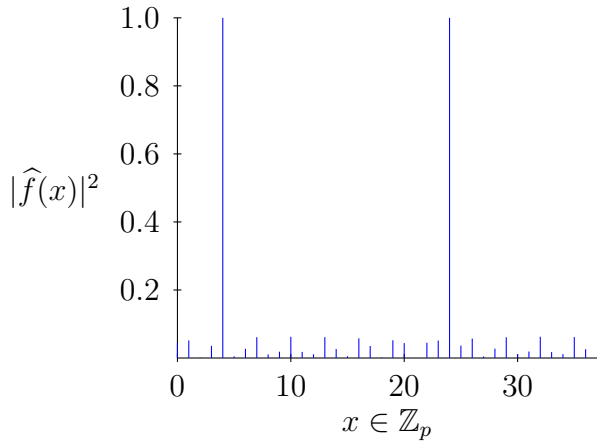


Figure 5.1: Plot of the magnitude of the Fourier coefficients of a function $f : \mathbb{Z}_p \to \mathbb{C}$. In this particular example $p = 37$.

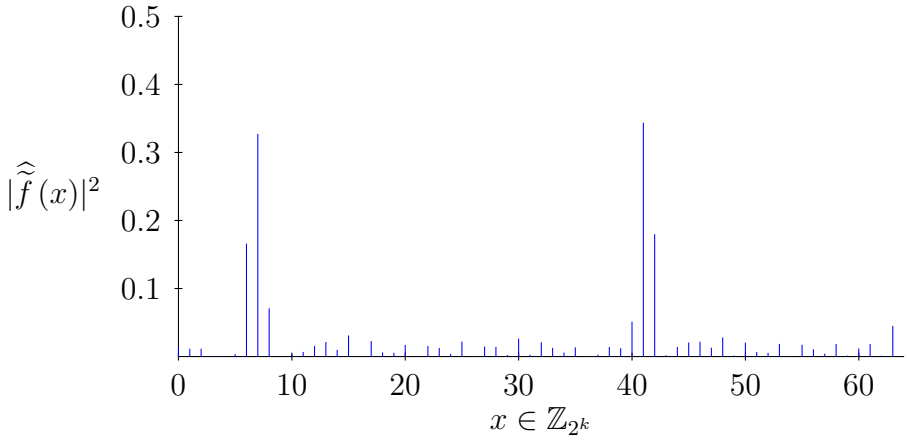


Figure 5.2: Plot of the magnitude of the Fourier coefficients of $\widetilde{f}$. Here $2^k = 64$. It appears as though two peaks correspond to the peaks in Figure 5.1.

5.1 Shor's Algorithm

The techniques in this section are similar to those used in Shor's quantum algorithm for factoring integers. In Shor's algorithm, the problem of factoring an integer N is reduced to the problem of locating the heavy Fourier coefficients of function $f : \mathbb{Z}_N \to \mathbb{C}$. In fact, the function is defined by $f(x) = a^x \pmod N$, where a is chosen randomly (for more details see the original paper [23]). The algorithm then uses the *quantum Fourier Transform* to calculate the Fourier coefficients of f. The coefficients are not calculated directly as the quantum Fourier transform only works when N is an integer power of 2. Instead, a related function $g : \mathbb{Z}_Q \to \mathbb{C}$ is constructed where Q is the smallest integer power of 2 such that $Q \geq N^2$. The function g is defined by $g(x) = f(x \pmod N)$. The quantum Fourier transform is used to compute the discrete Fourier transform of g and then this information is used to recover coefficients of the original function f.

In this chapter we use a similar technique but instead define the function g by $g(x) = f(x)$ if $0 \leq x \leq N-1$ and $g(x) = 0$ otherwise. Instead of choosing $Q \geq N^2$ we choose $Q \geq N$. We then show that if f statisfies certain conditions it is possible to recover the coefficients of f.

5.2 Tilde Notation

Given some $f : \mathbb{Z}_p \to \mathbb{C}$ we will define the function $\widetilde{f} : \mathbb{Z}_{2^k} \to \mathbb{C}$ in such a way so that if we know the heavy Fourier coefficients of $\widetilde{f}$ then this information can be used to recover the heavy Fourier coefficients of f.

Throughout this chapter we use p to refer to a fixed, positive integer and n will be fixed a positive integer such that $n > p$. For the particular application we have in mind p is a prime number and n is the smallest integer power of 2 such that $n > p$. However the assumption that p is prime and n is a power of 2 is not necessary for most of the theorems. We will therefore clearly state the restrictions on p and n in the hypotheses of each theorem.

Definition 15. *Let $p \in \mathbb{N}$. Let $n \in \mathbb{N}$ such that $n > p$. Let $f : \mathbb{Z}_p \to \mathbb{C}$. Define $\widetilde{f} : \mathbb{Z}_n \to \mathbb{C}$ by*

$$\widetilde{f}(z) = \begin{cases} f(z), & \text{if } 0 \leq z \leq p-1, \\ 0, & \text{if } p \leq z \leq n-1. \end{cases}$$

Note that this definition is different from the one used in Shor's algorithm.

It is easy to verify that the mapping $\widetilde{\cdot} : L^2(\mathbb{Z}_p) \to L^2(\mathbb{Z}_n)$ is both injective and linear. As the next lemma shows, it also behaves nicely with respect to the inner product.

Lemma 4. *Let $p \in \mathbb{N}$. Let $n \in \mathbb{N}$ such that $n > p$. Let $f, g \in L^2(\mathbb{Z}_p)$. Let $\widetilde{f}, \widetilde{g} \in L^2(\mathbb{Z}_n)$ be defined as in Definition 15. Then*

$$\langle \widetilde{f}, \widetilde{g} \rangle = \frac{p}{n} \langle f, g \rangle.$$

Proof. We have

$$\begin{aligned} \langle \widetilde{f}, \widetilde{g} \rangle &= \frac{1}{n} \sum_{z=0}^{n-1} \widetilde{f}(z) \overline{\widetilde{g}(z)} \\ &= \frac{1}{n} \sum_{z=0}^{p-1} f(z) \overline{g(z)} \\ &= \frac{p}{n} \langle f, g \rangle \end{aligned}$$

as required. □

Lemma 4 implies that if any two $g, h : \mathbb{Z}_p \to \mathbb{C}$ are orthogonal then $\widetilde{g}, \widetilde{h} : \mathbb{Z}_n \to \mathbb{C}$, are also orthogonal. In particular all the members of the set $\{\widetilde{\chi} : \mathbb{Z}_n \to \mathbb{C} \mid \chi : \mathbb{Z}_p \to \mathbb{C} \text{ is a character}\}$ are mutually orthogonal.

Corollary 4. *Let p, n and f be as in the Lemma. Then $\|\widetilde{f}\|_2^2 = \frac{p}{n} \|f\|_2^2$.*

Proof. By definition of the norm $\|\widetilde{f}\|_2^2 = \langle \widetilde{f}, \widetilde{f} \rangle = \frac{p}{n} \langle f, f \rangle = \frac{p}{n} \|f\|_2^2$. □

5.3 The Fourier coefficients of $\widetilde{\chi}_x$

Given some function $f : \mathbb{Z}_p \to \mathbb{C}$ we can write f as $f = \sum_{x \in \mathbb{Z}_p} \widehat{f}(x) \chi_x$. Since the map $f \mapsto \widetilde{f}$ is linear we then get $\widetilde{f} = \sum_{x \in \mathbb{Z}_p} \widehat{f}(x) \widetilde{\chi}_x$. This proves the following proposition.

Proposition 16. *Let $p \in \mathbb{N}$. Let $n \in \mathbb{N}$ such that $n > p$. Let $f : \mathbb{Z}_p \to \mathbb{C}$. The function f can be written as $f = \sum_{x \in \mathbb{Z}_p} \widehat{f}(x) \chi_x$. Then*

$$\widetilde{f} = \sum_{x \in \mathbb{Z}_p} \widehat{f}(x) \widetilde{\chi}_x.$$

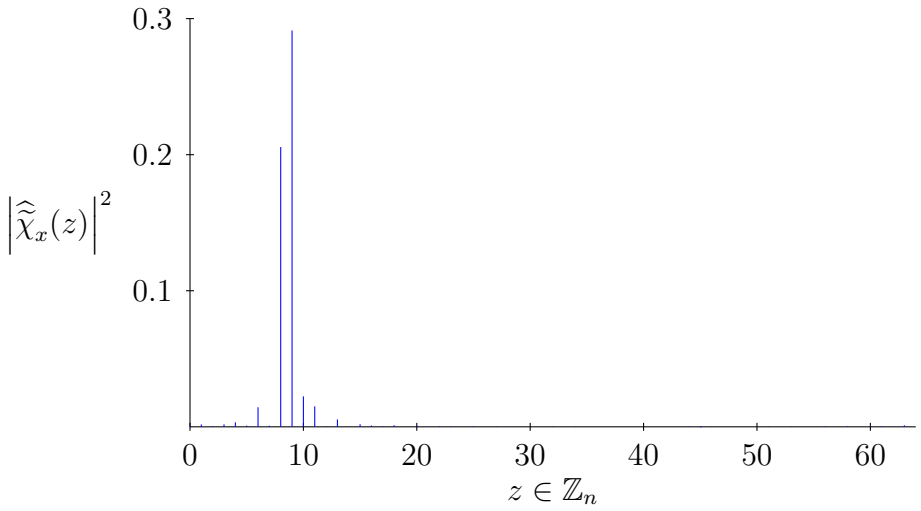


Figure 5.3: The magnitude of the Fourier coefficients for $\widehat{\widetilde{\chi}}_x$. Here $p = 37$, $n = 64$ and $x = 5$.

We will therefore first analyse the function $\widetilde{\chi}_x$ and later we will use this analysis to prove results about the function $\widetilde{f}$.

Figure 5.3 is a graph of the Fourier coefficients of $\widetilde{\chi}_x : \mathbb{Z}_n \to \mathbb{C}$. The graph appears to have a single peak and then decays rapidly.

In this section we prove two main results. The first, Proposition 18, is that the peak is at the closest integer to xn/p and the second (Lemma 6) is that the size of the y-th Fourier coefficients is inversely proportional to $|xn/p - y|$.

To avoid confusion we use χ for characters of $\mathbb{Z}_p$ and ψ for characters of $\mathbb{Z}_n$. So $\chi_w(z) = \exp(2\pi wz/p)$ and $\psi_x(z) = \exp(2\pi ixz/n)$.

Proposition 17. *Let $p \in \mathbb{N}$. Let $n \in \mathbb{N}$ such that $n > p$. Let $\chi_w : \mathbb{Z}_p \to \mathbb{C}$ be a character so $\widetilde{\chi}$ is a mapping from $\mathbb{Z}_n$ to $\mathbb{C}$. For all $x \in \mathbb{Z}_n$ we have*

$$|\widehat{\widetilde{\chi}}_w(x)|^2 = \begin{cases} \left(\frac{p}{n}\right)^2, & \text{if } \frac{w}{p} - \frac{x}{n} \in \mathbb{Z}, \\ \frac{1}{n^2} \frac{\sin^2\left(\frac{\pi p}{n} x\right)}{\sin^2\left(\frac{\pi}{n}\left(x - \frac{wn}{p}\right)\right)}, & \text{otherwise.} \end{cases} \tag{5.1}$$

Proof. Let $\psi_x(z) = \exp(2\pi ixz/n)$ be the x-th character of $\mathbb{Z}_n$. Then

$$\begin{aligned}\widehat{\widetilde{\chi}}(x) &= \langle \widetilde{\chi}_w, \psi_x \rangle \\ &= \frac{1}{n}\sum_{z=0}^{n-1} \widetilde{\chi_w}(z)\overline{\psi_x(z)} \\ &= \frac{1}{n}\sum_{z=0}^{p-1} \chi_w(z)\overline{\psi_x(z)}\end{aligned}$$

We now use the definition of the characters χ_w and ψ_x.

$$\begin{aligned}&= \frac{1}{n}\sum_{z=0}^{p-1} \exp(2\pi iwz/p)\exp(-2\pi ixz/n) \\ &= \frac{1}{n}\sum_{z=0}^{p-1} \exp\left(2\pi i\left(\frac{w}{p} - \frac{x}{n}\right)\right)^z.\end{aligned}$$

If $w/p - x/n \in \mathbb{Z}$ then $\exp\left(2\pi i\left(\frac{w}{p} - \frac{x}{n}\right)\right) = 1$ hence $\langle \widetilde{\chi}_w, \psi_x \rangle = p/n$. If $w/p - x/n \notin \mathbb{Z}$ we can use the formula for the sum of a geometric series. Then

$$\begin{aligned}\langle \widetilde{\chi}_w, \psi_x \rangle &= \frac{1}{n}\frac{1 - \exp\left(2\pi i\left(\frac{w}{p} - \frac{x}{n}\right)\right)^p}{1 - \exp\left(2\pi i\left(\frac{w}{p} - \frac{x}{n}\right)\right)} \\ &= \frac{1}{n}\frac{1 - \exp\left(2\pi ip\left(\frac{w}{p} - \frac{x}{n}\right)\right)}{1 - \exp\left(2\pi i\left(\frac{w}{p} - \frac{x}{n}\right)\right)}.\end{aligned}$$

For all $\theta \in \mathbb{R}$ we have $|1 - \exp(i\theta)|^2 = (1 - \cos(\theta))^2 + \sin^2(\theta) = 1 - 2\cos(\theta) + \cos^2(\theta) + \sin^2(\theta) = 2 - 2\cos(\theta) = 2 - 2\cos\left(2\frac{\theta}{2}\right) = 2 - 2\left(\cos^2\left(\frac{\theta}{2}\right) - \sin^2\left(\frac{\theta}{2}\right)\right) = 2\left(1 - \cos^2\left(\frac{\theta}{2}\right)\right) + 2\sin^2\left(\frac{\theta}{2}\right) = 2\sin^2\left(\frac{\theta}{2}\right) + 2\sin^2\left(\frac{\theta}{2}\right) = 4\sin^2\left(\frac{\theta}{2}\right)$. So by setting $\theta = 2\pi p\left(\frac{w}{p} - \frac{x}{n}\right)$ and $\theta = 2\pi\left(\frac{w}{p} - \frac{x}{n}\right)$ for the numerator and denominator re-

spectively we get

$$\begin{aligned}|\langle\widetilde{\chi}_w,\psi_x\rangle|^2 &= \left|\frac{1}{n}\frac{1-\exp\left(2\pi ip\left(\frac{w}{p}-\frac{x}{n}\right)\right)}{1-\exp\left(2\pi i\left(\frac{w}{p}-\frac{x}{n}\right)\right)}\right|^2\\ &= \frac{1}{n^2}\frac{\sin^2\left(\pi p\left(\frac{w}{p}-\frac{x}{n}\right)\right)}{\sin^2\left(\pi\left(\frac{w}{p}-\frac{x}{n}\right)\right)}\\ &= \frac{1}{n^2}\frac{\sin^2\left(\frac{\pi p}{n}x\right)}{\sin^2\left(\frac{\pi}{n}\left(x-\frac{wn}{p}\right)\right)}.\end{aligned}$$

□

The aim is to show that for fixed p, n, w the real number $|\langle\widetilde{\chi}_w,\psi_x\rangle|^2$ is large for a few $x\in\mathbb{Z}_n$ and small for all other $x\in\mathbb{Z}_n$. We will consider the right hand side of Equation (5.1) as a function of the *real* variable x. We therefore define $g:\mathbb{R}\to\mathbb{R}$ by

$$g(x)=\begin{cases}\left(\frac{p}{n}\right)^2, & \text{if } x\in\frac{wn}{p}+n\mathbb{Z},\\ \frac{1}{n^2}\frac{\sin^2\left(\frac{\pi p}{n}x\right)}{\sin^2\left(\frac{\pi}{n}\left(x-\frac{wn}{p}\right)\right)}, & \text{otherwise.}\end{cases}\tag{5.2}$$

(Note that $w/p-x/n\in\mathbb{Z}$ if and only if $x\in wn/p+n\mathbb{Z}=\{wn/p+\ell n\mid \ell\in\mathbb{Z}\}$.) The function g is periodic with period n. Figure 5.4 gives a plot of the Fourier coefficients of $\widetilde{\chi}_x$ (as in Figure 5.3) and g on the same axis.

Proposition 18. *Let $n\in\mathbb{N}$. Let $w\in\{0,1,\ldots,n-1\}$. Let $p\in\mathbb{Z}$ be such that $n/2<p<n$. Define $g:\mathbb{R}\to\mathbb{R}$ as in equation 5.2. Let x be the closest integer to wn/p. Then $g(x)\geq\frac{4}{\pi^2}\left(\frac{p}{n}\right)^2$.*

Proof. Let $x\in\mathbb{Z}$ be the integer closest to the real number wn/p. If $x=wn/p$ then $g(x)=\left(\frac{p}{n}\right)^2\geq\frac{4}{\pi^2}\left(\frac{p}{n}\right)^2$. If $x\neq wn/p$ then let $r=x-wn/p$. Then $0<|r|\leq 1/2$ and $x=wn/p+r$. Hence

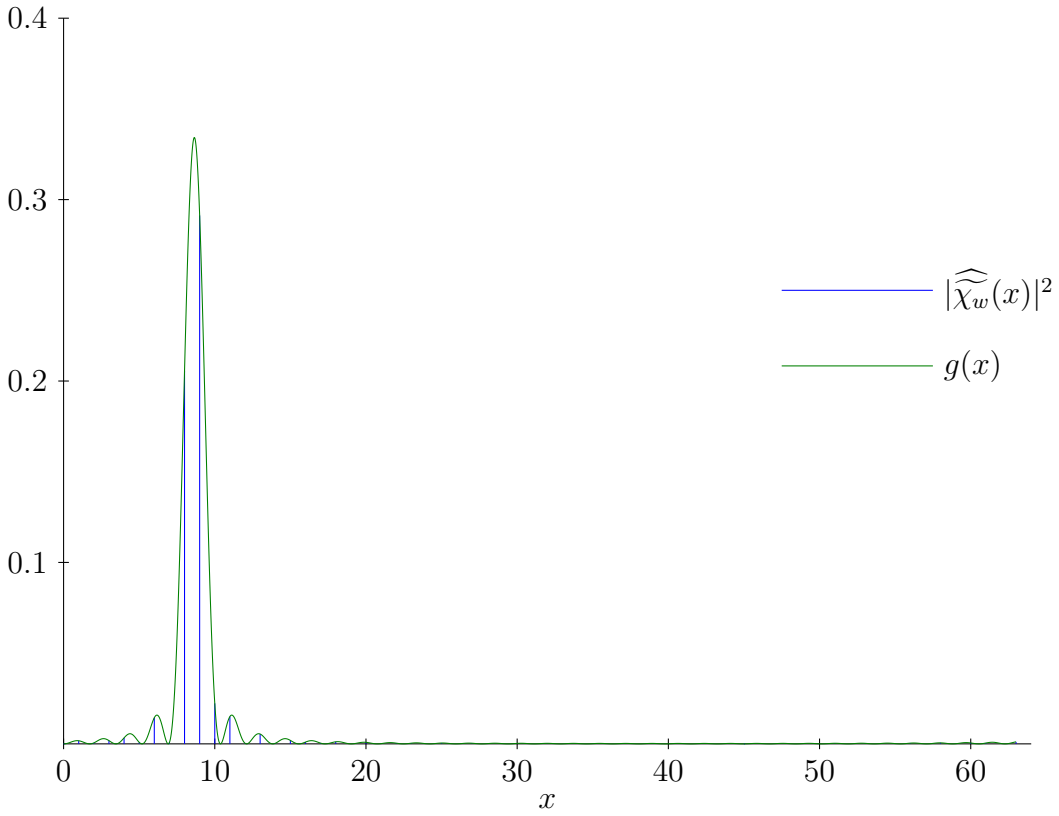


Figure 5.4: In blue: the magnitude of the Fourier coefficients of $\widetilde{\chi}_w$ where $p = 37$, $n = 64$ and $w = 5$. In green: the function g as defined in Equation 5.2.

$$\begin{aligned} g(x) &= \frac{1}{n^2}\frac{\sin^2\left(\frac{\pi p}{n}x\right)}{\sin^2\left(\frac{\pi}{n}\left(x-\frac{wn}{p}\right)\right)} \\ &= \frac{1}{n^2}\frac{\sin^2\left(\frac{\pi p}{n}\left(\frac{wn}{p}+r\right)\right)}{\sin^2\left(\frac{\pi}{n}\left(\frac{wn}{p}+r-\frac{wn}{p}\right)\right)} \\ &= \frac{1}{n^2}\frac{\sin^2\left(\frac{\pi p}{n}r\right)}{\sin^2\left(\frac{\pi}{n}r\right)}. \end{aligned}$$

We now use the inequalities $\sin^2(x) \leq x^2$, for all $x \in \mathbb{R}$ (this follows from the fact that $|\sin x| \leq |x|$ for all $x \in \mathbb{R}$) and $\sin^2(x) \geq \frac{4}{\pi^2}x^2$, for all $x \in [-\pi/2, \pi/2]$ (this follows from Jordan's inequality: $\sin(x) \geq \frac{2}{\pi}x$ for all $x \in [0, \pi/2]$) and apply them to the denominator and numerator respectively. We get

$$\begin{aligned} g(x) &\geq \frac{1}{n^2}\frac{\frac{4}{\pi^2}\left(\frac{\pi p}{n}r\right)^2}{\left(\frac{\pi}{n}r\right)^2} \\ &= \frac{4}{\pi^2}\frac{p^2}{n^2}. \end{aligned}$$

□

Corollary 5. *Let $p \in \mathbb{N}$. Let $n \in \mathbb{N}$ such that $n/2 < p < n$. Let $\chi_w : \mathbb{Z}_p \to \mathbb{C}$ be the w-th character of $\mathbb{Z}_p$. Let $x \in \mathbb{Z}_n$ be the closest integer to wn/p. Let $\psi_x : \mathbb{Z}_n \to \mathbb{C}$ be the x-th character in $\mathbb{Z}_n$. Then*

$$|\langle\widetilde{\chi_w}, \psi_x\rangle| \geq \frac{1}{\pi}.$$

Proof. The proposition gives $|\langle\widetilde{\chi_w}, \psi_x\rangle| = g(x) \geq \frac{2}{\pi}\frac{p}{n} \geq \frac{1}{\pi}$. □

Now that we have lower bounds, we now turn our attention to finding upper bounds for the Fourier coefficients of $\widetilde{\chi}_x$. Lemma 6 is the result we want, but first we prove a more technical lemma.

Definition 16. *Let $n \in \mathbb{N}$. For every $x \in \mathbb{R}$ let $[x]_n$ be the unique real number with the property that $x - [x]_n \in n\mathbb{Z}$ and $[x]_n$ is in the interval $(-n/2, n/2]$.*

Lemma 5. *Let $n \in \mathbb{N}$. Let $x \in \mathbb{R}$ such that $x \notin n\mathbb{Z}$. Then*

$$\frac{1}{\left|\sin(\frac{\pi}{n}x)\right|} \leq \frac{1}{\left|\frac{\pi}{n}[x]_n\right|} + 1.$$

Proof. The function $\left|\sin(\frac{\pi}{n}x)\right|$ is an even function with period n hence $|\sin(\frac{\pi}{n}x)| = |\sin(\frac{\pi}{n}[x]_n)|$ for all $x \in \mathbb{R}$. Since $u \leq \tan(u)$ for all $0 \leq u < \pi/2$ we have $u^2 \leq \tan^2(u) \Rightarrow \frac{1}{u^2} \geq \cot^2(u) = \operatorname{cosec}^2(x) - 1 \Rightarrow \frac{1}{\sin^2(u)} \leq \frac{1}{u^2} + 1$. Hence

$$\frac{1}{|\sin u|} = \sqrt{\frac{1}{\sin^2(u)}} \leq \sqrt{\frac{1}{u^2} + 1} \leq \sqrt{\frac{1}{u^2}} + \sqrt{1} = \frac{1}{|u|} + 1.$$

Finally

$$\frac{1}{\left|\sin(\frac{\pi}{n}x)\right|} = \frac{1}{\left|\sin(\frac{\pi}{n}[x]_n)\right|} \leq \frac{1}{\left|\frac{\pi}{n}[x]_n\right|} + 1.$$

□

Lemma 6. *Let $p \in \mathbb{N}$. Let $n \in \mathbb{N}$ such that $n/2 < p < n$. Let $\chi_w : \mathbb{Z}_p \to \mathbb{C}$ be a character of $\mathbb{Z}_p$. Let $x \in \mathbb{Z}_n$. Then*

$$|\widehat{\widetilde{\chi}}_w(x)| \leq \min\left(\frac{p}{n}, \frac{1}{\pi}\frac{1}{|[wn/p - x]_n|} + \frac{1}{n}\right).$$

Proof. First we observe that $|\widehat{\widetilde{\chi}}_w(x)|^2 \leq \sum_{w \in \mathbb{Z}_n} |\widehat{\widetilde{\chi}}_w(x)|^2 = \|\widetilde{\chi}_w\|_2^2 = \frac{p}{n}\|\chi_w\|_2^2 = \frac{p}{n}$. We now use the result from the Proposition 17 and the previous lemma which gives

$$|\langle \widetilde{\chi}_w, \psi_x \rangle| = \frac{1}{n}\left|\frac{\sin(\frac{\pi p}{n}x)}{\sin(\frac{\pi}{n}(x - wn/p))}\right| \leq \frac{1}{\pi}\frac{1}{|[x - wn/p]_n|} + \frac{1}{n}.$$

□

5.4 The Fourier coefficients of $\widetilde{f}$

The previous section tell us that the function $\widetilde{\chi}_w$ has a heavy Fourier coefficient at approximately wn/p. Since $\widetilde{f} = \sum_{w \in \mathbb{Z}_p} \widehat{f}(w)\widetilde{\chi}_w$ for any $f : \mathbb{Z}_p \to \mathbb{C}$ we would therefore hope that if $|\widehat{f}(w)|$ is large then $|\widehat{\widetilde{f}}(y)|$ is large, where y is the closest integer to wn/p.

To prove this we will suppose that there is some set $\Gamma \subseteq \mathbb{Z}_p$ such that for all $w \in \Gamma$ we have $|\widehat{f}(w)| > L$ for some $L \in \mathbb{R}$. That is, Γ contains all $w \in \mathbb{Z}_p$ for which $|\widehat{f}(w)|$ is large. We will also assume that for all $w \notin \Gamma$ we have $|\widehat{f}(w)| < \tau$, where τ is much smaller than L. Let $f_\Gamma = \sum_{w\in\Gamma} \widehat{f}(w)\chi_w$ and $f_{\mathbb{Z}_n\setminus\Gamma} = \sum_{w\in\mathbb{Z}_n\setminus\Gamma} \widehat{f}(w)\chi_w$. We can write f as $f = f_\Gamma + f_{\mathbb{Z}_n\setminus\Gamma}$ and hence $\widetilde{f} = \widetilde{f_\Gamma} + \widetilde{f}_{\mathbb{Z}_n\setminus\Gamma}$. In Proposition 19 we prove that all the Fourier coefficients of $\widetilde{f}_{\mathbb{Z}_n\setminus\Gamma}$ are $O(\tau \log p)$. In Proposition 8 we show that $|\widehat{\widetilde{f_\Gamma}}(y)|$ is $\Omega(L)$ where y is the closest integer to wn/p for some $w \in \Gamma$. Combining these two proposition we get that $|\widehat{\widetilde{f}}(y)|$ is $\Omega(L - \tau \log p)$. So if τ is sufficiently small then $|\widehat{\widetilde{f}}(y)|$ is large.

Proposition 19. *Let $p \in \mathbb{N}$. Let $n \in \mathbb{N}$ such that $n/2 < p < n$. Let $f = \sum_{w=0}^{p-1} \widehat{f}(w)\chi_w \in L^2(\mathbb{Z}_p)$. Suppose $|\widehat{f}(w)| < \tau$ for all $w \in \mathbb{Z}_p$. Then $|\widehat{\widetilde{f}}(y)| \leq \tau(3 + 2\log p)$ for all $y \in \mathbb{Z}_n$. So $|\widehat{\widetilde{f}}(y)|$ is $O(\tau \log p)$.*

Proof. Let $y \in \mathbb{Z}_n$. Then

$$\begin{aligned}
|\widehat{\widetilde{f}}(y)| &= |\langle \widetilde{f}, \psi_y \rangle| \\
&= \left| \langle \sum_{w=0}^{p-1} \widehat{f}(w)\widetilde{\chi}_w, \psi_y \rangle \right| \\
&= \left| \sum_{w=0}^{p-1} \widehat{f}(w) \langle \widetilde{\chi}_w, \psi_y \rangle \right| \\
&\leq \sum_{w=0}^{p-1} |\widehat{f}(w)| |\langle \widetilde{\chi}_w, \psi_y \rangle| \\
&\leq \tau \sum_{w=0}^{p-1} |\langle \widetilde{\chi}_w, \psi_y \rangle|
\end{aligned}$$

Let α be the closest integer to yp/n then so $\alpha = yp/n + \eta$ where $|\eta| \leq 1/2$. Then

$$\tau \sum_{w=0}^{p-1} |\langle \widetilde{\chi}_w, \psi_y \rangle| = \tau \sum_{w=0}^{p-1} |\langle \widetilde{\chi}_{\alpha+w}, \psi_y \rangle|,$$

where the indices $\alpha + w$ are taken mod p.

$$\begin{aligned}
\tau \sum_{w=0}^{p-1} |\langle \widetilde{\chi}_{\alpha+w}, \psi_y \rangle| &\leq \tau \left(3\sqrt{\frac{p}{n}} + \sum_{w=2}^{p-2} |\langle \widetilde{\chi}_{\alpha+w}, \psi_y \rangle| \right) \\
&\leq \tau \left(3 + \sum_{w=2}^{p-2} \left(\frac{1}{\pi} \frac{1}{|[(\alpha+w)n/p - y]_n|} + \frac{1}{n} \right) \right) \\
&\leq \tau \left(3 + \sum_{w=2}^{p-2} \left(\frac{1}{\pi} \frac{1}{|[\frac{n}{p}(\alpha+w-yp/n)]_n|} + \frac{1}{n} \right) \right) \\
&\leq \tau \left(4 + \sum_{w=2}^{p-2} \frac{1}{|[\frac{n}{p}(w+\eta)]_p|} \right)
\end{aligned}$$

Note that $|[\frac{n}{p}(w+\eta)]_n| = \frac{n}{p}(w+\eta)$ for $w \in (2, p/2-\eta)$ and $|[\frac{n}{p}(w+\eta)]_n| = |\frac{n}{p}(w+\eta) - n| = n - \frac{n}{p}(w-\eta)$ for $w \in (p/2-\eta, p-2)$. Furthermore the function $\frac{1}{\left[\frac{n}{p}(w+\eta)\right]_n}$ is decreasing on the interval $(2, p/2-\eta)$ and increasing on the interval $(p/2-\eta, p-2)$. We can therefore split the sum into two parts and bound by an integral.

$$\begin{aligned}
\tau \left(4 + \sum_{w=2}^{p-2} \frac{1}{|[\frac{n}{p}(w+\eta)]_p|} \right) &= \tau \left(4\frac{p}{n} + \int_1^{p/2-\eta} \frac{dw}{\frac{n}{p}(w+\eta)} + \int_{p/2-\eta}^{n-1} \frac{dw}{\frac{n}{p}(n-w-\eta)} \right) \\
&\leq \tau \left(4\frac{p}{n} + \int_1^{p/2-\eta} \frac{dw}{w+\eta} + \int_{p/2-\eta}^{n-1} \frac{dw}{n-w-\eta} \right) \\
&= \tau \left(4 + 2\log(p/2) - \log(1+\eta) - \log(1-\eta) \right) \\
&= \tau \left(4 + 2\log(p) - 2\log(2) - \log(1+\eta) - \log(1-\eta) \right) \\
&\leq \tau(3 + 2\log p).
\end{aligned}$$

As required. □

We now need some terminology to describe the heavy Fourier coefficients of $f : \mathbb{Z}_p \to \mathbb{C}$ being "far apart". We use the following definition.

Definition 17. *We say a set $\Gamma \subseteq \mathbb{Z}_p$ is r-spread if $|[x-y]_p| < r$ implies $x = y$ for all $x, y \in \Gamma$.*

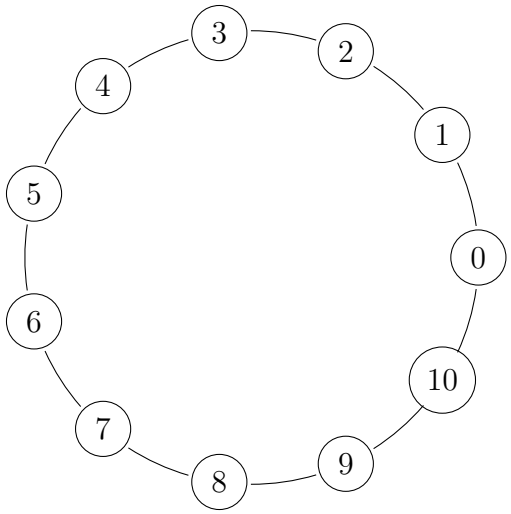


Figure 5.5: A graph of $\mathbb{Z}_{11}$.

If we draw $\mathbb{Z}_n$ as a graph, as in Figure 5.4, where the vertices are the elements of $\mathbb{Z}_n$ and the edges are between x and $x+1$ then the expression $|[x-y]_p|$ is equal to the distance between the vertices x and y in the graph.

This makes it easy to see that $\{0, 4, 8\} \subseteq \mathbb{Z}_{11}$ is a 4-spread set but not a 5-spread set and the set $\{1, 10\} \subseteq \mathbb{Z}_{11}$ is a 2-spread set but not a 3-spread set.

Theorem 8. *Let $p \in \mathbb{N}$. Let $n \in \mathbb{N}$ such that $n/2 < p < n$. Let $L \in \mathbb{R}$. Let $C > 1$. Let $r = 20C(1 + \log n) + 1$. Let $f : \mathbb{Z}_p \to \mathbb{C}$ and suppose there exists an r-spread set $\Gamma \subseteq \mathbb{Z}_p$ such that $\widehat{f}(w) = 0$ for all $w \notin \Gamma$. Moreover suppose that*

$$L \leq |\widehat{f}(w)| \leq CL$$

for all $w \in \Gamma$.

Let $y \in \Gamma$. Then $|\widehat{\widetilde{f}}\,(\alpha)| \geq \frac{1}{4}L$, where α is the closest integer to yn/p.

Proof. By the reverse triangle inequality

$$\begin{aligned}
|\widehat{\widetilde{f}}\,(\alpha)| &= |\langle \widetilde{f}, \psi_\alpha \rangle| \\
&= \left| \sum_{w \in \Gamma} \widehat{f}(w) \langle \widetilde{\chi}_w, \psi_\alpha \rangle \right| \\
&\geq \left| |\widehat{f}(y) \langle \widetilde{\chi}_y, \psi_\alpha \rangle| - \left| \sum_{\substack{w \in \Gamma \\ w \neq y}} \widehat{f}(w) \langle \widetilde{\chi}_w, \psi_\alpha \rangle \right| \right|
\end{aligned}$$

We know from Corollary 5 that $|\widehat{f}(y)\langle \widetilde{\chi}_y, \psi_\alpha \rangle| \geq \frac{1}{\pi} L$. Thus if we can upper bound $\left| \sum_{\substack{w \in \Gamma \\ w \neq y}} \widehat{f}(w) \langle \widetilde{\chi}_w, \psi_\alpha \rangle \right|$ by $L/20$ then this proves the proposition. We first use the hypothesis in the proposition

$$\left| \sum_{\substack{w \in \Gamma \\ w \neq y}} \widehat{f}(w) \langle \widetilde{\chi}_w, \psi_\alpha \rangle \right| \leq \sum_{\substack{w \in \Gamma \\ w \neq y}} \left| \widehat{f}(w) \langle \widetilde{\chi}_w, \psi_\alpha \rangle \right| \leq CL \sum_{\substack{w \in \Gamma \\ w \neq y}} |\langle \widetilde{\chi}_w, \psi_\alpha \rangle| \,.$$

We now upper bound $\sum_{\substack{w \in \Gamma \\ w \neq y}} |\langle \widetilde{\chi}_w, \psi_\alpha \rangle|$ using Lemma 6.

$$\begin{aligned}
\sum_{\substack{w \in \Gamma \\ w \neq y}} |\langle \widetilde{\chi}_w, \psi_\alpha \rangle| &\leq \sum_{\substack{w \in \Gamma \\ w \neq y}} \left(\frac{1}{\pi} \frac{1}{|[\alpha - wn/p]_n]|} + \frac{1}{n} \right) \\
&\leq \sum_{\substack{w \in \Gamma \\ w \neq y}} \frac{1}{\pi} \frac{1}{|[\alpha - wn/p]_n]|} + |\Gamma|/n \\
&\leq \sum_{\substack{w \in \Gamma \\ w \neq y}} \frac{1}{\pi} \frac{1}{|[\alpha - wn/p]_n]|} + \frac{1}{r}
\end{aligned}$$

Since α is the closest integer to yn/p we can write $\alpha = yn/p + \eta$ where $|\eta| \leq 1/2$. Thus

$$\sum_{\substack{w \in \Gamma \\ w \neq y}} |\langle \widetilde{\chi}_w, \psi_\alpha \rangle| \leq \sum_{\substack{w \in \Gamma \\ w \neq y}} \frac{1}{\pi} \frac{1}{|[\alpha - wn/p]_n]|} + \frac{1}{r} = \sum_{\substack{w \in \Gamma \\ w \neq y}} \frac{1}{\pi} \frac{1}{|[yn/p + \eta - wn/p]_n]|} + \frac{1}{r}.$$

Let $\varepsilon = \eta p/n$. Note that $|\varepsilon| \leq 1/2$. Hence

$$\sum_{\substack{w\in\Gamma\\ w\neq y}} |\langle \widetilde{\chi}_w, \psi_\alpha\rangle| \leq \sum_{\substack{w\in\Gamma\\ w\neq y}} \frac{1}{\pi} \frac{1}{|[\frac{n}{p}(y-w-\varepsilon)]_n]|} + \frac{1}{r}.$$

The expression $\frac{1}{|[\frac{n}{p}(w-y-\varepsilon)]_n]|}$ is maximised when w is close to y. Since $w \neq y$ and the members of Γ are all at least r apart thus the sum on the right hand side of this sum is maximised when $w - y = \pm r, \pm 2r, \pm 3r, \ldots$. We therefore have

$$\begin{aligned}
\sum_{\substack{w\in\Gamma\\ w\neq y}} \frac{1}{\pi} \frac{1}{|[\frac{n}{p}(w-y-\varepsilon)]_n]|} + \frac{1}{r} &\leq \sum_{\substack{-|\Gamma|/2\leq z\leq |\Gamma|/2\\ z\neq 0}} \frac{1}{\pi} \frac{1}{|[\frac{n}{p}(rz-\varepsilon)]_n]|} + \frac{1}{r} \\
&= \frac{1}{\pi} \frac{1}{|[\frac{n}{p}(r-\varepsilon)]_n]|} + \frac{1}{\pi} \frac{1}{|[\frac{n}{p}(-r-\varepsilon)]_n]|} \\
&\qquad + \sum_{\substack{-|\Gamma|/2\leq w\leq |\Gamma|/2\\ w\neq -1,0,1}} \frac{1}{\pi} \frac{1}{|[\frac{n}{p}(rw-\varepsilon)]_n]|} + \frac{1}{r} \\
&\leq \frac{2}{\pi} \frac{1}{\frac{n}{p}(r-1)} + \sum_{\substack{-|\Gamma|/2\leq w\leq |\Gamma|/2\\ w\neq -1,0,1}} \frac{1}{\pi} \frac{1}{|[\frac{n}{p}(rw-\varepsilon)]_n]|} + \frac{1}{r} \\
&\leq \frac{2}{\pi} \frac{1}{r-1} + \sum_{\substack{-|\Gamma|/2\leq w\leq |\Gamma|/2\\ w\neq -1,0,1}} \frac{1}{\pi} \frac{1}{|[\frac{n}{p}(rw-\varepsilon)]_n]|} + \frac{1}{r}.
\end{aligned}$$

This sum can be bounded with integrals. We use the fact that the function $\frac{1}{|[\frac{n}{p}(rw-\varepsilon)]_n]|}$ is decreasing on the interval $[1, |\Gamma|/2]$ and increasing on the interval

$[-|\Gamma|/2, -1]$. Then

$$\begin{aligned}
&\frac{2}{\pi}\frac{1}{r-1} + \sum_{\substack{-|\Gamma|/2 \le w \le |\Gamma|/2 \\ w \neq -1,0,1}} \frac{1}{\pi}\frac{1}{\left|\left[\frac{n}{p}(rw-\varepsilon)\right]_n\right|} + \frac{1}{r} \\
&\qquad \le \frac{1}{\pi}\frac{2}{r-1} + \frac{1}{\pi}\int_1^{|\Gamma|/2} \frac{dw}{\frac{n}{p}(rw-\varepsilon)} + \frac{1}{\pi}\int_{-|\Gamma|/2}^{-1} \frac{dw}{\frac{n}{p}|rw+\varepsilon|} \\
&\qquad \le \frac{1}{\pi}\frac{2}{r-1} + \frac{1}{\pi}\int_1^{|\Gamma|/2} \frac{dw}{rw-\varepsilon} + \frac{1}{\pi}\int_{-|\Gamma|/2}^{-1} \frac{dw}{|rw+\varepsilon|} \\
&\qquad = \frac{2}{\pi}\frac{1}{r-1} + \frac{1}{\pi r}\log\left(\frac{r|\Gamma|}{2} - \varepsilon\right) - \frac{1}{\pi r}\log(r-\varepsilon) \\
&\qquad\qquad\qquad + \frac{1}{\pi r}\log\left(\frac{r|\Gamma|}{2} + \varepsilon\right) - \frac{1}{\pi r}\log(r+\varepsilon) \\
&\qquad \le \frac{2}{\pi}\frac{1}{r-1} + \frac{2}{\pi r}\log\left(\frac{r|\Gamma|}{2} - 1\right) \\
&\qquad \le \frac{2}{\pi}\frac{1}{r-1} + \frac{2}{\pi r}\log(r|\Gamma| - 1).
\end{aligned}$$

Since the elements of Γ are at least r apart we know that $r|\Gamma| \le n$. So

$$\begin{aligned}
\frac{2}{\pi}\frac{1}{r-1} + \frac{2}{\pi r}\log(r|\Gamma|-1) &\le \frac{2}{\pi}\frac{1}{r-1} + \frac{2}{\pi r}\log n \\
&\le \frac{2}{\pi}\frac{1}{r-1}(1+\log n)
\end{aligned}$$

Substituting $r = 20C(\log n + 1) + 1$ gives us $\frac{1}{20C}$ as required. □

We can now combine to the previous two results to prove the main result of this chapter.

Proposition 20. *Let $L \in \mathbb{R}$. Let $C > 1$. Let $r = 20C(\log n + 1) + 1$. Let $f : \mathbb{Z}_p \to \mathbb{C}$ and suppose there exists an r-spread set $\Gamma \subseteq \mathbb{Z}_p$ such that $|\widehat{\widehat{f}}(w)| \le \tau < \frac{L}{20}\frac{1}{3+2\log p}$ for all $w \notin \Gamma$. Moreover suppose that $L \le |\widehat{\widehat{f}}(w)| \le CL$ for all $w \in \Gamma$.*

Let $y \in \Gamma$ and let α be the closest integer to yn/p. Then $|\widehat{\widehat{f}}(\alpha)| \ge \frac{1}{5}L$.

Proof. We can write f as $f = f_\Gamma + f_{\mathbb{Z}_n \setminus \Gamma}$. Then

$$|\widehat{\widetilde{f}}(\alpha)| = \left|\widehat{\widetilde{f}}_\Gamma(\alpha) + \widehat{\widetilde{f}}_{\mathbb{Z}_n \setminus \Gamma}(\alpha)\right| \geq \left||\widehat{\widetilde{f}}_\Gamma(\alpha)| - |\widehat{\widetilde{f}}_{\mathbb{Z}_n \setminus \Gamma}(\alpha)|\right|$$

We now apply Proposition 19 to $f_{\mathbb{Z}_n \setminus \Gamma}$ and Proposition 8 to f_Γ to obtain

$$\left||\widehat{\widetilde{f}}_\Gamma(\alpha)| - |\widehat{\widetilde{f}}_{\mathbb{Z}_n \setminus \Gamma}(\alpha)|\right| \geq \frac{1}{4}L - \frac{1}{20}L \geq \frac{1}{5}L.$$

□

Proposition 20 gives us a way of finding the heavy Fourier coefficients of $f : \mathbb{Z}_p \to \mathbb{C}$ for any $p \in \mathbb{N}$, provided f satisfies the hypotheses in the proposition. Let n be the smallest integer power of 2 greater than p. We can find the Fourier coefficients of $\widetilde{f} : \mathbb{Z}_n \to \mathbb{C}$ which are greater than $\frac{1}{5}L$ using some sparse Fourier transform algorithm which works for functions with domain size a power of 2. For every $\alpha \in \mathbb{Z}_n$ where $|\widehat{\widetilde{f}}(\alpha)| \geq \frac{1}{5}L$ we find the closest integer to $\alpha p/n$, call this integer y. Finally we check if $|\widehat{f}(y)| \geq L$ using the method described in Section 3.5. Proposition 20 guarantees that all Fourier coefficients of magnitude at least L will be recovered using this method.

We formalise this in the algorithm below.

Algorithm 5:

Input: A function $f : \mathbb{Z}_p \to \mathbb{C}$ that satisfies the hypotheses in Proposition 20.

Let n be the smallest integer power of 2 such that $n \geq p$.

Define $\widetilde{f} : \mathbb{Z}_n \to \mathbb{C}$ by $\widetilde{f}(x) = f(x)$ for $0 \leq x \leq p-1$ and $\widetilde{f}(x) = 0$ otherwise.

Use a sparse Fourier transform algorithm to find a list all $\alpha \in \mathbb{Z}_n$ such that $|\widehat{\widetilde{f}}(\alpha)| \geq L/5$.

Let $A = \emptyset$.

for *each* $\alpha \in \mathbb{Z}_n$ *such that* $|\widehat{\widetilde{f}}(\alpha)| \geq L/5$ **do**

  Let y be the closest integer to $\alpha p/n$

  Check if $|\widehat{f}(y)| \geq L$ using the method described in Section 3.5, if so, let $A = A \cup \{y\}$

return A

5.5 Further work

In Algorithm 5 we require that the set of all α such that $|\widehat{f}(\alpha)|$ is large must be an r-spread set (where r is specified in Proposition 20). The next step in this research will be to modify Algorithm 5 so that it works for a larger class of functions. Full details can be found in an upcoming paper by the author and Barak Shani.

It would also be interesting to compare how the time complexity of Algorithm 5 compares to the time complexity of the AGS algorithm.

Bibliography

[1] O. Abari, E. Hamed, H. Hassanieh, A. Agarwal, D. Katabi, A. P. Chandrakasan, and V. Stojanovic. 27.4 A 0.75-million-point fourier-transform chip for frequency-sparse signals. In *2014 IEEE International Solid- State Circuits Conference (ISSCC)*, pages 458–459. IEEE, 2014.

[2] A. Akavia. *Learning Noisy Characters, Multiplication Codes, and Cryptographic Hardcore Predicates* . PhD thesis, MASSACHUSETTS INSTITUTE OF TECHNOLOGY.

[3] A. Akavia. Advances in Cryptology - CRYPTO 2009: 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings. pages 337–354. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[4] A. Akavia. Deterministic Sparse Fourier Approximation Via Approximating Arithmetic Progressions. *IEEE Transactions on Information Theory*, 60(3):1733–1741, 2014.

[5] A. Akavia, S. Goldwasser, and S. Safra. Proving hard-core predicates using list decoding. In *44th Annual IEEE Symposium on Foundations of Computer Science - FOCS 2003*, pages 146–157. IEEE Computer. Soc, 2003.

[6] D. Boneh and R. Venkatesan. Hardness of Computing the Most Significant Bits of Secret Keys in Diffie-Hellman and Related Schemes. In *Advances in Cryptology — CRYPTO '96*, pages 129–142. Springer Berlin Heidelberg, Berlin, Heidelberg, Aug. 1996.

[7] P. Boufounos, V. Cevher, A. C. Gilbert, Y. Li, and M. J. Strauss. What's the Frequency, Kenneth?: Sublinear Fourier Sampling Off the Grid. *Algorithmica*, 73(2):261–288, 2015.

[8] W. E. Byerly. *An elementary treatise on Fourier's series and spherical, cylindrical, and ellipsoidal harmonics with applications to problems in mathematical physics*. Dover Publications, Inc, Mineola, New York, 2003.

[9] A. C. Gilbert, S. Guha, P. Indyk, S. Muthukrishnan, and M. Strauss. Near-optimal sparse fourier representations via sampling. In *the thiry-fourth annual ACM symposium*, pages 152–161, New York, New York, USA, 2002. ACM Press.

[10] A. C. Gilbert, S. Muthukrishnan, and M. Strauss. Improved time bounds for near-optimal sparse Fourier representations. *Optics & Photonics 2005*, 5914:59141A–59141A–15, 2005.

[11] O. Goldreich and L. A. Levin. A hard-core predicate for all one-way functions. In *the twenty-first annual ACM symposium*, pages 25–32, New York, New York, USA, 1989. ACM Press.

[12] S. Goldwasser, A. Akavia, S. Safra, YEDA Research & Dev. Co. Ltd, and Ramot At Tel Aviv University Ltd. Learning heavy fourier coefficients. US Patent Office, Nov. 2005.

[13] H. Hassanieh, P. Indyk, D. Katabi, and E. Price. Nearly optimal sparse fourier transform. In *the 44th symposium*, pages 563–578, New York, New York, USA, 2012. ACM Press.

[14] H. Hassanieh, P. Indyk, D. Katabi, and E. Price. Simple and Practical Algorithm for Sparse Fourier Transform. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1183–1194. Society for Industrial and Applied Mathematics, Philadelphia, PA, Dec. 2013.

[15] W. Hoeffding. Probability Inequalities for Sums of Bounded Random Variables. *Journal of the American Statistical Association*, 58(301):13–30, Apr. 2012.

[16] P. Indyk and M. Kapralov. Sample-Optimal Fourier Sampling in Any Constant Dimension. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 514–523. IEEE, 2014.

[17] P. Indyk, M. Kapralov, and E. Price. (Nearly) Sample-Optimal Sparse Fourier Transform. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM*

Symposium on Discrete Algorithms, pages 480–499. Society for Industrial and Applied Mathematics, Philadelphia, PA, Dec. 2013.

[18] M. A. Iwen. Combinatorial Sublinear-Time Fourier Algorithms. *Foundations of Computational Mathematics*, 10(3):303–338, Jan. 2010.

[19] E. Kushilevitz and Y. Mansour. Learning Decision Trees Using the Fourier Spectrum. *SIAM Journal on Computing*, 22(6):1331–1348, Dec. 1993.

[20] D. Lawlor, Y. Wang, and A. Christlieb. Adaptive Sub-Linear Time Fourier Algorithms. *Advances in Adaptive Data Analysis*, 05(01):1350003, Apr. 2013.

[21] Y. Mansour. Randomized Interpolation and Approximation of Sparse Polynomials. *SIAM Journal on Computing*, 24(2):357–368, Apr. 1995.

[22] S. Pawar and K. Ramchandran. Computing a k-sparse n-length Discrete Fourier Transform using at most 4k samples and O(k log k) complexity. *2013 IEEE International Symposium on Information Theory (ISIT)*, pages 464–468, 2013.

[23] P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *35th Annual Symposium on Foundations of Computer Science*, pages 124–134. IEEE Comput. Soc. Press, 1994.

[24] J. O. Smith III. *Mathematics of the Discrete Fourier Transform (DFT) with Audio Applications*. W3K Publishing, WA, 2007.

[25] G. Strang. *Computational science and engineering. Wellesley, MA: Wellesley*. Cambridge Press. Syed, 2007.

[26] A. Terras. *Fourier Analysis on Finite Groups and Applications*. Cambridge University Press, Cambridge, 2009.