2006

A Fast Fourier Transform for the Symmetric Group

Tristan Brand
Harvey Mudd College

A Fast Fourier Transform for the Symmetric Group

Tristan Brand

Michael Orrison, Advisor

Ghassan Sarkis, Reader

May, 2006

Harvey Mudd
College

Department of Mathematics

Abstract

A discrete Fourier transform, or DFT, is an isomorphism from a group algebra to a direct sum of matrix algebras. An algorithm that efficiently applies a DFT is called a fast Fourier transform, or FFT. The concept of a DFT will be introduced and examined from both a general and algebraic perspective. We will then present and analyze a specific FFT for the symmetric group.

Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank my adviser, Michael Orrison, for his invaluable advice, criticism, and encouragement during this project. I would also like to thank the Harvey Mudd math department for a great four years.

Chapter 1

Introduction

1.1 History of the Fast Fourier Transform

As with many other areas of modern mathematics, the fast Fourier transform owes its seed to the work of Gauss. At the beginning of the 19th century, Gauss was studying celestial orbits, specifically that of the asteroid Ceres. Ceres had suddenly vanished from astronomers' telescopes, and Gauss wanted to determine its path from previously known points. In order to do this he needed to interpolate the periodic orbit on n points. Due to the lack of advanced computational tools, this calculation would have to be done by hand and would require approximately n^2 arithmetic operations. Gauss was able to reduce his task by determining how to build the n-point interpolation from two $\frac{n}{2}$ point interpolations using a divide-and-conquer approach. Unfortunately, Gauss's work on this problem was never published. It was only in the middle of the 20th century, when Cooley and Tukey rediscovered and extended this type of algorithm, that Gauss' original work was revealed [8].

Cooley and Tukey's rediscovery was itself an interesting bit of history. In 1963, the Cold War was raging. In an attempt to curb rising tensions, a ban on nuclear testing for both sides was proposed. However, before the United States was willing to commit to this treaty, it was necessary that they have a method of detecting nuclear testing without having to physically visit the Soviet facilities. One idea was to use an offshore seismic detector to determine if nuclear tests were being performed. John Tukey, a member of Kennedy's Science Advisory Committee, began working on a method of analyzing the seismological time series data obtained from these detectors. He successfully came up with the mathematical methods necessary to do this. The actual implementation of these methods was charged to James Cooley under the guise of a more general problem due to security concerns. Cooley then proceeded to successfully code the algorithm. In 1965, Cooley and Tukey presented a paper with their results and the Cooley-Tukey fast Fourier transform, or FFT, was born. The algorithm was immediately and successfully applied to a variety of other problems and disciplines, helped by the parallel development of significantly more powerful analog to digital converters which had the power to produce digitized samples of time-varying voltage at hundreds of thousands of samples a second. A central reason that the FFT became such a widely used algorithm was that it was not patented and thus placed in the public domain immediately, allowing all researchers to access it and perhaps greatly hastening its maturation [9].

1.2 The Cooley-Tukey FFT for the Cyclic Group

We will examine DFTs for the cyclic group $G = \mathbb{Z}/n\mathbb{Z}$. The two main approaches to FFTs are decimation-in-time and decimation-in-frequency algorithms. The decimation-in-time approach was that originally pioneered by Cooley and Tukey in their 1965 paper. The terms come from the fact that one type of algorithm works by decomposing the time domain (the space where the data is found) while the other decomposes the frequency space (the space that contains the image of the DFT). The difference will become more apparent when we present an example of the two side by side.

We demonstrate how the Cooley-Tukey is performed on $\mathbb{Z}/n\mathbb{Z}$ for the case $n = pq$. This technique is easily generalizable to more factors. The presentation here is based on that given in [10] and [11]. The general idea will be to rewrite the Fourier transform on $\mathbb{Z}/n\mathbb{Z}$ using the Fourier transform on the subgroup $q\mathbb{Z}/n\mathbb{Z} \cong \mathbb{Z}/p\mathbb{Z}$ through the use of a change of variables. Let $f \in \mathbb{C}\mathbb{Z}/n\mathbb{Z}$ be a complex vector. Let D be the DFT. The kth component of $D(f) = \hat{f}$ is given by the formula

$$\hat{f}(k) = \sum_{j=0}^{n-1} f(j) e^{2\pi i jk/n}. \qquad (1)$$

The goal of the change of variables is to transform this one-dimensional formula into a two-dimensional formula which thus can be computed in two parts.

Define the variables j_1, j_2, k_1, k_2 as follows:

$$j = j_1 q + j_2, \qquad 0 \leq j_1 < p, 0 \leq j_2 < q;$$

$$k = k_2 p + k_1, \qquad 0 \leq k_2 < p, 0 \leq k_2 < q.$$

We can then rewrite (1) as

$$\hat{f}(k_2 p + k_1) = \sum_{j_2=0}^{q-1} e^{2\pi i j_2 (k_2 p + k_1)/n} \sum_{j_1=0}^{p-1} e^{2\pi i j_1 k_1 / p} f(j_1 q + j_2).$$

To emphasize the fact that we are changing variables to a two-dimensional system, we will denote $f(j_1 q + j_2)$ by $f(j_1, j_2)$. We can compute $\hat{f}$ in two stages. In the first stage, we calculate for each k_1 and j_2 the inner sum

$$\tilde{f}(k_1, j_2) = \sum_{j_1=0}^{p-1} e^{2\pi i j_1 k_1 / p} f(j_1, j_2).$$

This will require at most $p^2 q$ operations. In the second stage, for each k_1 and k_2 we calculate the outer sum

$$\hat{f}(k_1, k_2) = \sum_{j_2=0}^{q-1} e^{2\pi i j_2 (k_2 p + k_1)/n} \tilde{f}(k_1, j_2).$$

This will require an additional $q^2 p$ operations, for a total of $(p+q)pq$ operations. This is a clear savings over the $(pq)^2$ operations that would be required to compute the transform directly. The first stage is essentially a DFT on the subgroup $q\mathbb{Z}/n\mathbb{Z} \cong \mathbb{Z}/p\mathbb{Z}$ relative to the multiples of q, while the second stage is essentially of a DFT on $\mathbb{Z}/q\mathbb{Z}$. If n has additional factors, $n = abcde$, then we could simply set $n = (abc)(de)$ and then apply the same procedure on the smaller DFTs. In general, if $n = p_1 ... p_m$, then the algorithm requires roughly $n \sum_i p_i$ operations.

1.3 The Gentleman-Sande FFT for the Cyclic Group

We will briefly present the other major algorithm for abelian groups, the Gentleman-Sande. The Gentleman-Sande FFT works by decimating the frequency space rather than the time domain. On the cyclic group, it is equivalent to the Cooley-Tukey in terms of efficiency. On a deeper level, however, it operates using some very different ideas. The Gentleman-Sande FFT is not well understood in a general setting and lacks a solid algebraic framework. We introduce it here to contrast the way it breaks up the sums with the Cooley-Tukey; the presentation is based on that in the original Gentleman-Sande paper [4].

As with the Cooley-Tukey FFT, we consider the case where we are dealing with $\mathbb{Z}/n\mathbb{Z}$. This time let $n = ABC$ have three factors. Write $t = a + bA + cAB$ and $\hat{t} = \hat{c} + \hat{b}C + \hat{a}BC$, where $0 \leq a, \hat{a} \leq A - 1$, $0 \leq b, \hat{b} \leq B - 1$, and $0 \leq c, \hat{c} \leq C - 1$. Then we have

$$\hat{f}(\hat{c} + \hat{b}C + \hat{a}BC)$$

$$= \sum_{a=0}^{A-1} \sum_{b=0}^{B-1} \sum_{c=0}^{C-1} f(a + bA + cAB) e^{2\pi i((\hat{c}+\hat{b}C+\hat{a}BC)(a+bA+cAB)/ABC))}$$

$$= \sum_{a=0}^{A-1} e^{2\pi i a(\hat{c}+\hat{b}C+\hat{a}BC)/ABC} \sum_{b=0}^{B-1} e^{2\pi i(b(\hat{c}+\hat{b}C))/BC} \sum_{c=0}^{C-1} e^{2\pi i c\hat{c}/C} f(a + bA + cAB).$$

This is the Cooley-Tukey version. The Gentleman-Sande method changes this by factoring on the hatted variables instead, yielding

$$\sum_{a=0}^{A-1} e^{2\pi i a\hat{a}/A} \sum_{b=0}^{B-1} e^{2\pi i(\hat{b}(a+bA))/AB} \sum_{c=0}^{C-1} e^{2\pi i(\hat{c}(a+bA+cAB))/ABC} f(a + bA + cAB).$$

The algebra is tedious but it illustrates the practical difference between the two algorithms. When we move to more general cases, the basic algebraic trick of factoring different sets of variables may become a much deeper idea.

1.4 Overview

The discrete Fourier transform has been extensively studied since Cooley and Tukey first published their results in the 1960s. Discrete Fourier transforms have a wide range of application, from digital signal processing to quantum computing to analyzing data in voting theory. The goal in this section was to present a brief overview of the history and classical usage of the DFT. In Chapter 2, we will go into the algebraic theory behind the generalized discrete Fourier transform. Finally in Chapter 3 we will present and analyze an algorithm that efficiently implements a discrete Fourier transform for the symmetric group S_n.

Chapter 2

Theoretical Background

2.1 The Algebraic DFT

We will now examine the DFT from a group theoretic perspective. The reader is assumed to be familiar with the basic theory of modules and representations - a good reference for these is [3]. Consider a continuous periodic function

$$f : \mathbb{R} \to \mathbb{C}$$

sampled at n evenly spaced points $x_0, x_1, ..., x_{n-1}$ in a single period. The specific time period that these points are sampled at is not particularly relevant, as the Fourier coefficients will always be the same up to phase-shifts and magnitude [1]. This means that the DFT is invariant under cyclic shifts of the points $\{x_0, ..., x_{n-1}\}$. These cyclic shifts can be seen as the action of the group $\mathbb{Z}/n\mathbb{Z}$. The same action occurs if we replaced each x_i with the element $i \in \mathbb{Z}/n\mathbb{Z}$. Thus we can treat the points $x_0, ..., x_{n-1}$ as the elements of $\mathbb{Z}/n\mathbb{Z}$, allowing us to treat the function f as a function from $\mathbb{Z}/n\mathbb{Z}$ to $\mathbb{C}$. In fact, we can view f as an element of an object known as a group algebra. This is explored with greater depth in [1].

Definition 2.1.1. *Let G be a finite group with elements $\{g_1, g_2, ..., g_n\}$. The **group algebra** $\mathbb{C}G$ is defined to be the set of all formal sums of the form*

$$\sum_{i=1}^{n} \alpha_i g_i, \quad \textit{where } \alpha_i \in \mathbb{C}.$$

We define addition in the group algebra by

$$\sum_{i=1}^{n} \alpha_i g_i + \sum_{i=1}^{n} \beta_i g_i = \sum_{i=1}^{n} (\alpha_i + \beta_i) g_i.$$

Multiplication by a scalar (in $\mathbb{C}$) is given by

$$c\sum_{i=1}^{n}\alpha_i gi=\sum_{i=1}^{n}(c\alpha_i)g_i,$$

and multiplication in general is given by

$$\sum_{i=1}\alpha_i g_i\sum_{i=1}^{n}\beta_i g_i=\sum_{k=1}^{n}(\sum_{i,j,g_ig_j=g_k}\alpha_i\beta_j)g_k.$$

We can interpret $\mathbb{C}G$ as the set of all complex valued functions on G. Each coefficient α_i in the sum can be thought of as $f(g_i)$. By viewing f as an element of a group algebra we can use techniques from representation theory to interpret the DFT. This also gives us the capability to generalize the classical DFT to any group G.

Theorem 2.1.1. (Mashke) *Let G be a finite group. Every matrix representation*

$$D:G\to GL_n(\mathbb{C})$$

is equivalent to a direct sum of irreducible representations.

Any such representation corresponds to a $\mathbb{C}G$-module, and any $\mathbb{C}G$-module corresponds to some representation. Thus if we have an arbitrary $\mathbb{C}G$-module V, Maschke's theorem tells us that we can write

$$V\cong\oplus_{i=1}^{h}(R_i)^{n_i}$$

where h is the number of conjugacy classes in G, $R_1,...,R_h$ are the distinct irreducible representations of G, and $(R_i)^{n_i}$ indicates that there are n_i isomorphic copies of R_i in V.

Definition 2.1.2. *Let V be a $\mathbb{C}G$-module, and suppose*

$$V\cong\oplus_{i=1}^{h}(R_i)^{n_i}.$$

Then the spaces formed by each $R_i^{n_i}$ are the ***isotypic components*** *of V.*

Note that we may consider $\mathbb{C}G$ as a $\mathbb{C}G$-module by having it act on itself from the left [3]. The isotypic components correspond to minimal two-sided ideals of $\mathbb{C}G$ and the individual R_i's are minimal left ideals. If we consider our function $f\in\mathbb{C}G$, this result tells us that we can write f as the sum of elements in each of these components,

$$f=(r_{11}+...+r_{1n_1})+...+(r_{h1}+...+r_{hn_h})\qquad r_{ij}\in R_i.$$

We can extend this idea, viewing $\mathbb{C}G$ as the direct sum of an algebra consisting of block diagonal matrices as the following theorem demonstrates.

Theorem 2.1.2. (Wedderburn) *The group algebra $\mathbb{C}G$ of a finite group G is isomorphic to an algebra of block diagonal matrices:*

$$\mathbb{C}G \cong \bigoplus_{j=1}^{h} \mathbb{C}^{d_j \times d_j}.$$

The number h of blocks is equal to the number of conjugacy classes of G.

This theorem will allow us to formally define the generalized DFT.

Definition 2.1.3. *Every isomorphism*

$$D : \mathbb{C}G \to \oplus_{j=1}^{h} \mathbb{C}^{d_j \times d_j}$$

is a ***discrete Fourier transform****, or DFT for G. The space $\mathbb{C}G$ is often referred to as the time domain and $\oplus_{j=1}^{h} \mathbb{C}^{d_j \times d_j}$ as the frequency space.*

The Fourier coefficients of $f \in \mathbb{C}G$ are the coefficients in the matrix $D(f)$. An in-depth look at the motivation and proof of Wedderburn's theorem can be found in [1] and [3].

The basis for $\mathbb{C}G$ is fixed. We will use the usual basis that corresponds to the group elements of G. However, since we can pick a variety of bases for $\oplus_{i=1}^{h} \mathbb{C}^{d_j \times d_j}$, a DFT is not unique. Thus for each choice of basis we have a different DFT. We can view the DFT as a $|G| \times |G|$ matrix D. The columns of D correspond to the elements of G.

We note that D can be decomposed into irreducible representations D_j where D_j is defined as the component of D that maps $\mathbb{C}G$ to $\mathbb{C}^{d_j \times d_j}$, so

$$D = \oplus D_j.$$

Viewing D as a matrix, the rows of D can be parameterized using D_j:

$$\bigcup_{1 \leq j \leq h} \{(j,k,l) | 1 \leq k, l \leq d_j\}$$

where (j,k,l) corresponds to the position (k,l) in D_j. We can explicitly describe the entries in D as

$$D_{(j,k,l),g} = D_j(g)_{kl}$$

(see [1]). It is also useful to think about D as a linear transformation. When D is applied to $f \in \mathbb{C}G$, we get

$$Df = \hat{f}$$

where the components of $\hat{f}$ are the Fourier coefficients that we want to calculate. More explicitly, if we have $f \in \mathbb{C}G$, then f is a function with domain G and range $\mathbb{C}$. We can calculate the Fourier coefficients corresponding to the kth irreducible representation by

$$\hat{f}(D_j) = \sum_{g \in G} f(g) D_j(g).$$

We can calculate each individual entry,

$$\hat{f}(D_{j_{ik}}) = \sum_{g \in G} f(x) D_{ik}(x)$$

We can also calculate the inverse transform.

$$f(x) = \frac{1}{|G|} \sum_{j=1}^{h} d_j \operatorname{Tr}(\hat{f}(D_j) D_j(g^{-1}))$$

(see [10]). This conception of the Fourier transform easily allows us to place some bounds on the number of arithmetic operations required. Because applying the DFT is simply a matrix-vector product, we know that we will use no more then $2|G|^2$ arithmetic operations to apply the DFT. An algorithm which efficiently computes the Fourier transform called a fast Fourier transform or FFT. FFT research seeks to find algorithms which can reduce the bound $2|G|^2$ on the number of required operations.

2.2 Re-interpretation of the Cooley-Tukey

We will re-examine the Cooley-Tukey FFT described in the previous chapter from a group-theoretic perspective. Consider $G = \mathbb{Z}/n\mathbb{Z}$. Abelian groups are significantly easier to analyze because all of their irreducible representations are 1-dimensional. We want to use the DFT for $\mathbb{Z}/n\mathbb{Z}$ when we have a periodic continuous complex-valued function f sampled at n points. In this example we will again consider $n = pq$. We can view f as an element in $\mathbb{C}\mathbb{Z}_n$. The kth component of $\hat{f}$ is given by the formula

$$\hat{f}(k) = \sum_{j=0}^{n-1} f(j) e^{2\pi i jk/n}. \qquad (1)$$

The key here is in how we view the change in variables

$$\hat{j} = j_1 q + j_2, \qquad 0 \le j_1 < p, 0 \le j_2 < q;$$

$$\hat{k} = k_2 p + k_1, \qquad 0 \leq k_2 < p, 0 \leq k_2 < q.$$

We can view this as a factorization of the element $j \in \mathbb{Z}/n\mathbb{Z}$ as the sum of $j_1 q \in q\mathbb{Z}/n\mathbb{Z}$ with the coset representative j_2. We write $G = \mathbb{Z}/n\mathbb{Z}$ and $H = q\mathbb{Z}/n\mathbb{Z}$. The first change of variables then becomes

$$g = y + h, \quad y \in Y, \quad h \in H$$

where Y is the set of coset representatives of $\mathbb{Z}/n\mathbb{Z}$. The way that we can see the second change of variables (the one involving k) is by considering the restriction of representations. Since $H \leq G$, a representation of G restricted to H results in a representation of H. Going back to the $n = pq$ case, note that

$$e^{2\pi i j_1 q(k_2 p + k_1)/n} = e^{2\pi i j_1 k_1/p}.$$

The equality follows because if we look at

$$2\pi i j_1 q(k_2 p + k_1)/n = (2\pi i j_1 q k_2 p n + 2\pi i j_1 q k_1)/n,$$

the first term has a factor of q in it and is therefore 0 (mod q). The second term becomes $2\pi i j_1 q k_1/n = 2\pi i j_1 k_1 p$ since $q/n = p$. Thus we have

$$\hat{f}(k_2 p + k_1) = \sum_{j_2=0}^{q-1} e^{2\pi i j_2(k_2 p + k_1)/n} \sum_{j_1=0}^{p-1} e^{2\pi i j_1 k_1/p} f(j_1 q + j_2). \qquad (2)$$

The use of subgroup restriction will be a crucial point in constructing the FFT algorithm we present in Chapter 3.

2.3 Seminormal Representations

We will now examine the concept of a seminormal basis and the corresponding concept of a seminormal representation. In the end we want to apply this to the symmetric group, though the basics of this theory hold for all groups in general. The presentation here will closely follow that given by Ram in [7].

Let $1 = G_0 \leq G_1 \leq ... \leq G_n = G$ be a subgroup chain of finite groups. Let V be an irreducible $\mathbb{C}G$-module. If we consider the restriction of V to the group G_{n-1}, then we can write

$$V = V_1 \oplus ... \oplus V_k$$

where each V_i is an irreducible $\mathbb{C}G_{n-1}$-module. We can repeat this process for each V_i, decomposing them into irreducible modules for $\mathbb{C}G_{n-2}$, $\mathbb{C}G_{n-3}$, and so on.

Definition 2.3.1. *A **seminormal basis** of V with respect to the subgroup chain $G_0 \leq G_1 \leq ... \leq G_n$ is a basis $B = \{b_1, ..., b_r\}$ such that there exists a partition of B into subsets $B_1, ..., B_k$ with $V_i = \text{ span } (B_i)$ and*

$$V = V_1 \oplus ... \oplus V_k.$$

(See [7]). It is important to notice that we have an '$=$' sign rather then '$\cong$'. It is also required that there exists a partition of each B_i into subsets which equal the decomposition with restriction to G_{n-2}, and so on down the subgroup chain. We can say that B is an adapted basis to the particular subgroup chain since it is dependent on the specific choice for that chain. Note that as we go from G_i to G_{i-1} in the chain, the partitions become more and more refined.

Example 2.3.1. *Let $G = \mathbb{Z}/6\mathbb{Z}$. Consider the subgroup chain*

$$1 \leq 3\mathbb{Z}/6\mathbb{Z} \leq \mathbb{Z}/6\mathbb{Z}.$$

We explicitly describe a seminormal basis. Let ω be a primitive 6th root of unity (for example, $\omega = \frac{1+i\sqrt{3}}{2}$). We define the jth basis element b_j as

$$b_j = \begin{bmatrix} \omega^0 \\ \omega^j \\ \omega^{2j} \\ \omega^{3j} \\ \omega^{4j} \\ \omega^{5j} \end{bmatrix}.$$

Note that $\omega^3 = -1$ and $\omega^6 = 1$. The basis elements are

$$b_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad b_1 = \begin{bmatrix} 1 \\ \omega \\ \omega^2 \\ -1 \\ \omega^4 \\ \omega^5 \end{bmatrix} \quad b_2 = \begin{bmatrix} 1 \\ \omega^2 \\ \omega^4 \\ 1 \\ \omega^2 \\ \omega^4 \end{bmatrix}$$

$$b_3 = \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \end{bmatrix} \quad b_4 = \begin{bmatrix} 1 \\ \omega^4 \\ \omega^2 \\ 1 \\ \omega^4 \\ \omega^2 \end{bmatrix} \quad b_5 = \begin{bmatrix} 1 \\ \omega^5 \\ \omega^4 \\ -1 \\ \omega^2 \\ \omega \end{bmatrix}.$$

Each of these basis elements span a 1-dimensional irreducible representation. The entries are ordered such that the first entry corresponds to the element $\bar{0}$, the second to $\bar{1}$, etc. Note that $3\mathbb{Z}/6\mathbb{Z}$ is generated by the element $\bar{3}$. Thus when we consider the action of $\bar{3}$ on the entries, we note that it permutes the elements by shifting each by three. Under this action, we have that

$$b_0 \to b_0, \quad b_1 \to -b_1, \quad b_2 \to b_2,$$

$$b_3 \to -b_3, \quad b_4 \to b_4, \quad b_5 \to -b_5.$$

It is clear that $3\mathbb{Z}/6\mathbb{Z} \cong \mathbb{Z}/2\mathbb{Z}$, where $\bar{3}$ gets mapped to $\bar{1}$. The basis for $\mathbb{Z}/2\mathbb{Z}$ consists of

$$b_0' = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad b_1' = \begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

Note that the action of $\bar{3}$ on $\{b_0, b_2, b_4\}$ is equivalent to the action of $\bar{1}$ on b_0', and the action of $\bar{3}$ on $\{b_1, b_3, b_5\}$ is equivalent to the action of $\bar{1}$ on b_1'.

The representation corresponding to the seminormal basis is known as the seminormal representation. The fact that choosing a seminormal basis allows the restrictions to induce a partition on the basis will be vital when we introduce an algorithm that efficiently applies the seminormal *DFT* in the next chapter.

2.4 The Seminormal Representation of the Symmetric Group

In [7], Ram presents an in-depth discussion of the seminormal representations for the Weyl groups as well as the explicit construction of their representations. Because much of the general theory is not directly relevant to our results, we will only cover the materiel relating to the symmetric group in this section.

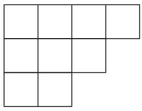
We know that the number of irreducible representations of a group is equal to the number of conjugacy classes of that group. In the case of $G = S_n$, the number of conjugacy classes is exactly the number of partitions of n. By using the partitions of $1, 2, ..., n-1, n$ for the subgroup chain $1 \leq S_1 \leq ... \leq S_n$ we will be able to build a diagram known as a Young lattice that serves to index these partitions and restrictions based on the partitions of the positive integers. We first introduce some definitions and terminology. The presentation here is based on that of Sagan in [13].

If $\lambda = (\lambda_1, ..., \lambda_k)$ is a partition of n, then we write $\lambda \dashv n$.

Definition 2.4.1. *Suppose $\lambda = (\lambda_1, ..., \lambda_k) \dashv n$. The* ***Ferrers diagram*** *or shape of λ is an array of cells into k left-justified rows with row i containing λ_i cells for all i.*

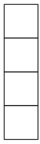
We can identify any particular cell by its coordinates (i, j).

Example 2.4.1. *The Ferrers diagram corresponding to the partition $(4, 3, 2)$ is*



.

The Ferrers diagram corresponding to the partition $(1, 1, 1, 1)$ is



.

Definition 2.4.2. *Suppose $\lambda \dashv n$. A* ***Young tableau*** *t* ***of shape*** *λ, is an array obtained by bijectively placing the numbers $1, 2, ...n$ in the boxes of the Ferrers diagram for λ.*

Example 2.4.2. *One possible Young tableau of shape $(3, 3, 2, 1)$ is*

$$\begin{array}{|c|c|c|}\hline 5 & 1 & 2 \\ \hline 7 & 9 & 6 \\ \hline 3 & 8 \\ \cline{1-2} 4 \\ \cline{1-1}\end{array}$$

Definition 2.4.3. *Suppose $\lambda \vdash n$. A* ***standard Young tableau*** *t of shape λ is a Young tableau of shape λ such that the entries in any row or column are strictly increasing.*

Example 2.4.3. *Consider the shape $(3, 2, 1)$. Then*

$$\begin{array}{|c|c|c|}\hline 1 & 2 & 3 \\ \hline 4 & 5 \\ \cline{1-2} 6 \\ \cline{1-1}\end{array}$$

is a standard Young tableaux. However,

$$\begin{array}{|c|c|c|}\hline 3 & 5 & 6 \\ \hline 1 & 4 \\ \cline{1-2} 2 \\ \cline{1-1}\end{array}$$

is not a standard Young tableaux since, for example, the entries in the second column are not strictly increasing.

If we have a Young tableau of shape λ, we will denote it t^{λ}. Note that since there are n boxes to fill with numbers in t^{λ} and upon filling a box we have exactly one less number to choose from in filling the next, there are precisely $n!$ Young tableaux for any shape $\lambda \dashv n$. The Young lattice is a graded, directed diagram of the Young tableaux corresponding to each S_m in the subgroup chain $1 = S_1 \leq S_2 \leq ... \leq S_n$. The Young lattice serves to index the irreducible representations of S_n and describe the manner in which they restrict under each subgroup in the chain. We will demonstrate this with an example.

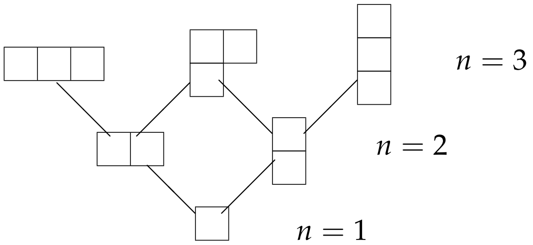
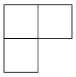
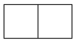


Figure 1. The Young lattice for S_3

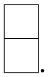
Example 2.4.4. *Figure 1 shows the Young lattice for S_3. A Ferrers diagram in row k corresponds to an irreducible representation of S_k. Each of those representations is equal, under restriction, to a direct sum of all of the representations that are connected to it from below. For example, the representation corresponding to*



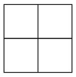
restricts to a direct sum of the representations corresponding to



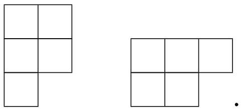
and

.

We have a line between two shapes at different levels if the shape at the higher level can be constructed by adding a single block. Recall that Ferrers diagrams must be *left-justified, so we can only add blocks below and to the right of existing blocks. For instance, on the level $n = 4$ the shape*



would connect up to the following two shapes in the level $n = 5$:



It is important to note that the restrictions are all multiplicity free in this case.

If we have a representation R_λ corresponding to some shape λ then the standard Young tableaux of shape λ each correspond to a basis element of R_λ - this is demonstrated in great detail in [13]. In fact, the entries in the standard Young tableaux tell us how that basis element is going to restrict down. Consider the standard Young tableau (for S_4)

$$\begin{array}{|c|c|}\hline 1 & 4 \\ \hline 2 \\ \cline{1-1} 3 \\ \cline{1-1}\end{array}\quad .$$

Then when we restrict to S_3, we drop the box containing a 4 and get

$$\begin{array}{|c|}\hline 1 \\ \hline 2 \\ \hline 3 \\ \hline\end{array}.$$

However, if we started with

$$\begin{array}{|c|c|}\hline 1 & 3 \\ \hline 2 \\ \cline{1-1} 4 \\ \cline{1-1}\end{array}$$

then when we restrict we would end up with

$$\begin{array}{|c|c|}\hline 1 & 3 \\ \hline 2 \\ \cline{1-1}\end{array}\quad .$$

We will make use of this idea in the next chapter.

We can now use the Young lattice to help us compute the seminormal representations and therefore the seminormal DFT for S_n. We will use the technique described in [5] to perform this computation explicitly for S_3.

Definition 2.4.4. *Let t_i^λ be a standard Young tableau. Then the* ***axial distance*** *between a and b, where a and b are numbers in the tableau, is defined to be the signed number of moves required to go from a to b, where a move is defined as going to an adjacent cell. A move up or to the right is positive, a move down or to the left is negative. Denote the axial distance between two entries a and b as $d_i(a, b)$.*

Example 2.4.5. *Consider the standard Young tableaux*

$$\begin{array}{|c|c|}\hline 1 & 2 \\ \hline 3 & 4 \\ \hline 5 \\ \cline{1-1}\end{array}\;.$$

Then

$$d(1,2) = 1$$
$$d(2,5) = -3$$
$$d(1,4) = -2$$
$$d(4,1) = 2.$$

One more definition is necessary before we can begin some computations.

Definition 2.4.5. *We define the* ***last letter ordering*** *of the standard Young tableaux by showing how to compare any two tableaux. Let t_i^λ and t_j^λ be two standard Young tableaux for n. If the entry n occurs in a higher row in t_i^λ than in t_j^λ, then t_i^λ occurs before t_j^λ in the ordering. If n occurs in the same row in both tableaux, then repeat the comparison for the entry $n - 1$ and so on.*

Example 2.4.6. *Consider the shape*

$$\begin{array}{|c|c|}\hline & \\ \hline & \\ \hline \\ \cline{1-1}\end{array}\;.$$

This shape has five associated standard Young tableaux:

$$\begin{array}{|c|c|}\hline 1 & 4 \\ \hline 2 & 5 \\ \hline 3 \\ \cline{1-1}\end{array}\quad
\begin{array}{|c|c|}\hline 1 & 2 \\ \hline 3 & 4 \\ \hline 5 \\ \cline{1-1}\end{array}\quad
\begin{array}{|c|c|}\hline 1 & 2 \\ \hline 3 & 5 \\ \hline 4 \\ \cline{1-1}\end{array}\quad
\begin{array}{|c|c|}\hline 1 & 3 \\ \hline 2 & 4 \\ \hline 5 \\ \cline{1-1}\end{array}\quad
\begin{array}{|c|c|}\hline 1 & 3 \\ \hline 2 & 5 \\ \hline 4 \\ \cline{1-1}\end{array}\;.$$

Then the last letter ordering is:

$$\begin{array}{|c|c|}\hline 1&4\\\hline 2&5\\\hline 3&\multicolumn{1}{c}{}\\\cline{1-1}\end{array}\;,\quad \begin{array}{|c|c|}\hline 1&3\\\hline 2&5\\\hline 4&\multicolumn{1}{c}{}\\\cline{1-1}\end{array}\;,\quad \begin{array}{|c|c|}\hline 1&2\\\hline 3&5\\\hline 4&\multicolumn{1}{c}{}\\\cline{1-1}\end{array}\;,\quad \begin{array}{|c|c|}\hline 1&3\\\hline 2&4\\\hline 5&\multicolumn{1}{c}{}\\\cline{1-1}\end{array}\;,\quad \begin{array}{|c|c|}\hline 1&2\\\hline 3&4\\\hline 5&\multicolumn{1}{c}{}\\\cline{1-1}\end{array}\;.$$

We now want to construct the seminormal representation for S_n. We will do this by calculating all the seminormal representations (with respect to each shape λ) for the elements in a generating set $G_{S_n} = \{s_1, ..., s_r\}$ for S_n. If x is any element in S_n, we have $x = s_{t_1}...s_{t_l}$ where $s_{t_i} \in S$; note that each element in this product is not necessarily distinct. If $D^\lambda(s_1), ..., D^\lambda(s_r)$ are the representations with respect to λ of the generating set, we can compute $D^\lambda(x)$ as

$$D^\lambda(x) = D^\lambda(s_{t_1}...s_{t_l}) = D^\lambda(s_{t_1})...D^\lambda(s_{t_l})$$

since D^λ is a homomorphism.

A convenient generating set for S_n is the set $G_{S_n} = \{(1,2), (2,3), ..., (n-1, n)\}$. We will now calculate the representations D^λ. Let $t_1^\lambda, ..., t_m^\lambda$ be the standard Young tableaux arranged in last letter order. Then we compute $D^\lambda((j-1,j))$ as follows:

1. $D_{ii}^\lambda((j-1,j)) = 1$ if t_i^λ contains $j-1$ and j in the same row.
2. $D_{ii}^\lambda((j-1,j)) = -1$ if t_i^λ contains $j-1$ and j in the same column.
3. If $i < k$ and $t_i^\lambda = (j-1,j)t_k^\lambda$, then we have the following submatrix:

$$\begin{bmatrix} D_{ii}^\lambda((j-1,j)) & D_{ik}^\lambda((j-1,j)) \\ D_{ki}^\lambda((j-1,j)) & D_{kk}^\lambda((j-1,j)) \end{bmatrix} = \begin{bmatrix} -d_i(j-1,j)^{-1} & 1 - d_i(j-1,j)^{-2} \\ 1 & d_i(j-1,j)^{-1} \end{bmatrix}.$$

Note that in the seminormal DFT, each column corresponds to a group element and each row to an entry of a representation. Thus once we have computed all of the representations for all of the elements in S_n, we construct the seminormal DFT simply by placing the entries from each representation in the corresponding location in the DFT matrix. This will become clearer when we go through an example.

Example 2.4.7. *We will construct the seminormal DFT for S_3. The generating set for S_3 is simply $\{(12), (23)\}$. We first examine the irreducible representation for S_3 corresponding to*

$$\begin{array}{|c|c|c|}\hline \ & \ & \ \\\hline\end{array}.$$

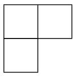
Note that there is only one standard Young tableau for this shape, specifically

$$\begin{array}{|c|c|c|}\hline 1 & 2 & 3 \\ \hline \end{array}.$$

Since all of the entries are in the same row, when we use the formula given above on (12) *and* (23) *we simply get one dimensional matrices with a* 1 *as their only entry. Thus the representations for all of the elements in* S_3 *corresponding to this shape are simply the 1* $\times$ *1 matrix*

$$[1].$$

The second irreducible representation corresponds to the shape



.

There are two standard Young tableaux for this shape, given in last letter order:

$$\begin{array}{|c|c|}\hline 1 & 3 \\ \hline 2 \\ \cline{1-1} \end{array} \qquad \begin{array}{|c|c|}\hline 1 & 2 \\ \hline 3 \\ \cline{1-1} \end{array}.$$

Using our formula, we get that

$$D^{(2,1)}((12)) = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

and

$$D^{(2,1)}((23)) = \begin{bmatrix} \frac{1}{2} & \frac{3}{4} \\ 1 & -\frac{1}{2} \end{bmatrix}.$$

Then by using the identities

$$(12) * (23) = (132),$$

$$(23) * (12) = (123),$$

$$(12) * (12) = 1,$$

$$(123) * (12) = (13),$$

we can compute the remaining representations corresponding to this shape:

$$D^{(2,1)}((123)) = \begin{bmatrix} -\frac{1}{2} & -\frac{3}{4} \\ 1 & -\frac{1}{2} \end{bmatrix}.$$

$$D^{(2,1)}((132)) = \begin{bmatrix} -\frac{1}{2} & \frac{3}{4} \\ -1 & -\frac{1}{2} \end{bmatrix}.$$

$$D^{(2,1)}(1) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

$$D^{(2,1)}(13) = \begin{bmatrix} \frac{1}{2} & -\frac{3}{4} \\ -1 & -\frac{1}{2} \end{bmatrix}.$$

Finally, we consider the representation $D^{(1,1,1)}$ corresponding to

$$\begin{array}{|c|} \hline \\ \hline \\ \hline \\ \hline \end{array}.$$

The only standard Young tableau we have is

$$\begin{array}{|c|} \hline 1 \\ \hline 2 \\ \hline 3 \\ \hline \end{array}.$$

Since all elements are in the same column, for both (12) and (23) we get 1×1 matrices with a -1. Then the corresponding matrices for the remaining elements are:

$$D^{(1,1,1)}((123)) = [1],$$
$$D^{(1,1,1)}((132)) = [1],$$
$$D^{(1,1,1)}(1) = [1],$$
$$D^{(1,1,1)}(13) = [-1].$$

We are now ready to construct the seminormal DFT for S_3. As stated earlier, each column of DFT_{S_3} will correspond to an element of S_3. The group ordering of the columns we will use is $1, (12), (13), (132), (23), (123)$. This is based on the right cosets of S_2 in S_3 with coset representatives $1, (13), (23)$. The rows correspond to the entries in each representation, the first row to the representation $D_{(3)}$, the second row to the first entry in $D_{(2,1)}$, etc. Thus we have that

$$DFT_{S_3} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & -\frac{3}{4} & \frac{3}{4} & \frac{3}{4} & -\frac{3}{4} \\ 0 & 0 & -1 & -1 & 1 & 1 \\ 1 & 1 & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ 1 & -1 & -1 & 1 & -1 & 1 \end{bmatrix}.$$

We now have prepared the theoretical tools necessary to understand the FFT algorithm that will be constructed and analyzed in the next chapter.

Chapter 3

The Algorithm

3.1 A General Framework

Let G be a finite group, and let H be a subgroup of G. We may write $\mathbb{C}G$ as a direct sum of irreducible $\mathbb{C}G$-modules:

$$\mathbb{C}G \cong R_1 \oplus R_2 \oplus ... \oplus R_k.$$

If we view $\mathbb{C}G$ as a $\mathbb{C}H$-module, each of the irreducible modules decompose into irreducible $\mathbb{C}H$-modules:

$$\mathbb{C}G \downarrow_H = (R_{11} \oplus ... \oplus R_{1a_1}) \oplus ... \oplus (R_{k1} \oplus ... \oplus R_{ka_k})$$

where R_{ij} occurs in the decomposition of R_i. Then with respect to some subgroup G_r of H, we could repeat this process. Therefore if we have a subgroup chain $0 = G_1 \leq ... \leq G_r \leq H \leq G$ we can recursively apply this process to create a decomposition with respect to each group in the chain:

$$\mathbb{C}G \cong R_1 \oplus R_2 \oplus ... \oplus R_k$$

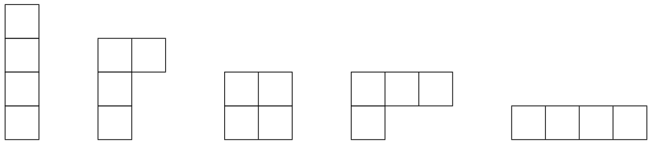
$$\mathbb{C}G \downarrow_H \cong (R_{11} \oplus ... \oplus R_{1a_1}) \oplus ... \oplus (R_{k1} \oplus ... \oplus R_{ka_k})$$

$$\mathbb{C}G \downarrow_{G_r} \cong (R_{111} \oplus ... R_{11b_1}) \oplus ... \oplus (R_{ka_k1} \oplus ... \oplus R_{ka_kb_k})$$

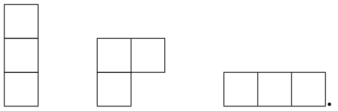
$$\vdots$$

We will make this clearer with an example.

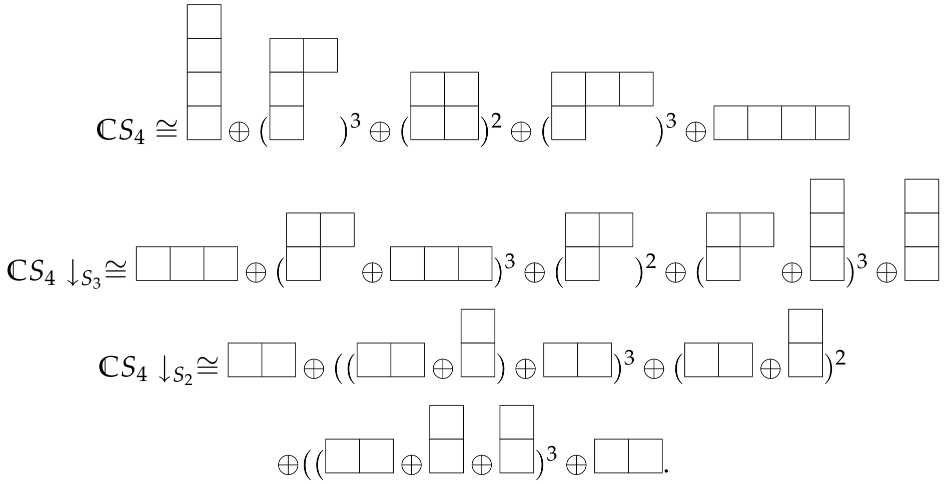
Example 3.1.1. *Consider $G = S_4$. Recall that we may use a Young lattice to encode how the irreducible representations of S_n decompose with respect to a semi-normal basis. There are five possible Young diagrams for S_4:*



and three possible Young diagrams for S_3:



We will use these Young diagrams directly in order to demonstrate the decomposition of $\mathbb{C}S_4$ with restriction to S_3 and S_2.


ℂS₄ ≅ ⊕ ( )³ ⊕ ( )² ⊕ ( )³ ⊕
ℂS₄ ↓S₃ ≅ ⊕ (⊕)³ ⊕ ( )² ⊕ (⊕)³ ⊕
ℂS₄ ↓S₂ ≅ ⊕ ((⊕) ⊕)³ ⊕ (⊕)²
⊕((⊕ ⊕)³ ⊕ .

We can clearly see here with the help of the Young diagrams how each representation restricts to a sum of irreducible representations of the next group in the subgroup chain. In the next section, we will take advantage of the structure that these restrictions create to develop a matrix factorization of the DFT for S_n.

We will now use this structure to construct an algorithm that applies the DFT of G by expressing it as a product of matrix factors.

3.2 Matrix Factorization

Using the theoretical tools that we have examined in the previous sections, we now turn our attention to the development of an FFT for G. Let G be a group and let $H \leq G$. Let DFT_G denote a DFT for G. Consider the matrix D_H, where D_H is a block diagonal matrix of dimension $|G|$ where each block is equal to DFT_H. The goal here will be to assume that we have applied the matrix D_H already and to use that information to our advantage in applying DFT_G.

Recall that the columns of DFT_G correspond to elements of G. If we order the elements based on the right cosets of H, we can factor DFT_G by computing the product

$$F_G = DFT_G D_H^{-1}$$

where

$$D_H = \begin{bmatrix} DFT_H & & & 0 \\ & DFT_H & & \\ & & ... & \\ 0 & & & DFT_H \end{bmatrix}.$$

We emphasize the column ordering used in DFT_G, since each copy of DFT_H corresponds to a Hg_b coset, where g_b is a coset representative for H in G. Thus we can write

$$DFT_G = F_G D_H.$$

This idea can be generalized to a subgroup chain. If $G_0 = 0 \leq G_1 \leq ... \leq G_n = G$, the overall factorization would look like

$$DFT_G = F_G * \begin{bmatrix} F_{G_{n-1}} & & 0 \\ & ... & \\ 0 & & F_{G_{n-1}} \end{bmatrix} \begin{bmatrix} F_{G_{n-2}} & & 0 \\ & ... & \\ 0 & & F_{G_{n-2}} \end{bmatrix} ... \begin{bmatrix} 1 & & 0 \\ & ... & \\ 0 & & 1 \end{bmatrix}.$$

Note that the final identity matrix occurs because the final group in the subgroup chain is the trivial group, whose only irreducible representation is the one dimensional trivial representation.

Example 3.2.1. *We will give the factorization when $G = S_3$ with respect to the natural subgroup chain $S_1 \leq S_2 \leq S_3$. The group element ordering we will use for the columns of DFT_{S_3} will be*

$$1, \quad (12), \quad (13), \quad (132), \quad (23), \quad (123).$$

Note that this is based on the right cosets of S_2 in S_3, using coset representatives 1, (13), *and* (23). *In general we can construct the cosets of S_{n-1} in S_n by using the elements $(1n), (2n), ..., (n-1n)$ as representatives. The DFT for S_3 is*

$$DFT_{S_3} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & -\frac{3}{4} & \frac{3}{4} & \frac{3}{4} & -\frac{3}{4} \\ 0 & 0 & -1 & -1 & 1 & 1 \\ 1 & 1 & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ 1 & -1 & -1 & 1 & -1 & 1 \end{bmatrix}.$$

The DFT for S_2 is

$$DFT_{S_2} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

We can solve for the intermediate F_{S_3} in $DFT_{S_3} = F_{S_3} D_{S_2}$ by calculating $F_{S_3} = DFT_{S_3} D_{S_2}^{-1}$. Thus the factorization becomes

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & -\frac{3}{4} & \frac{3}{4} & \frac{3}{4} & -\frac{3}{4} \\ 0 & 0 & -1 & -1 & 1 & 1 \\ 1 & 1 & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ 1 & -1 & -1 & 1 & -1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 0 & -\frac{3}{4} & 0 & \frac{3}{4} \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 1 & 0 & -\frac{1}{2} & 0 & -\frac{1}{2} & 0 \\ 0 & 1 & 0 & -1 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}.$$

We will now discuss the connection between these matrices and the more general algebraic structure discussed in the previous section. Assume we want to apply the DFT to some function $f \in \mathbb{C}G$ given in the usual basis based on the elements of G. The DFT for G can then be viewed as a change of basis to the basis that we have chosen for the frequency space. The restriction of the decompositions of $\mathbb{C}G$ with respect to some subgroup chain allows us to choose a seminormal basis, using the manner of restriction to build up the necessary partitions that go along with the seminormal basis.

If we build the DFT using the irreducible representations that are adapted to the a given chain of subgroups, the DFT for G can be viewed as a change of basis to the seminormal basis.

Let B be a seminormal basis for $\mathbb{C}G$, and let $H \leq G$. Let $R_1, ..., R_k$ be the irreducible representations for H. We know that when we consider the restriction of $\mathbb{C}G$ as a $\mathbb{C}H$-module, we end up with multiple isomorphic copies of the R_i in the decomposition of $\mathbb{C}G$. Due to the right coset structure, if we have

$$\mathbb{C}H \cong R_1 \oplus ... \oplus R_k$$

then we can decompose $\mathbb{C}G$ restricted to a $\mathbb{C}H$-module as

$$\mathbb{C}G \downarrow_H \cong (R_1 \oplus ... \oplus R_k)^{[G:H]}.$$

For each copy of R_i there exists a partition of B into sets $B_1, ..., B_m$ such that $R_i = \text{span}(B_i)$ for all i. DFT_H acts as a change of basis onto the part of B that spans the R_i. Thus each block in the matrix D_H acts locally as a change in basis onto a copy of that subset of the seminormal basis B. We can view D_H as an intermediate change in basis. The matrix F_G will then change the basis from that intermediate basis to the seminormal basis B. Thus when we look at the full factorization corresponding to some subgroup chain, each of the F_{G_j} blocks functions as an intermediate change of basis, so the factorization becomes a series of change in basis that result in the final desired basis.

This can be seen in Example 1.1 when we factor the DFT for S_3. Each block in the matrix D_{S_2} acts like the change of basis onto the seminormal basis for S_2 on a S_2-dimensional subspace of $\mathbb{C}S_3$. However, with respect to S_3 this certainly is not the seminormal basis for S_3, so the matrix F_{S_3} is used to complete the change of basis.

3.3 Developing an upper bound

We want to develop a bound on the number of operations required to apply the matrix factorization to an arbitrary vector. Note that the efficiency of this algorithm is dependent on the F_{G_i}s, so in order to determine a bound we need to understand their structure and interaction with each other.

We will take a recursive approach to this task. Having done all the necessary work to reach a certain point, how much more work do we need to do to finish the computation? Consider the situation where we have applied D_H to a vector and now wish to determine the additional number of operations required to apply DFT_G, taking advantage of the fact that D_H has already done much of the work for us. In other words, if

$$DFT_G = F_G D_H$$

then we need to know how many arithmetic operations it will take to apply F_G.

We need to develop some notation to help talk about the manner in which the irreducible representatives of G restrict with respect to H. Note that rather then looking at $\mathbb{C}G$ acting on itself as we have been, we will consider a general $\mathbb{C}G$-module V. We will next consider a lengthy example that will give us a extremely useful way in which to discuss the irreducible representations, their restrictions, and how they act at each point in a subgroup chain.

Example 3.3.1. *Let us consider a specific (if somewhat contrived) example that will help illustrate several key points, useful in the proof of our upcoming theorem. Let V be a representation of G and let T_1, T_2, T_3, T_4 be irreducible representations of G with dimensions*

$$d_{T_1} = 1, \quad d_{T_2} = 3, \quad d_{T_3} = 4, \quad d_{T_4} = 2$$

such that

$$V \cong T_1^1 \oplus T_2^3 \oplus T_3^2 \oplus T_4^4.$$

Let H be a subgroup of G with irreducible representations R_1, R_2, and R_3 with dimensions

$$d_{R_1} = 1, \quad d_{R_2} = 2, \quad d_{R_3} = 3.$$

Upon restriction to H, let

$$T_1 \cong R_1, \quad T_2 \cong R_1 \oplus R_2, \quad T_3 \cong R_2 \oplus R_3, \quad T_4 \cong R_3.$$

We consider the two copies of T_3. Let $\alpha \in \mathbb{C}G$. We can write the component of α corresponding to the copies of T_3 as a block diagonal matrix

$$[\alpha]_{T_3} = \begin{bmatrix} * & * & * & * & & & & \\ * & * & * & * & & & & \\ * & * & * & * & & & & \\ * & * & * & * & & & & \\ & & & & * & * & * & * \\ & & & & * & * & * & * \\ & & & & * & * & * & * \\ & & & & * & * & * & * \end{bmatrix}.$$

We note that since the two blocks are isomorphic (as they both correspond to T_3), we can choose a basis $\{b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8\}$ such that the two block are equal. But then we can encode the action of α on some element in T_3 by the action of the basis elements. Since the basis elements are chosen such that the two blocks are equal, we know that they will act the same on the top four entries of a vector as the bottom four entries. That means we could view the action on the top four and the bottom four entries simultaneously. We can encode this action by considering the action of an appropriate matrix on the following object:

$$\begin{bmatrix} \ast & \ast \\ \ast & \ast \\ \ast & \ast \\ \ast & \ast \end{bmatrix}.$$

Each $\star$ can be thought of as corresponding to one of the basis elements b_i. Extending this idea to all the representations of V, we can then write

$$V \cong \begin{bmatrix} \star \end{bmatrix}_{T_1} \oplus \begin{bmatrix} \star & \star & \star \\ \star & \star & \star \\ \star & \star & \star \end{bmatrix}_{T_2} \oplus \begin{bmatrix} \star & \star \\ \star & \star \\ \star & \star \\ \star & \star \end{bmatrix}_{T_3} \oplus \begin{bmatrix} \star & \star & \star & \star \\ \star & \star & \star & \star \end{bmatrix}_{T_4}.$$

We next want to construct an operator that projects onto all of the basis elements of a given type. To consider a specific example, lets say that we want to project onto all of the basis elements that correspond to the first row of T_2. The operator will consist of the direct sums of matrices which act on the matrix encoding that we have just given for V. We can write this operator, $\Delta^{(1)}_{T_2}$ as

$$\Delta^{(1)}_{T_2} = \begin{bmatrix} 0 \end{bmatrix}_{T_1} \oplus \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}_{T_2} \oplus \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}_{T_3} \oplus \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}_{T_4}.$$

Note that this operator is equivalent to a primitive idempotent of V. Now consider the irreducible representation R_2 of H. Note that there are two elements in the basis for R_2; we will encode them with the symbols $\diamond$ and $\bullet$ respectively. Then we can identify all the copies of R_2 in terms of the basis elements $\diamond$ and $\bullet$ as

$$V \cong \begin{bmatrix} \star \end{bmatrix}_{T_1} \oplus \begin{bmatrix} \star\star\star \\ \diamond\diamond\diamond \\ \bullet\bullet\bullet \end{bmatrix}_{T_2} \oplus \begin{bmatrix} \diamond\diamond\diamond \\ \bullet\bullet\bullet \\ \star\star\star \\ \star\star\star \end{bmatrix}_{T_3} \oplus \begin{bmatrix} \diamond\diamond \\ \bullet\bullet \end{bmatrix}_{T_4}.$$

If we wanted to project onto the basis elements of type $\diamond$, we could construct an element $\Delta^{(1)}_{R_2}$:

$$\Delta^{(1)}_{R_2} = \begin{bmatrix}0\end{bmatrix}_{T_1} \oplus \begin{bmatrix}0 & 0 & 0\\ 1 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}_{T_2} \oplus \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}_{T_3} \oplus \begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix}_{T_4}.$$

This element is the sum of three primitive idempotents of V.

We can use this idea to calculate the number of arithmetic operations required to project onto a seminormal basis for G assuming that we have done all the work in a subgroup chain $1 \leq ... \leq H \leq G$ up to H.

Example 3.3.2. *Consider the space that $\Delta^{(1)}_{R_2}$ (from Example 3.3.1) is projecting onto. Assume that we have already computed this projection, and we want to determine the number of arithmetic operations that will be required to separate the $\diamond$ into those corresponding with T_2, T_3, and T_4. Note that the dimension of this space is 8, as it is generated by the number of $\diamond$ basis elements. Then in order to seperate the $\diamond$s corresponding to T_2, T_3, and T_4 we will have to apply an 8×8 matrix which will require at most $8^2 = 64$ operations. The same thought process applies for the $\bullet$-type basis elements, giving us at most $8^2 * d_{R_2} = 8^2 * 2 = 128$ total arithmetic operations. If we repeated this computation for all the T_i, then we would have an upper bound on the number of operations required to project onto the seminormal basis for V.*

We need to make precise the notion of the $\diamond$ and $\bullet$ entries that occur in Example 3.3.1 .

Definition 3.3.1. *Let G be a finite group and let $G_0 \leq G_1 \leq ... \leq G_n = G$ be a subgroup chain. Let V be a $\mathbb{C}G$-module, and let B be a seminormal basis for V. Let $R^{G_i}_1, ..., R^{G_i}_{n_i}$ be the irreducible representations of G_i and let*

$$R^{G_i}_j = span(\{b_0, ..., b_{m_{ij}}\})$$

where $\{b_0, ..., b_{m_{ij}} \subset B$. Note that if we consider V as a $\mathbb{C}G_i$ module, then there will be multiple isomorphic copies of $R^{G_1}_j$. Assume that we have constructed the matrix encoding of V as described in Example 3.3.1, and identified each copy of b_k for all k. Note that we are in effect identifying particular rows of the encoding, as the copies of b_l will always occupy a full row by construction Then let the operator $\Delta^{(b_k)}_{R^{G_i}_j}$ be defined as the direct sum of matrices that act on the objects given in *the encoding that projects onto the space of the entries corresponding to b_k. The explicit construction of $\Delta^{(b_k)}_{R^{G_i}_j}$ is done by placing* 1*s in the appropriate locations on the diagonals and* 0*s everywhere else. $\Delta^{(b_k)}_{R^{G_i}_j}$ will be a sum of primitive idempotents of $\mathbb{C}G$, and as i decreases the number of primitive idempotents in the sum will increase.*

Definition 3.3.2. *Let V be a $\mathbb{C}G$-module and let B be a seminormal basis for V. Consider the subgroup chain $G_0 \leq G_1 \leq ... \leq G_n = G$. Let $b \in B$. Then for each G_i in the chain, by construction there exists an operator $\Delta^{(b_k)}_{R^{G_i}_j}$ that projects onto a space whose span includes b. Let the* ***type*** *of b be the n-tuple whose ith entry is $\Delta^{(b_k)}_{R^{G_i}_j}$.*

We illustrate the definition of type with a specific example.

Example 3.3.3. *Consider the subgroup chain*

$$S_2 \leq S_3 \leq S_4.$$

We have already examined the decomposition of $\mathbb{C}S_4$ with restriction to S_3, so we will not repeat the full details here. We will denote all representations and basis elements directly by their associated Young diagrams and standard Young tableaux. Since the Young lattice gives us an indexing scheme for the representations and basis elements of S_n, when we consider the type of a basis element for S_n we may bijectively replace the Δ-operators with the basis elements they project onto. Let B be the seminormal basis associated with $\mathbb{C}S_4$, and let $b \in B$. We could look at all b with type

$$(\begin{array}{|c|}\hline 1 \\ \hline 2 \\ \hline \end{array}\ ,\ \begin{array}{|c|c|}\hline 1 & 3 \\ \hline 2 & \multicolumn{1}{c}{} \\ \cline{1-1} \end{array}\ ).$$

Note that with respect to $\mathbb{C}S_4$, there can be multiple basis elements with the same type. Consider the basis elements corresponding to the standard Young tableaux

$$\begin{array}{|c|c|}\hline 1 & 3 \\ \hline 2 & 4 \\ \hline \end{array}$$

and

$$\begin{array}{|c|c|}\hline 1 & 3 \\ \hline 2 & \multicolumn{1}{c}{} \\ \cline{1-1} 4 & \multicolumn{1}{c}{} \\ \cline{1-1} \end{array}\ .$$

When we restrict down, we drop the box with a 4 in it, so both of these basis elements look like

$$\begin{array}{|c|c|}\hline 1 & 3 \\ \hline 2 & \multicolumn{1}{c}{} \\ \cline{1-1}\end{array}$$

with restriction to S_3. Thus they both would have type ($\begin{array}{|c|}\hline 1 \\ \hline 2 \\ \hline\end{array}$, $\begin{array}{|c|c|}\hline 1 & 3 \\ \hline 2 & \multicolumn{1}{c}{} \\ \cline{1-1}\end{array}$).

Using these examples we may now present and prove a theorem that gives an upper bound to the number of arithmetic operations required by our algorithm.

Theorem 3.3.1. *Let G be a group and let $H \leq G$. Let the irreducible representations of G be given by $T_1, ..., T_h$ and the irreducible representations of H be given by $R_1, ..., R_k$. Let V be a $\mathbb{C}G$-module. For each R_j, let the number of times R_j appears in a decomposition of any T_i under restriction to a $\mathbb{C}H$-module be denoted Υ_j. Let d_j denote the dimension of R_j. Let $L_V(G)$ be the number of arithmetic operations required to compute the change of basis onto the seminormal basis of V with respect to the subgroup chain $H \leq G$. Then*

$$L_V(G) \leq \sum_{j=1}^{k} (\Upsilon_j)^2 d_j + L_V(H).$$

Proof. We may write

$$V = \oplus_{i=1}^{h} T_i^{n_i}.$$

Let B be a seminormal basis for V. When we restrict V as $\mathbb{C}H$-module, for each T_i in the decomposition of V we may write

$$T_i = R_{c_1} \oplus ... \oplus R_{c_m}$$

where $\{R_{c_1}, ...R_{c_m}\} \subset \{R_1, ..., R_k\}$. Consider a specific irreducible representation for H, R_j with dimension d_j. Then

$$R_j = \text{ span}(\{b_1, b_2, ..., b_{d_j}\})$$

for a set of basis elements $B_j = \{b_1, b_2, ..., b_{d_j}\} \subset B$ by definition of a seminormal basis adapted to the subgroup chain $H \leq G$. Consider $b_k \in B_j$ and let its type be given by $(\Delta_{R_j}^{(b_k)})$. By construction, the operator $\Delta_{R_j}^{(b_k)}$ will project onto all basis elements of that type. There will be one basis element of type $(\Delta_{R_j}^{(b_k)})$ for each copy of R_i in the decomposition of V restricted to H, so there will be Υ_j total such basis elements. Then consider the space that

$\Delta_{R_j}^{(b_k)}$ projects onto. Note that we are assuming we have already computed this projection - the number of operations required to do so is contained in the term $L_V(H)$. We must now take this space and project it such that we can separate the basis elements of type $((\Delta_{R_j}^{(b_k)})$ into those that appear in the decomposition of T_1, T_2, and so on in order to compute the change of basis onto the seminormal basis of V. At worst, the number of arithmetic operations is bounded by the dimension of the space (since we will have an $\Upsilon_j \times \Upsilon_j$ matrix), so we have no more then Υ_j^2 arithmetic operations for this step of the calculation. Repeating this step for all $b_k \in B_j$ yields the same number of arithmetic operations for each, so relative to the representation R_j we need $\Upsilon_j^2 d_j$ arithmetic operations. We then sum over all R_j; adding the term $L_V(H)$ yields an upper bound on the number of arithmetic operations as

$$L_V(G) \leq \sum_{j=1}^{k} (\Upsilon_j)^2 d_j + L_V(H).$$

□

Since the *DFT* is a change of basis onto the seminormal basis, we have determined a bound on the number of operations required to go up a step in the factoring chain in our algorithm. In the case where $V = \mathbb{C}G$, acting on itself as a $\mathbb{C}G$-module, we get a much nicer statement.

Corollary 3.3.1. *Let G be a group and let $H \leq G$. Let the irreducible representations of H be given by $T_1, ..., T_k$, and let their respective dimensions be given by $d_1, ..., d_k$. Let $L_{\mathbb{C}G}(G)$ be the number of operations required to compute DFT_G with respect to $\mathbb{C}G$. Then*

$$L_{\mathbb{C}G}(G) \leq \sum_{i=1}^{k} ([G:H]d_i)^2 d_i + L_{\mathbb{C}G}(H).$$

Proof. Note that since we have $\mathbb{C}G$ acting on itself, we can induce a right coset structure with respect to H. Let $R_1, ..., R_k$ be the irreducible representations of H with respective dimensions $d_1, ..., d_k$. Thus when we consider the decomposition of $\mathbb{C}G$ with restriction to H, we end up with

$$\mathbb{C}G \cong (R_1^{d_1} \oplus R_2^{d_2} ... \oplus R_k^{d_k})^{[G:H]}.$$

Therefore if we consider any individual representation R_i, we know that its multiplicity in $\mathbb{C}G$ is $\Upsilon_i = [G:H]d_i$. Thus by Theorem 3.1, we have

$$L_{\mathbb{C}G}(G) \leq \sum_{i=1}^{k} ([G:H]d_i)^2 d_i + L_{\mathbb{C}G}(H).$$

□

Thus we now have a theoretical bound on the number of operations our algorithm requires at each step of the subgroup chain.

Example 3.3.4. *Consider the DFT for S_3. The key 'factor' that determines how fast we can apply the DFT will be the number of non-zero terms in the factorization compared with the number of non-zero terms in the original DFT. Strictly in terms of zero and non zero terms, we can write the factorization as*

$$\begin{bmatrix} \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ & & \bullet & \bullet & \bullet & \bullet \\ & & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \end{bmatrix} = \begin{bmatrix} \bullet & & \bullet & & \bullet & \\ & \bullet & & \bullet & & \bullet \\ & & & \bullet & & \bullet \\ & & \bullet & & \bullet & \\ \bullet & & \bullet & & \bullet & \\ & \bullet & & \bullet & & \bullet \end{bmatrix} \begin{bmatrix} \bullet & \bullet & & & & \\ \bullet & \bullet & & & & \\ & & \bullet & \bullet & & \\ & & \bullet & \bullet & & \\ & & & & \bullet & \bullet \\ & & & & \bullet & \bullet \end{bmatrix}.$$

The number of non-zero entries in DFT_{S_3} is 32. The number of non-zero entries in the factorization is 28. This does not seem like much of an improvement; however we will see in the next section that as n increases we save a great deal of time over directly computing the DFT.

3.4 Analyzing the Running Time

We will now analyze the bound given by Corollary 3.3.1. We will mostly look at the case where $G = S_n$ but some of the results apply to a general group. Our goal is come up with a reasonable assertion as to an upper bound of the total number of operations required by the algorithm.

Lemma 3.4.1. *Let G be a finite group and let $d_1, ..., d_k$ denote the dimensions of the distinct irreducible representations of G. Then*

$$\sum_{i=1}^{k} d_i^3 \leq |G|^{3/2}.$$

Proof. Let $D = \max(\{d_1, ..., d_k\})$. We know that $D \leq |G|^{1/2}$. It follows that

$$\sum_{i=1}^{k} d_i^3 \leq \sum_{i=1}^{k} d_i^2 D.$$

But since $\sum_{i=1}^{k} d_i^2 = |G|$, we have

$$\sum_{i=1}^{k} d_i^2 D \leq |G|^{1/2}|G| = |G|^{3/2}$$

and the lemma follows. □

If we apply this lemma to the result from Theorem 3.3.1, we get that

$$L(G) \leq \sum_{i=1}^{k} ([G:H]d_i)^2 d_i + [G:H]L(H) = [G:H]^2 \sum_{i=1}^{k} d_i^3 + [G:H]L(H)$$

$$= \frac{|G|^2}{|H|^2} \sum_{i=1}^{k} d_i^3 + [G:H]L(H) \leq \frac{|G|^2}{|H|^2}|H|^{3/2} + [G:H]L(H)$$

$$= |G|[G:H]|H|^{1/2} + [G:H]L(H). \qquad (1)$$

This is a more convenient form to handle. We consider a couple examples.

Example 3.4.1. *Let $G = S_n$ and $H = S_{n-1}$. Then applying (1) we get that*

$$L(S_n) \leq |S_n|[S_n : S_{n-1}]|S_{n-1}|^{1/2} + [S_n : S_{n-1}]L(S_{n-1})$$

$$= n^2((n-1)!^{3/2}) + nL(S_{n-1}).$$

Example 3.4.2. *Let G be an abelian group with $H \leq G$. Note that for an abelian group, all the irreducible representations are of dimension one. Thus we have*

$$\sum_{i=1}^{k} d_i^3 = \sum_{i=1}^{k} 1^3 = \sum_{i=1}^{k} 1^2 = \sum_{i=1}^{k} d_i^2 = |G|.$$

Then

$$L(G) \leq [G:H]^2 \sum_{i=1}^{k} d_i^3 + [G:H]L(H)$$

$$= \frac{|G|^2}{|H|^2}|H| + [G:H]L(H) = |G|[G:H] + [G:H]L(H)$$

$$= [G:H](|G| + L(H)).$$

We now give a form of the result shown in Corollary 3.3.1 that will be slightly easier to use when calculating $L(G)$. First we introduce a little notation. For any group G, if the dimensions of the irreducible representations of G are given by $d_1, ..., d_k$ then let $d^3(G) = \sum_{i=1}^{k} d_i^3$. Given a subgroup chain $0 = G_0 \leq ... \leq G_n = G$, let $\rho_n = [G_n : G_{n-1}]$. Then given groups $G_{n-1} \leq G_n$ with $d_1, ..., d_k$ being the dimensions of the irreducible representations of G_{n-1}, we have that

$$L(G_n) \leq \sum_{i=1}^{k} (\rho_n d_i)^2 d_i + \rho_n L(G_{n-1}) = \rho_n^2 d^3(G_{n-1}) + \rho_n L(G_{n-1})$$

$$= \rho_n (L(G_{n-1}) + \rho_n d^3(G_{n-1})).$$

Let $B(G_n)$ be the upper bound given in the above expression, so

$$B(G_n) = \rho_n (L(G_{n-1}) + \rho_n d^3(G_{n-1})).$$

We use this expression to examine $B(S_n)$ for some small values of n. By analyzing those results we will be able to come up with an upper bound for the running time of the algorithm for those values of n.

We first note that the FFT on S_n will require at minimum $n!$ operations. For example, we know that the row in the DFT corresponding to the identity representation consists of all $1'$s; therefore the application of the DFT will require at minimum one addition and one multiplication per 1. Since there are $n!$ 1s, we have at least $n!$ total operations. However, our total operation count will not be just $n!$, but rather $q(n)n!$ for some function q. Since the real distinction between the running times of different algorithms comes from the function q, it becomes easier to compare the operation counts if we normalize all our data by $n!$. We can then fit the normalized data to a known function to create an upper bound on the operation count. The values $B(S_n)/n!$ for small n are shown in Table 3.1. This is a significant improvement over directly computing the DFT for S_n, which would require $n!^2$ operations, normalized by $n!$ to simply $n!$, also shown in Table 3.1.

We plot the data from Table 3.1 in Figure 3.1. Due to the growth of $n!$, graphing our data against $n!$ is impractical so we will simply leave that information in tabular format.

The plot suggests an exponential growth. This is compared to e^n in Figure 3.2.

Note that the lines connecting the points for $B(S_n)/n!$ in Figure 3.2 are simply straight edges joining the points and are not meant to imply a function fitted to $B(S_n)/n!$. Clearly $B(S_n)/n!$ grows noticeably slower then e^n

n	$B(S_n)/n!$	$n!$
1	1	1
2	2	2
3	6	6
4	11.67	24
5	25	120
6	54.80	720
7	133.12	5040
8	340.16	40320
9	906	362880
10	2554	3628800

Table 3.1: Values of $B(S_n)/n!$ for $n \leq 10$.

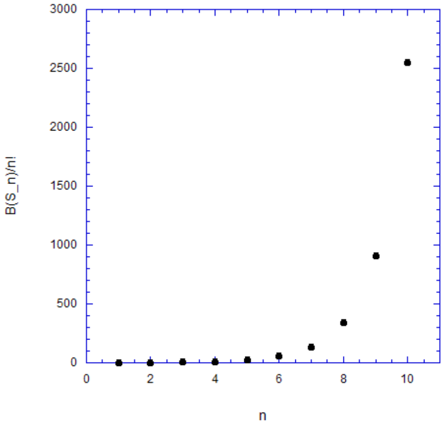


Figure 3.1: Graph of $B(S_n)/n!$ for $n \leq 10$

for $n \leq 10$. By having the program Kaleidagraph perform a curve fit on our data, we see that the curve given by

$$f(x) = .36e^{.86n}$$

fits our data very closely, as demonstrated in Figure 3.2. This suggests that

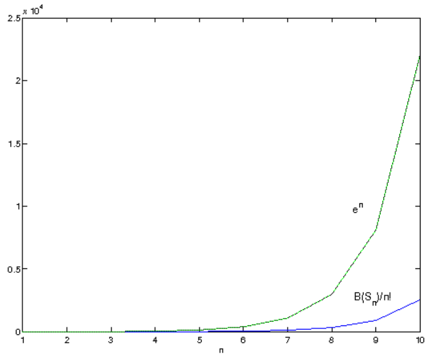


Figure 3.2: Graph of $B(S_n)/n!$ compared with e^n for $n \leq 10$

the operation count for S_n, $n \leq 10$ can be approximated by $.36e^{.86n}n!$.

We compare our results to two known bounds. In [2], Clausen and Baum present an algorithm where the upper bound on the operation count is $.5(n^3 + n^2)n!$. We plot our results against that bound in Figure 3.4.

In his paper *The Efficient Computation of Fourier Transforms on the Symmetric Group* [6], David Maslen presents an algorithm with the following bound:

$$L(S_n) \leq 1.5(n-1)n(n!).$$

We plot the upper bound for our algorithm against Maslen's upper bound for $n \leq 10$ in Figure 3.5. Notice that for $n \leq 6$, the algorithms have nearly an identical bound. However, for $7 \leq n \leq 10$ the exponential growth rate of our algorithm catches up and the operation count is significantly worse for our algorithm compared to Maslen's.

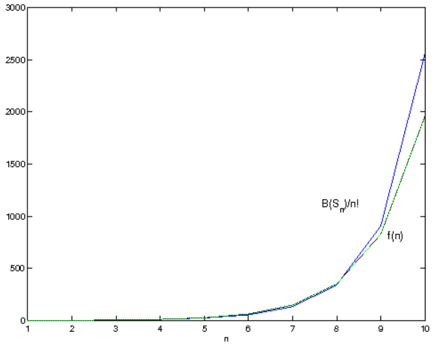


Figure 3.3: Graph of $B(S_n)/n!$ compared with $f(n)$ for $n \leq 10$

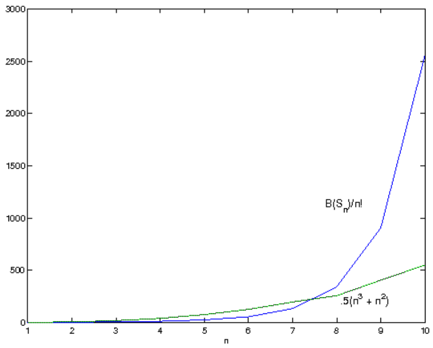


Figure 3.4: Graph of $B(S_n)/n!$ compared with $.5(n^3 + n^2)$

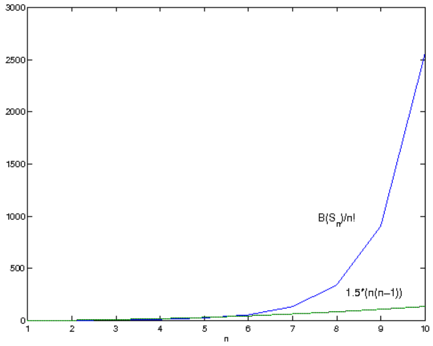


Figure 3.5: Graph of $B(S_n)/n!$ compared with $1.5(n-1)n$

Chapter 4

Conclusions and Future Work

One of the advantages to the algorithm that we have described in Chapter 3 is that, despite the deep theoretical mechanisms behind it, it seems to be easily implementable. Much of the work is simply the precomputation of the DFT matrices, for which we have a fairly straightforward method of construction. After that all that is required is the computation of a series of matrix vector products. Another advantage of the algorithm is that, despite the fact that several other published algorithms outperform it for $n \geq 7$, it maintains essentially the same efficiency for small values of n. For applications that do not require the use of the larger symmetric groups the algorithm is of practical value.

There are several directions that future research in this area could take. One is to study the structures of the matrices produced by the factorization and see if there is a way to further reduce the bound of operations we gave in the Theorem 3.3.1. By simply counting the non-zero entries in the factorizations for S_4 and S_3, it appears that the upper bound given in theorem can be improved upon.

Another direction is to consider the use of two-sided cosets rather than the single-sided cosets that were used in this discussion. The major constraint in terms of having to use additional arithmetic operations in our algorithm is that we are unable to project onto less then every basis element of a given type, so the spaces that we project onto are quite large. However, by using two-sided cosets it should be possible to construct elements that not only project onto every basis element of a given type but that distinguish between the various elements of the same type. This would significantly reduce the operation count. Previous work done by several students of Professor Michael Orrison of Harvey Mudd College suggests the possibility that this type of algorithm could have order $O(n^2 n!)$ operations, which is the same order of efficiency that Maslen's algorithm, the current fastest known for the symmetric group, achieves [6].

Appendix A

Calculation of Representations and Factors for S_4

A.1 Factorization of the DFT of S_4

Given here is the factorization of DFT_{S_4} in terms of the non-zero entries of the matrices. DFT_{S_4} has 464 non-zero entries. The combined number of non-zero entries of the three factors in the factorization is 257.

$$D_{S_4} =$$

$$\begin{bmatrix}
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & & & \bullet & \bullet & \bullet & & \bullet & \bullet & & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
& & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & & & \bullet & \bullet & \bullet & \bullet \\
& & & & & & \bullet & \bullet & & \bullet & \bullet & & \bullet & \bullet & \bullet & & & \bullet & & \bullet & \bullet & \bullet & \bullet & \\
& & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & & \bullet & \bullet & \bullet & \bullet & \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
& & & & & & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
& & & & & & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & & & & & & \\
& & & & & & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
& & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & & & \bullet & \bullet & \bullet & & \bullet & \bullet & & & & \bullet & \bullet & \bullet & \bullet \\
& & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & & & \bullet & \bullet & \bullet & & \bullet & \bullet & & & & \bullet & \bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
& & & & & & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
& & & & & & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & & & & & & \\
& & & & & & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
& & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & & & \bullet & \bullet & \bullet & \bullet \\
& & & & & & \bullet & \bullet & & \bullet & \bullet & & \bullet & \bullet & \bullet & & & \bullet & & & \bullet & \bullet & \bullet & \bullet \\
& & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & & & \bullet & \bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & & & \bullet & \bullet & \bullet & & \bullet & \bullet & & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet
\end{bmatrix}$$

$$
= \begin{bmatrix}
\bullet &&&&&& \bullet &&&&&& \bullet &&&&&& \bullet &&&&& \\
& \bullet &&&&&& \bullet & \bullet &&&&& \bullet & \bullet &&&&& \bullet &&&& \\
&& \bullet &&&&& \bullet & \bullet &&&&& \bullet & \bullet &&&&&& \bullet &&& \\
&&&&&&& \bullet & \bullet &&&&& \bullet\bullet &&&&&& \bullet &&&& \\
&&& \bullet &&&&&& \bullet & \bullet &&&&& \bullet & \bullet &&&&& \bullet && \\
&&&& \bullet &&&&& \bullet & \bullet &&&&& \bullet & \bullet &&&&&& \bullet & \\
&&&&&&&&& \bullet & \bullet &&&&& \bullet & \bullet &&&&&& \bullet & \\
&&&&& \bullet &&&&&& \bullet &&&&&&&&&&&& \\
&&&&&& \bullet &&&&&& \bullet &&&&&& \bullet &&&&& \\
\bullet &&&&&& \bullet &&&&&& \bullet &&&&&& \bullet &&&&& \\
&& \bullet &&&&& \bullet & \bullet &&&&& \bullet & \bullet &&&&&& \bullet &&& \\
&&& \bullet &&&&&& \bullet & \bullet &&&&& \bullet & \bullet &&&&& \bullet && \\
&&&& \bullet && \bullet &&& \bullet & \bullet &&&&& \bullet & \bullet &&&&&& \bullet & \\
& \bullet && \bullet && \bullet && \bullet && \bullet && \bullet && \bullet && \bullet && \bullet && \bullet && \bullet && \bullet \\
&&&&&& \bullet && \bullet && \bullet && \bullet && \bullet && \bullet && \bullet && \bullet && \bullet & \\
&&&&&& \bullet && \bullet && \bullet && \bullet && \bullet && \bullet &&&&&&& \\
&&&&&& \bullet & \bullet &&&&& \bullet & \bullet &&&&& \bullet &&&&& \\
&&&&&& \bullet & \bullet &&&&& \bullet & \bullet &&&&& \bullet &&&&& \\
&&&&&& \bullet & \bullet &&&&& \bullet & \bullet &&&&&& \bullet &&&& \\
&&&&&&&& \bullet & \bullet &&&&& \bullet & \bullet &&&&& \bullet &&& \\
&&& \bullet &&&&& \bullet & \bullet &&&&& \bullet & \bullet &&&&& \bullet &&& \\
&&&& \bullet &&&& \bullet & \bullet &&&&& \bullet & \bullet &&&&& \bullet &&& \\
& \bullet && \bullet && \bullet && \bullet && \bullet && \bullet && \bullet && \bullet && \bullet && \bullet && \bullet && \bullet
\end{bmatrix}
$$

$$
\begin{bmatrix}
\bullet && \bullet && \bullet &&&&&&&&&&&&&&&&&&& \\
& \bullet && \bullet && \bullet &&&&&&&&&&&&&&&&&& \\
&&& \bullet && \bullet &&&&&&&&&&&&&&&&&& \\
&& \bullet && \bullet &&&&&&&&&&&&&&&&&&& \\
\bullet && \bullet && \bullet &&&&&&&&&&&&&&&&&&& \\
& \bullet && \bullet && \bullet &&&&&&&&&&&&&&&&&& \\
&&&&&& \bullet && \bullet && \bullet &&&&&&&&&&&&& \\
&&&&&&& \bullet && \bullet && \bullet &&&&&&&&&&&& \\
&&&&&&&&& \bullet && \bullet &&&&&&&&&&&& \\
&&&&&&&& \bullet && \bullet &&&&&&&&&&&&& \\
&&&&&& \bullet && \bullet && \bullet &&&&&&&&&&&&& \\
&&&&&&& \bullet && \bullet && \bullet &&&&&&&&&&&& \\
&&&&&&&&&&&& \bullet && \bullet && \bullet &&&&&&& \\
&&&&&&&&&&&&& \bullet && \bullet && \bullet &&&&&& \\
&&&&&&&&&&&&&&& \bullet && \bullet &&&&&& \\
&&&&&&&&&&&&&& \bullet && \bullet &&&&&&& \\
&&&&&&&&&&&& \bullet && \bullet && \bullet &&&&&&& \\
&&&&&&&&&&&&& \bullet && \bullet && \bullet &&&&&& \\
&&&&&&&&&&&&&&&&&& \bullet && \bullet && \bullet & \\
&&&&&&&&&&&&&&&&&&& \bullet && \bullet && \bullet \\
&&&&&&&&&&&&&&&&&&&&& \bullet && \bullet \\
&&&&&&&&&&&&&&&&&&&& \bullet && \bullet & \\
&&&&&&&&&&&&&&&&&& \bullet && \bullet && \bullet & \\
&&&&&&&&&&&&&&&&&&& \bullet && \bullet && \bullet
\end{bmatrix}
$$

$$
\begin{bmatrix}
\bullet & \bullet &&&&&&&&&&&&&&&&&&&&&& \\
\bullet & \bullet &&&&&&&&&&&&&&&&&&&&&& \\
&& \bullet & \bullet &&&&&&&&&&&&&&&&&&&& \\
&& \bullet & \bullet &&&&&&&&&&&&&&&&&&&& \\
&&&& \bullet & \bullet &&&&&&&&&&&&&&&&&& \\
&&&& \bullet & \bullet &&&&&&&&&&&&&&&&&& \\
&&&&&& \bullet & \bullet &&&&&&&&&&&&&&&& \\
&&&&&& \bullet & \bullet &&&&&&&&&&&&&&&& \\
&&&&&&&& \bullet & \bullet &&&&&&&&&&&&&& \\
&&&&&&&& \bullet & \bullet &&&&&&&&&&&&&& \\
&&&&&&&&&& \bullet & \bullet &&&&&&&&&&&& \\
&&&&&&&&&& \bullet & \bullet &&&&&&&&&&&& \\
&&&&&&&&&&&& \bullet & \bullet &&&&&&&&&& \\
&&&&&&&&&&&& \bullet & \bullet &&&&&&&&&& \\
&&&&&&&&&&&&&& \bullet & \bullet &&&&&&&& \\
&&&&&&&&&&&&&& \bullet & \bullet &&&&&&&& \\
&&&&&&&&&&&&&&&& \bullet & \bullet &&&&&& \\
&&&&&&&&&&&&&&&& \bullet & \bullet &&&&&& \\
&&&&&&&&&&&&&&&&&& \bullet & \bullet &&&& \\
&&&&&&&&&&&&&&&&&& \bullet & \bullet &&&& \\
&&&&&&&&&&&&&&&&&&&& \bullet & \bullet && \\
&&&&&&&&&&&&&&&&&&&& \bullet & \bullet && \\
&&&&&&&&&&&&&&&&&&&&&& \bullet & \bullet \\
&&&&&&&&&&&&&&&&&&&&&& \bullet & \bullet
\end{bmatrix} .
$$

A.2 Seminormal Representations for S_4

Tables 1 and 2 give the seminormal representations for all of the elements of S_4. The irreducible representations of S_4 correspond to the Young diagrams for $n = 4$ which are listed as follows:

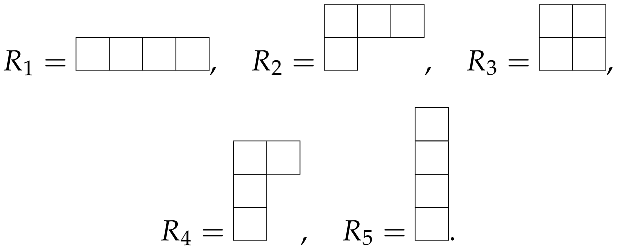


Table A.1: Seminormal representations for elements of S_4

$s \in S_4$	$D_{R_1}(s)$	$D_{R_2}(s)$	$D_{R_3}(s)$	$D_{R_4}(s)$	$D_{R_5}(s)$
1	$[1]$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$[1]$
(12)	$[1]$	$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$[-1]$
(13)	$[1]$	$\begin{bmatrix} 1/2 & -3/4 & 0 \\ -1 & -1/2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1/2 & -3/4 \\ -1 & -1/2 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1/2 & -3/4 \\ 0 & -1 & -1/2 \end{bmatrix}$	$[-1]$
(132)	$[1]$	$\begin{bmatrix} -1/2 & -3/4 & 0 \\ 0 & 1 & -1/2 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1/2 & -3/4 \\ 1 & -1/2 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1/2 & -3/4 \\ 0 & 1 & -1/2 \end{bmatrix}$	$[1]$
1	$[1]$	$\begin{bmatrix} 1/2 & 3/4 & 0 \\ 1 & -1/2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1/2 & 3/4 \\ 1 & -1/2 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1/2 & 3/4 \\ 0 & 1 & -1/2 \end{bmatrix}$	$[-1]$
(123)	$[1]$	$\begin{bmatrix} -1/2 & 3/4 & 0 \\ -1 & -1/2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1/2 & 3/4 \\ -1 & -1/2 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1/2 & 3/4 \\ 0 & -1 & -1/2 \end{bmatrix}$	$[1]$
(14)	$[1]$	$\begin{bmatrix} 1/2 & -1/4 & -2/3 \\ -1/3 & 5/6 & -4/9 \\ -1 & -1/2 & -1/3 \end{bmatrix}$	$\begin{bmatrix} -1/2 & -3/4 \\ -1 & 1/2 \end{bmatrix}$	$\begin{bmatrix} 1/3 & -4/9 & 2/3 \\ -1/2 & -5/6 & -1/4 \\ 1 & -1/3 & -1/2 \end{bmatrix}$	$[-1]$
(124)	$[1]$	$\begin{bmatrix} -1/2 & 1/4 & 2/3 \\ -1/3 & 5/6 & -4/9 \\ -1 & -1/2 & -1/3 \end{bmatrix}$	$\begin{bmatrix} 1/2 & 3/4 \\ -1 & 1/2 \end{bmatrix}$	$\begin{bmatrix} -1/3 & 4/9 & -2/3 \\ 1/2 & 5/6 & 1/4 \\ 1 & -1/3 & -1/2 \end{bmatrix}$	$[1]$
(134)	$[1]$	$\begin{bmatrix} 1/2 & -3/4 & 0 \\ -1/3 & -1/6 & 8/9 \\ -1 & -1/2 & -1/3 \end{bmatrix}$	$\begin{bmatrix} 1/2 & -3/4 \\ 1 & 1/2 \end{bmatrix}$	$\begin{bmatrix} -1/3 & 4/9 & -2/3 \\ -1 & -1/6 & 1/4 \\ 0 & 1 & 1/2 \end{bmatrix}$	$[1]$
(1324)	$[1]$	$\begin{bmatrix} 0 & -1/2 & 2/3 \\ 2/3 & -2/3 & -4/90 \\ -1 & -1/2 & -1/3 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1/3 & -4/9 & 2/3 \\ -1/2 & 2/3 & 1/2 \\ -1 & -2/3 & 0 \end{bmatrix}$	$[-1]$

Table A.2: Seminormal representations for elements of S_4

$s \in S_4$	$D_{R_1}(s)$	$D_{R_2}(s)$	$D_{R_3}(s)$	$D_{R_4}(s)$	$D_{R_5}(s)$
(14)(23)	$[1]$	$\begin{bmatrix} 0 & 1/2 & -2/3 \\ 2/3 & -2/3 & -4/9 \\ -1 & -1/2 & -1/3 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1/3 & 4/9 & -2/3 \\ 1/2 & -2/3 & -1/2 \\ -1 & -2/3 & 0 \end{bmatrix}$	$[1]$
(1234)	$[1]$	$\begin{bmatrix} -1/2 & 3/4 & 0 \\ -1/3 & -1/6 & 8/9 \\ -1 & -1/2 & -1/3 \end{bmatrix}$	$\begin{bmatrix} -1/2 & 3/4 \\ 1 & 1/2 \end{bmatrix}$	$\begin{bmatrix} 1/3 & -4/9 & 2/3 \\ 1 & 1/6 & -1/4 \\ 0 & 1 & 1/2 \end{bmatrix}$	$[-1]$
(24)	$[1]$	$\begin{bmatrix} 1/2 & 1/4 & 2/3 \\ 1/3 & 5/6 & -4/9 \\ 1 & -1/2 & -1/3 \end{bmatrix}$	$\begin{bmatrix} -1/2 & 3/4 \\ 1 & 1/2 \end{bmatrix}$	$\begin{bmatrix} 1/3 & -4/9 & -2/3 \\ -1/2 & -5/6 & 1/4 \\ -1 & 1/3 & -1/2 \end{bmatrix}$	$[-1]$
(142)	$[1]$	$\begin{bmatrix} -1/2 & -1/4 & -2/3 \\ 1/3 & 5/6 & -4/9 \\ 1 & -1/2 & -1/3 \end{bmatrix}$	$\begin{bmatrix} 1/2 & -3/4 \\ 1 & 1/2 \end{bmatrix}$	$\begin{bmatrix} -1/3 & 4/9 & 2/3 \\ 1/2 & 5/6 & -1/4 \\ -1 & 1/3 & -1/2 \end{bmatrix}$	$[1]$
(13)(24)	$[1]$	$\begin{bmatrix} 0 & -1/2 & 2/3 \\ -2/3 & -2/3 & -4/9 \\ 1 & -1/2 & -1/3 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$	$\begin{bmatrix} -1/3 & 4/9 & 2/3 \\ 1/2 & -2/3 & 1/2 \\ 1 & 2/3 & 0 \end{bmatrix}$	$[1]$
(1342)	$[1]$	$\begin{bmatrix} -1/2 & -3/4 & 0 \\ 1/3 & -1/6 & 8/9 \\ 1 & -1/2 & -1/3 \end{bmatrix}$	$\begin{bmatrix} -1/2 & -3/4 \\ -1 & 1/2 \end{bmatrix}$	$\begin{bmatrix} 1/3 & 4/9 & -2/3 \\ 1 & 1/6 & 1/4 \\ 0 & -1 & 1/2 \end{bmatrix}$	$[-1]$
(243)	$[1]$	$\begin{bmatrix} 1/2 & 3/4 & 0 \\ 1/3 & -1/6 & 8/9 \\ 1 & -1/2 & -1/3 \end{bmatrix}$	$\begin{bmatrix} 1/2 & 3/4 \\ -1 & 1/2 \end{bmatrix}$	$\begin{bmatrix} -1/3 & 4/9 & 2/3 \\ -1 & -1/6 & -1/4 \\ 0 & -1 & 1/2 \end{bmatrix}$	$[1]$
(1423)	$[1]$	$\begin{bmatrix} 0 & 1/2 & -2/3 \\ -2/3 & -2/3 & -4/9 \\ 1 & -1/2 & -1/3 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 1/3 & -4/9 & -2/3 \\ -1/2 & 2/3 & -1/2 \\ 1 & 2/3 & 0 \end{bmatrix}$	$[-1]$
(34)	$[1]$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1/3 & 8/9 \\ 0 & 1 & -1/3 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 1/3 & 8/9 & 0 \\ 1 & -1/3 & 0 \\ 0 & 0 & -1 \end{bmatrix}$	$[-1]$
(12)(34)	$[1]$	$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1/3 & 8/9 \\ 0 & 1 & -1/3 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$	$\begin{bmatrix} -1/3 & -8/9 & 0 \\ 0 & -1 & 1/3 \\ 0 & 0 & -1 \end{bmatrix}$	$[1]$
(143)	$[1]$	$\begin{bmatrix} 1/2 & -1/4 & -2/3 \\ -1 & -1/6 & -4/9 \\ 0 & 1 & -1/3 \end{bmatrix}$	$\begin{bmatrix} 1/2 & 3/4 \\ -1 & 1/2 \end{bmatrix}$	$\begin{bmatrix} -1/3 & 8/9 & 0 \\ 1/2 & -1/6 & 3/4 \\ -1 & 1/3 & 1/2 \end{bmatrix}$	$[1]$
(1432)	$[1]$	$\begin{bmatrix} -1/2 & -1/4 & -2/3 \\ 1 & -1/6 & -4/9 \\ 0 & 1 & -1/3 \end{bmatrix}$	$\begin{bmatrix} -1/2 & 3/4 \\ 1 & 1/2 \end{bmatrix}$	$\begin{bmatrix} 1/3 & 8/9 & 0 \\ -1/2 & 1/6 & 3/4 \\ 1 & -1/3 & 1/2 \end{bmatrix}$	$[-1]$
(243)	$[1]$	$\begin{bmatrix} 1/2 & 1/4 & 2/3 \\ 1 & -1/6 & -4/9 \\ 0 & 1 & -1/3 \end{bmatrix}$	$\begin{bmatrix} 1/2 & -3/4 \\ 1 & 1/2 \end{bmatrix}$	$\begin{bmatrix} -1/3 & -8/9 & 0 \\ 1/2 & -1/6 & -3/4 \\ 1 & -1/3 & 1/2 \end{bmatrix}$	$[1]$
(1243)	$[1]$	$\begin{bmatrix} -1/2 & 1/4 & 2/3 \\ -1 & -1/6 & -4/9 \\ 0 & 1 & -1/3 \end{bmatrix}$	$\begin{bmatrix} -1/2 & -3/4 \\ -1 & 1/2 \end{bmatrix}$	$\begin{bmatrix} 1/3 & 8/9 & 0 \\ -1/2 & 1/6 & -3/4 \\ -1 & 1/3 & 1/2 \end{bmatrix}$	$[-1]$

Bibliography

[1] Clausen, M. and U. Baum, Fast Fourier Transforms, BI-Wissenschaftsverlag, Manheim, Germany, 1993.

[2] Clausen, M. and U. Baum, Fast Fourier Transforms for Symmetric Groups: Theory and Implementation, Mathematics of Computation, Volumne 61, Number 204, October, 1993.

[3] Dummit, D. and R. Foote, Abstract Algebra, John Wiley and Sons, New York, 2004.

[4] Gentleman, W. M. and G. Sande, Fast Fourier Transforms - For Fun and Profit, Proceedings - Fall Joint Computer Conference, 1966.

[5] James, G. and A. Kerber, The Representation Theory of the Symmetric Group, Addison-Wesley Publishing Company, 1981.

[6] Maslen, D., The Efficient Computation of Fourier Transforms on the Symmetric Group, Mathematics of Computation, vol. 76 no. 223, July 1998.

[7] Ram, A., Seminormal Representations of Weyl Groups and Iwahori-Hecke Algebras, Proceedings of the London Mathematics Society, 1991.

[8] Rockmore, D., Recent Progress and Applications in Group FFTs, NATO Advanced Study Institute on Computational Noncommutative Algebra and Applications, 2003.

[9] Rockmore, D., The FFT - An Algorithm the whole family can uses, Computing in Science & Engineering, **2** 2000.

[10] Rockmore, D. and D. Maslen, The Cooley-Tukey FFT and Group Theory, Notices of the AMS, 2001.

[11] Rockmore, D. and D. Maslen, Generalized FFTs – A survey of some recent results, Proceedings of the DIMACS Workshop on Groups and Computation, June 7-10, 1995 eds. L. Finklestein and W.Kantor, 1997.

[12] Rockmore, D., Recent Progress and Applications in Group FFTs, NATO Advanced Study Institute on Computational Noncommutative Algebra and Applications, 2003.

[13] Sagan , B., The Symmetric Group, Springer, 2001.