THE COMPUTATION OF FOURIER TRANSFORMS ON THE SYMMETRIC GROUP

DAVID K. MASLEN

Abstract. This paper introduces new techniques for the efficient computation of Fourier transforms on symmetric groups and their homogeneous spaces. We replace the matrix multiplications in Clausen's algorithm with sums indexed by combinatorial objects that generalize Young tableaux, and write the result in a form similar to Horner's rule. The algorithm we obtain computes the Fourier transform of a function on S_n in no more than $\frac{3}{4}n(n-1)|S_n|$ multiplications and the same number of additions. Analysis of our algorithm leads to several enumerative combinatorial problems that generalize path counting. We describe corresponding results for inverse transforms and transforms on homogeneous spaces.

1. Introduction

The harmonic analysis of a complex function on a finite cyclic group is the expansion of that function in a basis of complex exponential functions. This is equivalent to the discrete Fourier transform of a finite data sequence, and may be computed efficiently using the fast Fourier transform algorithms of Cooley and Tukey [6] or their many variants (see e.g. [11]).

In the current paper we study the harmonic analysis of a function on the symmetric group. The analogs of the complex exponentials are the matrix entries of a complete set of irreducible complex matrix representations of S_n, called matrix coefficients, and the expansion of functions in this basis may be computed by a generalized Fourier transform on the symmetric group. We describe efficient algorithms for computing the harmonic analysis of a function on the symmetric group, or equivalently, its generalized Fourier transform. Thus our results may be considered a generalization of the fast Fourier transform to the symmetric group.

Fourier transforms on finite groups have been studied by many authors. The books of Beth [1], Clausen and Baum [2], and the survey article [18] are general references for the computational aspects of these transforms. Rockmore [21] and Diaconis [8] contain discussions of the applications. For applications more specific to symmetric groups, see [7] and [10].

The computation of Fourier transforms on symmetric groups was first studied by Clausen [4] [5], and Diaconis and Rockmore [9], using approaches related to the one taken in the current paper; also see [3] for a detailed discussion of Clausen's algorithm and its implementation. Linton, Michler, and Olsson [14] use a different method that involves the decomposition of Fourier transforms taken at a monomial representations.

The algorithms we develop in the current paper are refinements of Clausen's algorithm [4] for computing Fourier transforms on the symmetric group, and of algorithms developed

Date: 9 April 1997.

1991 *Mathematics Subject Classification.* 20C30, 20C40; Secondary 65T20, 05E10.

The full version of this extended abstract appears as the IHES preprint IHES/M/96/52, and will appear in Math. Comp.

to compute Fourier transforms on compact Lie groups [16]. To describe our main results, let f be a complex function on S_n and let ρ be an irreducible matrix representation of S_n given in Young's orthogonal form (see [13] for terminology). Then the **Fourier transform of f at ρ** is the matrix sum

$$\hat{f}(\rho) = \sum_{s \in S_n} f(s)\rho(s). \tag{1.1}$$

Computation of the transforms (1.1) at a complete set of irreducible representations in Young's orthogonal form gives us the harmonic analysis of f, because the scaled matrix entry $\frac{\dim \rho}{|S_n|}\left[\hat{f}(\rho)\right]_{ij}$ is the coefficient of the function $(s \mapsto [\rho(s)]_{ij})$ in the expansion of f in the basis of matrix coefficients. We prove the following theorem, which counts the number of arithmetic operations required to compute a collection of Fourier transforms on S_n.

Theorem 1.1. *The Fourier transform of a complex function on the symmetric group S_n may be computed at a complete set of irreducible matrix representations in Young's orthogonal form in no more than $\frac{3n(n-1)}{4}|S_n|$ multiplications and the same number of additions.*

Although we have stated Theorem 1.1 for Young's orthogonal form, the same techniques may be used to prove a more general result that applies, e.g., to Young's seminormal form as well. Results on the complexity of the corresponding inverse Fourier transform follow immediately by considering the transpose of our algorithms.

There are several novel features of our approach to the computation of Fourier transforms. One is the use of a kind of commutativity in the group algebra of the symmetric group that lets us replace an iterated group algebra product by a sequence of bilinear maps. This allows us to write an expression for the Fourier transform in a form similar to Horner's rule, and leads to an efficient algorithm.

Another interesting feature is the appearance of certain combinatorial objects that generalize Young tableaux. It is well known that Young tableaux may be associated with sequences of partitions, each obtained by adding a box to the Young diagram of the previous one. This corresponds to an upward walk in a partially ordered set called Young's lattice (see [23] and [24] for a discussion of combinatorial problems associated with these and other walks). In the current paper we encounter sequences of partitions that satisfy more general relations corresponding to the mapping of a multiply-connected graph into Young's lattice. In joint work with Dan Rockmore, such ideas have been generalized to apply to the computation of Fourier transforms on other finite groups [20].

The organization of the paper is as follows. Section 2 contains a sketch of the proof of the main theorem modulo four 'properties' and some combinatorial lemmas. In Section 3 we prove the combinatorial lemmas, and discuss their extension to counting problems in Bratteli diagrams. We state results on the computation of Fourier transforms on homogeneous spaces, and their inverse transforms, in Section 4. We conclude in Section 5.

All that follows makes up an extended abstract of [17]. Although we have tried to make the paper relatively self-contained, we do use a number of facts from representation theory, that may be found in the books of Serre [22], James and Kerber [13], and Macdonald [15]. Background from the theory of computation of Fourier transforms may be found in the book of Clausen and Baum [2], and in the articles [19] and [18].

2. Fast transforms on the symmetric group

In 1930, Alfred Young gave an explicit description of the irreducible representations of the symmetric group, called Young's orthogonal form. Let α_n be a partition of n. Then the

Young's orthogonal form Δ_{α_n} is a matrix representation of S_n whose rows and columns are indexed by standard Young tableaux on the Young diagram of α_n, or equivalently [15], by sequences of partitions

$$\boldsymbol{\alpha} = \quad \alpha_n \longrightarrow \alpha_{n-1} \longrightarrow \cdots \longrightarrow \alpha_2 \longrightarrow \alpha_1 \longrightarrow (\alpha_0 = \phi) \tag{2.1}$$

where α_i is a partition of i, and $\alpha \longrightarrow \beta$ indicates that the partition β may be obtained from α by removing a single box. Δ_{α_n} is determined by the representing matrices of the simple transpositions $s_i = (i-1\ i)$, given in coordinates as

$$[\Delta_{\alpha_n}(s_i)]_{\boldsymbol{\beta},\boldsymbol{\alpha}} = \delta_{\alpha_{n-1},\beta_{n-1}} \cdots \delta_{\alpha_i,\beta_i} \cdot P_{s_i}\begin{pmatrix} \beta_i & \beta_{i-1} \\ \alpha_{i-1} & \alpha_{i-2} \end{pmatrix} \cdot \delta_{\alpha_{i-2},\beta_{i-2}} \cdots \delta_{\alpha_1,\beta_1}, \tag{2.2}$$

where $\boldsymbol{\beta}, \boldsymbol{\alpha}$ are Young tableaux on the Young diagram of α_n, and P_{s_i} is defined as follows: For any two boxes b_1 and b_2 in a Young diagram, we define the axial distance from b_1 to b_2 to be $d(b_1, b_2)$, where

$$d(b_1, b_2) = \mathrm{row}(b_1) - \mathrm{row}(b_2) + \mathrm{column}(b_1) - \mathrm{column}(b_2).$$

Now suppose $\alpha_i = \beta_i$ and $\alpha_{i-2} = \beta_{i-2}$. Then the skew diagrams of $\beta_i - \beta_{i-1}$ and $\beta_{i-1} - \alpha_{i-2}$ each consist of a single box, and P_{s_i} may be written as

$$P_{s_i}\begin{pmatrix} \beta_i & \beta_{i-1} \\ \alpha_{i-1} & \alpha_{i-2} \end{pmatrix} = \begin{cases} d(\beta_i - \beta_{i-1}, \beta_{i-1} - \alpha_{i-2})^{-1} & \text{if } \alpha_{i-1} = \beta_{i-1}, \\ \sqrt{1 - d(\beta_i - \beta_{i-1}, \beta_{i-1} - \alpha_{i-2})^{-2}} & \text{if } \alpha_{i-1} \neq \beta_{i-1}. \end{cases} \tag{2.3}$$

For a proof of this formula, in slightly different notation, see [13], Chapter 3.

The algorithm that we present for computing Fourier transforms on symmetric groups depends on a combinatorial indexing scheme that generalizes Young tableaux. We consider, for $2 \le i \le n$, sequences of partitions $\gamma_1, \ldots, \gamma_{n-1}$, $\beta_1, \ldots, \beta_i$, $\alpha_{i-2}, \ldots, \alpha_{n-2}$ (with α_j a partition of j etc.), which satisfy the relations

(2.4)
$$\begin{tikzcd}
 & & & \beta_i \arrow[r] \arrow[d] & \beta_{i-1} \arrow[r] \arrow[d] & \cdots \arrow[r] & \beta_1 \arrow[d] \; . \\
\gamma_{n-1} \arrow[r] \arrow[dr] & \alpha_{n-2} \arrow[r] & \cdots \arrow[r] & \alpha_{i-1} \arrow[r] & \alpha_{i-2} & & \phi \\
 & \gamma_{n-2} \arrow[r] & {} & \cdots & {} \arrow[r] & \gamma_1 \arrow[ur] &
\end{tikzcd}$$

Let $\mathcal{F}_i^n$ denote the number of such sequences, and let $f_i = \mathcal{F}_i^i/(i-1)!$.

The number of arithmetic operations taken by our algorithm may be expressed in terms of $\mathcal{F}_i^n$, and the combinatorial lemmas proven in Section 3 allow us to further express this count in terms of f_i. This gives us the statement of Theorem 2.1. To obtain Theorem 1.1 of the introduction, note that by Lemma 3.3 of Section 3, we have $f_i \le 3(i-1)$.

Theorem 2.1. *The Fourier transform of a complex function on the symmetric group S_n may be computed at a complete set of irreducible matrix representations in Young's orthogonal form in no more than*

$$\left(\sum_{k=2}^{n}\sum_{i=2}^{k}\frac{1}{k}f_i\right)\cdot|S_n|$$

multiplications and the same number of additions.

Proof Sketch. Let f be any complex function on S_n, and let $\Delta = \bigoplus_{\alpha_n \vdash n} \Delta_{\alpha_n}$ be the model representation obtained as the direct sum of all the representations of Young's orthogonal form. Let t_i denote the image $\Delta(s_i)$ of the simple transposition $s_i = (i-1\ i)$.

We start by noting that the group elements

$$s_2 \cdots s_n, \quad s_3 \cdots s_n, \quad \ldots, \quad s_n, \quad e,$$

form a complete set of coset representatives for S_n relative to S_{n-1}. Factoring each element of S_n as a product of one of these minimal coset representatives and an element of S_{n-1}, and substituting the result into (1.1) shows that

$$\bigoplus_{\alpha_n \vdash n} \hat{f}(\Delta_{\alpha_n}) = \Sigma = \sum_{i=1}^{n} t_{i+1} \cdots t_n \cdot F_i, \tag{2.5}$$

where

$$F_i = \sum_{g \in S_{n-1}} \Delta(g) f(s_{i+1} \ldots s_n \cdot g). \tag{2.6}$$

Any representation in Young's orthogonal form has the property that its restriction to S_{n-1} is a matrix direct sum of representations of S_{n-1} in Young's orthogonal form according to the branching theorem. Thus, equation (2.6) shows that F_i is a direct sum of Fourier transforms on S_{n-1}.

This suggests a recursive procedure for computing Fourier transforms. We first compute the Fourier transforms of the functions $g \mapsto f(s_{i+1} \ldots s_n g)$ at all irreducible representations of S_{n-1} in Young's orthogonal form, and then assemble the results to form the block diagonal matrices F_i. Finally, we compute the matrix sum of products (2.5) to obtain the Fourier transforms of f.

We shall rearrange the sum (2.5) in a form similar to Horner's rule, and show that such a sum may be computed in no more than $\sum_{i=2}^{n} \mathcal{F}_i^n$ scalar operations, given the matrices F_i. Together with Lemma 3.2 and the recursive nature of the algorithm, this will allow us to prove the theorem by induction.

The key to rearranging the sum (2.5) is to permute the order in which the matrix multiplications are performed. Let $A_j = \operatorname{span}_{\mathbf{C}} \Delta(S_j)$ denote the image of the group algebra of S_i under Δ, so that $F_i \in A_{n-1}$ and $\Sigma \in A_n$. We claim that there is a sequence of bilinear maps $\underset{2}{*}, \ldots, \underset{n}{*}$, and spaces $V_1, \ldots, V_n$, $C_2, \ldots, C_n$, such that the following four properties hold.

Prop. 1. $V_1 = A_{n-1}$ and $V_n = A_n$. For $2 \leq i \leq n$,

$$C_i = A_i \cap \text{Centralizer}(A_{i-2}),$$

and the map $\underset{i}{*} : V_{i-1} \times C_i \to V_i$ is bilinear.

Prop. 2. If $F \in V_1$ and $c_i \in C_i$ for $2 \leq i \leq n$, then

$$c_2 \cdot c_3 \cdots c_n \cdot F = \big(\cdots \big(F \underset{2}{*} c_2\big) \underset{3}{*} c_3 \cdots\big) \underset{n}{*} c_n$$

Prop. 3. For each i with $2 \leq i \leq n$, the map $F \longmapsto \big(\cdots\big(F \underset{2}{*} e\big) \underset{3}{*} e \cdots\big) \underset{i}{*} e \ \in V_i$ requires no arithmetic computation to apply, where e denotes the identity element in A_n.

Prop. 4. Given $v_{i-1} \in V_{i-1}$, $c_i \in C_i$ and $v_i \in V_i$, we may compute $v_{i-1} \underset{i}{*} c_i + v_i$ by means of $\mathcal{F}_i^n$ multiplications and the same number of additions.

In order to simplify the presentation, we shall defer the construction of $\underset{i}{*}$ and V_i until after the proof. Prop. 1–4 are simple consequences of the definitions, and we refer the reader to the full version of this paper [17], for their proofs. In particular, Prop. 4 follows immediately from equation (2.12). The spaces V_i, $1 < i < n$ will be constructed with a natural choice of basis, and it is with respect to these bases that the complexity statements Prop. 3 and Prop. 4 are to be interpreted.

Using Prop. 1 and Prop. 2, it is easy to rearrange (2.5) into a more manageable form.

$$
\begin{aligned}
\Sigma &= \sum_{i=1}^{n} e \cdots e \cdot t_{i+1} \cdots t_n \cdot F_i \\
&= \sum_{i=1}^{n} \Big(\cdots\Big(\Big(\cdots\big(F_i \underset{2}{*} e\big) \underset{3}{*} \cdots\Big) \underset{i}{*} e\Big) \underset{i+1}{*} t_{i+1} \cdots\Big) \underset{n}{*} t_n \\
&= \Bigg[\Bigg[\cdots\Big[\Big[F_1 \underset{2}{*} t_2 + F_2 \underset{2}{*} e\Big] \underset{3}{*} t_3 + \big(F_3 \underset{3}{*} e\big) \underset{3}{*} e\Big] \underset{4}{*} t_4 \\
&\qquad \cdots \qquad\qquad + \cdots \qquad\qquad \Bigg] \underset{n-1}{*} t_{n-1} \\
&\qquad\qquad + \Big(\cdots\big(F_{n-1} \underset{2}{*} e\big) \underset{3}{*} \cdots\Big) \underset{n-1}{*} e\Bigg] \underset{n}{*} t_n \\
&\qquad\qquad + \Big(\cdots\big(F_n \underset{2}{*} e\big) \underset{3}{*} \cdots\Big) \underset{n}{*} e
\end{aligned}
\tag{2.7}
$$

The algorithm for computing Σ given $F_1, \ldots F_n$ proceeds in the obvious way:

Stage 1. Let $G_1 = F_1$.

Stage i. Let $G_i = G_{i-1} \underset{i}{*} t_i + \big(\ldots\big(F_i \underset{2}{*} e\big) \underset{3}{*} \ldots\big) \underset{i}{*} e$, for $2 \leq i \leq n$,

Stage n. Let $\Sigma = G_n = G_{n-1} \underset{n}{*} t_n + F_n$.

A quick look at (2.7) verifies that $\Sigma = G_n$.

Assume that the F_i are given and the t_i have been precomputed in using (2.3). Then Stage 1 requires no computation, and by Prop. 3 and Prop. 4 the computation of G_i from G_{i-1} and F_i at Stage i requires no more than $\mathcal{F}_i^n$ scalar operations. Adding the operation counts for all the stages shows that the computation of Σ given $F_1, \ldots, F_n$ takes $\sum_{i=2}^{n} \mathcal{F}_i^n$ multiplications and the same number of additions.

Now let T_{S_n} denote the number of multiplications (or additions) required by our algorithm to compute the Fourier transform of any function on S_n at the representations of Young's orthogonal form. Then the computation of each matrix F_i takes $T_{S_{n-1}}$ multiplications (or additions). Therefore T_{S_n} satisfies the recursion relation

$$
T_{S_n} \leq n T_{S_{n-1}} + \sum_{i=2}^{n} \mathcal{F}_i^n. \tag{2.8}
$$

To solve this recursion, divide both sides by $n!$, and note that by Lemma 3.2, $\mathcal{F}_i^n = (n-1)! f_i$, to get

$$
\frac{1}{n!} T_{S_n} \leq \frac{1}{(n-1)!} T_{S_{n-1}} + \frac{1}{n} \sum_{i=2}^{n} f_i. \tag{2.9}
$$

Applying (2.9) recursively gives the result. □

Remark 2.2. Clausen's algorithm [4] calculates the products $t_{i+1} \cdots t_n \cdot F_i$ occurring in (2.5) by matrix multiplication of the corresponding matrices in the order from right to left. By (2.2), the matrices corresponding to to t_j are sparse so the product $t_j \cdot (t_{j+1} \cdots t_n \cdot F_i)$, $i < j$ may be computed efficiently given t_j and $t_{j+1} \cdots t_n \cdot F_i$.

Clausen's algorithm requires roughly $\frac{(n+1)n(n-1)}{3}|S_n|$ multiplications or additions, so Theorems 1.1 and 2.1 represent an improvement of order a factor n.

2.1. **Definition of the spaces and maps.** We shall first define spaces $\overline{V}_i, \overline{C}_i$ that almost satisfy Prop. 1–4, except for a minor technicality. For $1 \le i \le n$ we define $\overline{V}_i$ to be the space of complex functions of the form $G(\beta_i, \ldots, \beta_1; \alpha_{n-2}, \ldots, \alpha_{i-1}; \gamma_{n-1} \ldots \gamma_1)$, where $\alpha_{i-1}, \ldots, \alpha_{n-2}$, $\beta_1, \ldots, \beta_i$, and $\gamma_1, \ldots, \gamma_{n-1}$, range over partitions satisfying the restriction relations (2.10).

(2.10)
$$\begin{tikzcd}
 & & & \beta_i \arrow[r, dash] \arrow[d, dash] & \beta_{i-1} \arrow[r, dash] & \cdots \arrow[r, dash] & \beta_1 \arrow[d, dash] \\
\gamma_{n-1} \arrow[r, dash] \arrow[dr, dash] & \alpha_{n-2} \arrow[r, dash] & \cdots \arrow[r, dash] & \alpha_{i-1} & & & \phi \arrow[dl, dash] \\
 & \gamma_{n-2} \arrow[r, dash] & & \cdots & & \gamma_1 \arrow[l, dash] &
\end{tikzcd}$$

Define $\overline{C}_i$ to be the space of complex functions of the form $P(\beta_i, \beta_{i-1}; \alpha_{i-1}, \alpha_{i-2})$, where $\beta_i, \beta_{i-1}; \alpha_{i-1}, \alpha_{i-2}$ are partitions (with β_i a partition of i, etc.) which satisfy (2.11).

(2.11)
$$\begin{tikzcd}
\beta_i \arrow[r, dash] \arrow[d, dash] & \beta_{i-1} \arrow[d, dash] \\
\alpha_{i-1} \arrow[r, dash] & \alpha_{i-2}
\end{tikzcd}$$

Assume that $2 \le i \le n$, that $G_{i-1} \in \overline{V}_{i-1}$, and that $P_i \in \overline{C}_i$. Then we define $G_{i-1} \overline{\underset{i}{*}} P_i \in \overline{V}_i$ by

$$\left[G_{i-1} \overline{\underset{i}{*}} P_i\right] \begin{pmatrix} \beta_i & \ldots & \beta_1 \\ \alpha_{n-2} \ldots & \alpha_{i-1} \\ \gamma_{n-1} & \cdots & \gamma_1 \end{pmatrix} = \sum_{\alpha_{i-2}} G_{i-1} \begin{pmatrix} \beta_{i-1} & \ldots & \beta_1 \\ \alpha_{n-2} \ldots & \alpha_{i-2} \\ \gamma_{n-1} & \cdots & \gamma_1 \end{pmatrix} \cdot P_i \begin{pmatrix} \beta_i & \beta_{i-1} \\ \alpha_{i-1} & \alpha_{i-2} \end{pmatrix}, \tag{2.12}$$

where α_{i-2} satisfies (2.11). Notice that in going from G_{i-1} to $G_{i-1} \overline{\underset{i}{*}} P_i$ we remove a dependence on α_{i-2} and add a dependence on β_i.

We may now define the spaces V_i. For $1 < i \le n$ let $V_i = \overline{V}_i$. When $i = n$, a collection of partitions satisfying (2.10) is equivalent to specifying a pair of standard Young tableaux of the same shape, so $V_n = A_n$ as desired. On the other hand, by a standard characterization of subalgebras and centralizer algebras (see [12] Proposition 2.3.12 for a proof in different notation), we know that the following expressions define isomorphisms of $\overline{C}_i$ and $\overline{V}_1$ into C_i and A_{n-1} respectively:

$$[P_i]_{\boldsymbol{\beta},\boldsymbol{\alpha}} = \delta_{\alpha_{n-1},\beta_{n-1}} \cdots \delta_{\alpha_i,\beta_i} \cdot P_i \begin{pmatrix} \beta_i & \beta_{i-1} \\ \alpha_{i-1} & \alpha_{i-2} \end{pmatrix} \cdot \delta_{\alpha_{i-2},\beta_{i-2}} \cdots \delta_{\alpha_1,\beta_1}$$
$$[G_1]_{\boldsymbol{\alpha},\boldsymbol{\gamma}} = \delta_{\alpha_{n-1},\beta_{n-1}} \cdot G_1 \begin{pmatrix} & (1) \\ \alpha_{n-2} \ldots & \alpha_0 \\ \gamma_{n-1} \cdots & \gamma_1 \end{pmatrix},$$

where the left hand sides denote the matrix entries of the corresponding elements of C_i and A_{n-1}. To define $\underset{i}{*}$ simply pull back the definition of $\overline{\underset{i}{*}}$ along these isomorphisms.

3. Combinatorial lemmas

We now turn to the combinatorial lemmas needed to complete the proof of Theorem 2.1. Assume that $i \geq j$, that α is a partition of i, and that β is a partition of j. Then let $\mathcal{M}(\alpha, \beta)$ denote the number of sequences of partitions $\alpha_j, \dots, \alpha_i$ such that

$$(\alpha = \alpha_i) \longleftarrow \alpha_{i-1} \longleftarrow \cdots \longleftarrow \alpha_{j+1} \longleftarrow (\alpha_j = \beta) \; .$$

The function $\mathcal{M}$ has a number of other equivalent definitions.

$$\begin{aligned} \mathcal{M}(\alpha, \beta) &= \text{multiplicity of } \Delta_\beta \text{ in the restriction of } \alpha \text{ to } S_j. \\ &= \text{number of standard tableaux on the skew diagram } \alpha - \beta \\ &= \text{number of ways of removing boxes from } \alpha \text{ to get } \beta. \end{aligned}$$

These numbers are a special case of the Kostka numbers [15] and are usually denoted $K_{\alpha-\beta,(1^{|\alpha|-|\beta|})}$, although [13] writes $k_{\alpha/\beta,(1^{|\alpha|-|\beta|})}$. It is easily shown (cf. [12] Corollary 2.3.2) that if α, β are partitions of i and j respectively, and $j \leq k \leq i$, then

$$\mathcal{M}(\alpha, \beta) = \sum_{\alpha_k} \mathcal{M}(\alpha, \alpha_k) \mathcal{M}(\alpha_k, \beta), \tag{3.1}$$

where α_k ranges over all partitions of k.

We shall also use the notation $d_\alpha = \mathcal{M}(\alpha, \phi)$. Thus d_α is the dimension of the representation Δ_α, and may be calculated using the famous hook-length formula of Frame, Robinson, and Thrall (see [15] or [13]).

Recall that $\mathcal{F}_i^n$ denotes the number of sequences of partitions $\gamma_1, \dots, \gamma_{n-1}$, $\beta_1, \dots, \beta_i$, $\alpha_{i-2}, \dots, \alpha_{n-2}$ (with α_j a partition of j etc.), which satisfy the relations (2.4).

Lemma 3.1.

$$\mathcal{F}_i^n = \sum_{\substack{\beta_i, \beta_{i-1} \\ \gamma_{n-1}, \alpha_{i-1}, \alpha_{i-2}}} \mathcal{M}(\gamma_{n-1}, \alpha_{i-1}) \mathcal{M}(\alpha_{i-1}, \alpha_{i-2}) \mathcal{M}(\beta_i, \alpha_{i-1}) \mathcal{M}(\beta_i, \beta_{i-1}) \mathcal{M}(\beta_{i-1}, \alpha_{i-2}) d_{\beta_{i-1}} d_{\gamma_{n-1}}, \tag{3.2}$$

where $\alpha_j, \beta_j, \gamma_j$ range over partitions of j.

Proof. We count the sequences satisfying (2.4) as follows. First choose the partitions $\alpha_{i-2}, \alpha_{i-1}, \beta_{i-1}, \beta_i, \gamma_{n-1}$ subject only to the restrictions that α_{i-2} is a partition of $i-2$, etc. Then the number of ways of choosing the chain of partitions from α_{i-1} to γ_{n-1} is $\mathcal{M}(\gamma_{n-1}, \alpha_{i-1})$. Similarly, the number of ways of choosing the chain of partitions $\phi, \gamma_1, \dots$ from ϕ to γ_{n-1} is $d_{\gamma_{n-1}}$, and the number of ways of choosing the chain of partitions from ϕ to β_{i-1} is $d_{\beta_{i-1}}$. Furthermore, these choices are independent given the choices of γ_{n-1}, α_{i-1} and β_{i-1}. Finally we note that the choice of $\alpha_{i-2}, \alpha_{i-1}, \beta_{i-1}, \beta_i$ is only consistent with (2.4) when the product

$$\mathcal{M}(\alpha_{i-1}, \alpha_{i-2}) \mathcal{M}(\beta_i, \alpha_{i-1}) \mathcal{M}(\beta_i, \beta_{i-1}) \mathcal{M}(\beta_{i-1}, \alpha_{i-2})$$

is nonzero. This product is always either 0 or 1, so the number of sequences satisfying (2.4) may be found by summing the product

$$\mathcal{M}(\gamma_{n-1}, \alpha_{i-1}) d_{\gamma_{n-1}} d_{\beta_{n-1}} \mathcal{M}(\alpha_{i-1}, \alpha_{i-2}) \mathcal{M}(\beta_i, \alpha_{i-1}) \mathcal{M}(\beta_i, \beta_{i-1}) \mathcal{M}(\beta_{i-1}, \alpha_{i-2})$$

over all choices of $\alpha_{i-2}, \alpha_{i-1}, \beta_{i-1}, \beta_i, \gamma_{n-1}$. ∎

Suppose β is a partition. Then let $\mathrm{jmp}(\beta)$ denote the number of jumps in the Young diagram of β. For example, if $\beta = (4, 3, 3, 1, 1)$, then $\mathrm{jmp}(\beta) = 3$.

Lemma 3.2. 1. $\mathcal{F}_i^n = \frac{(n-1)!}{(i-1)!}\mathcal{F}_i^i$.

2. $\mathcal{F}_i^i = (i-1)\cdot(i-1)! + \sum_{\beta_{i-1}} \mathrm{jmp}(\beta_{i-1})^2 d_{\beta_{i-1}}^2$, *where β_{i-1} ranges over partitions of $i-1$.*

Proof. 1 follows immediately from (3.2) by Frobenius reciprocity, since for any α_{i-1} we have

$$\sum_{\gamma_{n-1}} \mathcal{M}(\gamma_{n-1},\alpha_{i-1})d_{\gamma_{n-1}} = \dim \mathrm{Ind}_{S_{i-1}}^{S_{n-1}} \Delta_{\alpha_{i-1}} = |S_{n-1}/S_{i-1}|\cdot d_{\alpha_{i-1}}.$$

For 2 we start with the sum (3.2) in the case $n=i$, and split it into two parts, distinguishing the cases where $\alpha_{i-1}\neq\beta_{i-1}$ and $\alpha_{i-1}=\beta_{i-1}$. If α_{i-1} and β_{i-1} are distinct partitions of $i-1$ which are both obtained from β_i by removing a box, then they jointly determine β_i (and α_{i-2}), since the boxes removed to get to these two partitions are distinct. Thus the contribution to $\mathcal{F}_i^i$ from terms with $\alpha_{i-1}\neq\beta_{i-1}$ may be written as

$$\begin{aligned}\sum_{\substack{\alpha_{i-2}\\ \alpha_{i-1},\beta_{i-1}}} \mathcal{M}(\beta_{i-1},\alpha_{i-2})\mathcal{M}(\alpha_{i-1},\alpha_{i-2})d_{\alpha_{i-1}}d_{\beta_{i-1}} &- \sum_{\beta_{i-1},\alpha_{i-2}} \mathcal{M}(\beta_{i-1},\alpha_{i-2})^2 d_{\beta_{i-1}}^2\\ &= (i-1)\,|S_{i-1}| - \sum_{\beta_{i-1}} \mathrm{jmp}(\beta_{i-1})d_{\beta_{i-1}}^2,\end{aligned}$$

where we have used (3.1) and Frobenius reciprocity to simplify the result.

On the other hand, if $\alpha_{i-1}=\beta_{i-1}$, then the only conditions on β_i and α_{i-2} are that they may be obtained from β_{i-1} by adding or removing a box, respectively. In this case, given β_{i-1}, there are $\mathrm{jmp}(\beta_{i-1})+1$ ways of choosing β_i, and $\mathrm{jmp}(\beta_{i-1})$ ways of choosing α_{i-2}. Thus the contribution to $\mathcal{F}_i^i$ from terms with $\alpha_{i-1}=\beta_{i-1}$ is

$$\sum_{\beta_{i-1}} \mathrm{jmp}(\beta_{i-1})(\mathrm{jmp}(\beta_{i-1})+1)d_{\beta_{i-1}}^2.$$

□

Lemma 3.3. $\mathcal{F}_i^n \leq \frac{3(i-1)}{n}\,|S_n|$, *and hence $f_i \leq 3(i-1)$.*

Proof. In light of Lemma 3.2, it suffices to show that for any partition β_i of i, we have $\mathrm{jmp}(\beta_i)^2 \leq 2i$. Let $a=\mathrm{jmp}(\beta_i)$. By deleting rows and columns from the Young diagram of β_i, we may obtain a new partition with fewer boxes, but the same number of jumps, and the Young diagram of this new partition can be made to have a staircase form, i.e., the new partition is exactly $(a,a-1,\dots,1)$. For an example, see Figure 1. The number of boxes in the staircase $(a,a-1,\dots,1)$ is $\frac{1}{2}a(a+1)$, which shows that $a(a+1)\leq 2i$. □

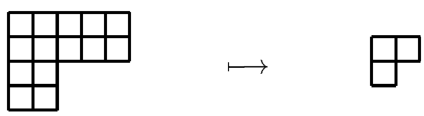

FIGURE 1. Removing rows and columns to obtain a staircase.

Remark 3.4. The same techniques used to prove Lemma 3.2 part 1 also show that

$$\dim V_i = \frac{i}{n}\,|S_n|\,.$$

The analogous problem of finding an explicit formula for $\mathcal{F}_i^n$ in closed form, if one exists, appears to be much more difficult.

Lemma 3.2 allows us to calculate $\mathcal{F}_i^i$, and hence the exact complexity of our algorithm for computing Fourier transforms on S_i, for small values of i. We have done this for $1 \leq i \leq 50$, which certainly includes all cases where the algorithm might ever be implemented. In Table 1 we display these values for $1 \leq i \leq 8$, where T_{S_i} denotes the number of additions (or the number of multiplications) taken by the algorithm for computing Fourier transforms on S_i.

i	1	2	3	4	5	6	7	8
$\sum_{\beta_i} \mathrm{jmp}(\beta_i)^2 d_{\beta_i}^2$	1	2	18	78	474	4004	32404	290558
$\mathcal{F}_i^i$	0	2	6	36	174	1074	8324	67684
T_{S_i}	0	2	16	130	1088	9792	96452	1034656
$\frac{1}{\lvert S_i \rvert} T_{S_i}$	0	1	2.67	5.42	9.07	13.6	19.1	25.7

TABLE 1. Exact sequence and operation counts.

3.1. **Remarks concerning the combinatorial lemmas.** Lemmas 3.1–3.3 have some simple generalizations, which become important when we extend the algorithm for Fourier transforms on the symmetric group to other finite groups and semisimple algebras. The main observation is that Young's lattice may be replaced by other Bratteli diagrams.

Let $\mathbf{N}$ denote the non-negative integers. A **Bratteli diagram** (see [12] and [25]) is a connected $\mathbf{N}$-graded multigraph such that

(i) Each level (vertices with the same grading) has only finitely many vertices, and finitely many edges connected to it.
(ii) Edges only connect adjacent levels, and if two adjacent levels are nonempty, then the bipartite graph, consisting of those two levels and the edges connecting them, is connected.
(iii) The zeroth level contains a unique vertex, denoted ϕ.

Given any Bratteli diagram and two vertices α, β in the diagram, we let $\mathcal{M}(\alpha, \beta)$ denote the number of upward paths in the diagram from β to α. As before, we let $d_\alpha = \mathcal{M}(\alpha, \phi)$

The definition of $\mathcal{F}_i^n$ is easy to generalize to any Bratteli diagram which has at least $n+1$ levels: given such a Bratteli diagram, we define $\mathcal{F}_i^n$ to be the number of grading-preserving maps from a graded graph of the form (2.4) into the Bratteli diagram. Each such map not only sends $\alpha_j, \beta_j, \gamma_j$ into vertices of level j, but also sends each edge into an edge of the Bratteli diagram.

With these definitions of $\mathcal{M}$ and $\mathcal{F}_i^n$, the statement of Lemma 3.1 holds with the only change being that $\alpha_j, \beta_j, \gamma_j$ now range over vertices at level j in the Bratteli diagram. To generalize Lemma 3.2, we need to place some extra conditions on our Bratteli diagrams.

Any Bratteli diagram is uniquely associated to a chain of semisimple algebras, called path algebras (see [12] 2.3.11). The i^{th} path algebra A_i has dimension $\dim A_i = \sum_{\beta_i} d_{\beta_i}^2$, where β_i ranges over vertices at level i, and A_i contains A_{i-1} as a subalgebra. A Bratteli diagram is **locally free** if A_i is free over A_{i-1} for $i \geq 1$.

A Bratteli diagram is **multiplicity free** if $\mathcal{M}$ is either 0 or 1 for any two adjacent vertices.

Lemma 3.5. 1. *Assume $n \geq i \geq 1$. Then for any locally free Bratteli diagram,* $\dim A_i$ *is an integer multiple of* $\dim A_{i-1}$*, and*

$$\mathcal{F}_i^n = \frac{\dim A_{n-1}}{\dim A_{i-1}} \mathcal{F}_i^i$$

2. *For any locally free, multiplicity free Bratteli diagram,*

$$\mathcal{F}_i^i = \frac{(\dim A_{i-1})^2}{\dim A_{i-2}} + \sum_{\beta_{i-1}} (c^+(\beta_{i-1}) - 1)c^-(\beta_{i-1})d^2_{\beta_{i-1}}$$

where β_{i-1} *ranges over vertices at level* $i-1$*, and* $c^+(\beta_{i-1}), c^-(\beta_{i-1})$ *are the number of edges from* β_{i-1} *to levels* i *and* $i-2$ *respectively.*

Several other combinatorial results for Young's lattice also extend to locally free Bratteli diagrams: In particular, the theorems of Stanley ([23] Theorem 3.7 and [24] Theorem 2.7) which count the number of paths in a differential poset, which start and end at ϕ, hold in this more general setting. See [20] for a more detailed treatment of these and similar results.

4. Homogeneous spaces and inverse transforms

Any complex function on a homogeneous space may also be considered to be a function on the group which is constant on cosets. In this way we may apply Fourier analysis on the group to functions on any homogeneous space. We direct the reader to [19] for general background on the computation of Fourier transforms on homogeneous spaces.

Let f denote a complex function on the homogeneous space S_n/S_{n-k}, and let ρ be any (class-1) irreducible matrix representation of S_n given in Young's orthogonal form. Then the **Fourier transform of f at** ρ is defined to be the Fourier transform at ρ of the right S_{n-k}-invariant function $\tilde{f}$ on S_n defined by

$$\tilde{f}(g) = \frac{1}{|S_{n-k}|} f(gS_{n-k}).$$

To compute Fourier transforms on S_n/S_{n-k} efficiently, it suffices to see how the algorithm of Section 2 simplifies when applied to a right S_{n-k}-invariant function. In practice, this means following through the algorithms and proofs of Sections 2 and 3 under the added assumption that γ_{n-k} is equal to the single row partition $(n-k)$.

Theorem 4.1. *The Fourier transform of a complex function on the homogeneous space* S_n/S_{n-k} *may be computed at a complete set of (class-1) irreducible matrix representations in Young's orthogonal form in no more than* $\frac{3k(2n-k-1)}{4}|S_n/S_{n-k}|$ *multiplications and the same number of additions.*

A proof of Theorem 4.1 is given in the full version of this paper [17].

Remark 4.2. Theorem 4.1 is an improvement on the result of Maslen and Rockmore [19] Theorem 6.5, which was obtained by applying Clausen's algorithm [4] to a right invariant function on the symmetric group. They showed that the Fourier transform of a complex function on S_n/S_{n-k} could be computed at an adapted set of representations in no more than $k\left(n^2 - kn + \frac{1}{3}(k^2-1)\right)|S_n/S_{n-k}|$ scalar operations.

4.1. Inverse transforms.

Theorems 2.1 and 4.1 immediately give us a method for computing inverse Fourier transforms as well. The inverse Fourier transform operator is the same as the transpose of a Fourier transform, up to a diagonal matrix (i.e. scaling the inputs by $\frac{\dim \rho}{|S_n/S_{n-k}|}$, see [17]). A well known principle (see [2], Chapter 3) states that the transpose algorithm of an invertible linear algorithm has the same number of operations as the original, so by transposing the algorithms used to prove Theorems 2.1 and 4.1 we obtain Theorem 4.3, which we state without further proof.

Theorem 4.3. *For each partition $\alpha \vdash n$, let $F(\alpha)$ be a complex $d_\alpha \times d_\alpha$ matrix with zeroes in those columns which are not S_{n-k}-invariant columns of Δ_α. Then the inverse Fourier transform*

$$f(gS_{n-k}) = \frac{1}{|S_n/S_{n-k}|} \sum_{\alpha \vdash n} d_\alpha \operatorname{Trace}(F(\alpha)\Delta_\alpha(g^{-1})) \tag{4.1}$$

may be computed in no more than $\frac{3k(2n-k-1)}{4} |S_n/S_{n-k}|$ multiplications and the same number of additions.

Remark 4.4. Diaconis and Rockmore [10] discuss the computation of isotypic projections of functions on a homogeneous space X. The current techniques can be applied to efficiently compute the isotypic projections of functions on the space S_n/S_{n-k}, by means of a Fourier transform, truncation of data, and an inverse Fourier transform.

5. Conclusion

Although the results presented in this paper are specific to the symmetric group, the techniques used to obtain them are much more general. The use of Gel'fand-Tsetlin bases which generalize Young's orthogonal form, the choice of factorizations for group elements or coset representatives, and the rearrangement of sums similar to Horner's rule are all well known tools for computing Fourier transforms on finite groups [2] and compact Lie groups [16]. Together they form the basis for the general 'separation of variables' method for constructing Fourier transform algorithms [19] [20].

The construction of the bilinear maps at the end of Section 2 may be generalized to any finite group or semisimple algebra. Given a system of Gel'fand-Tsetlin bases, a collection of products of group elements, and a permutation, there is a well defined sequence of bilinear maps that allow the products to be rearranged (cf., Prop. 2) according to the chosen permutation. The spaces on which the bilinear maps are defined are associated to diagrams generalizing (2.10) (2.4), and formulae for the number of operations needed to apply these maps can be read off the diagrams, in terms of restriction multiplicities. In joint work with Dan Rockmore [20], such ideas have been systematically developed, and applied to the computation of Fourier transforms on a variety of groups and algebras.

The methods used in Section 3 also raise some new combinatorial questions. Walks generalizing (2.10) have already been studied on Young's lattice and other posets [23] [24], but the appearance of multiply-connected configurations, e.g., (2.4), appears to be a new phenomenon; also see [16] and [19]. More generally, one may consider mappings from any graded diagram into a Bratteli diagram. Such objects appear in the construction of Fourier transform algorithms on other finite groups [20], and the complexity of the algorithms may again be obtained by counting the mappings. There are always expressions for the numbers of these objects, generalizing equation (3.2) for $\mathcal{F}_i^n$, but it is not clear when these expressions may be evaluated in closed form. We do not even know the answer for (3.2) itself.

Finally, we should note that, even for the symmetric groups, the problem of computing Fourier transforms is far from completely solved. In particular, some applications [7] [10] require the use of Gel'fand-Tsetlin bases for chains of parabolic subgroups other than the standard chain. Although Clausen's algorithm and the algorithms in this paper may both be adapted to these new situations, the results are less convincing.

We have not yet implemented the algorithms in this paper. Because of their close relationship with Clausen's algorithm, we expect these algorithms to be stable and efficient, in practice as well as in theory.

Acknowledgements

I would like to thank Persi Diaconis, Dan Rockmore, and Micheal Clausen for their encouragement, comments, and advice. I would also like to thank Institut des Hautes Études Scientifiques and Universiteit Utrecht, which supported me during the writing of this paper.

References

[1] T. Beth, *Verfahren der schnellen Fourier–Transformation*, Teubner Studienbücher, Stuttgart, 1984.

[2] M. Clausen and U. Baum, *Fast Fourier transforms*, Wissenschaftsverlag, Mannheim, 1993.

[3] ———, *Fast Fourier transforms for symmetric groups, theory and implementation*, Math. Comp. **61**(204) (1993), 833–847.

[4] M. Clausen, *Fast generalized Fourier transforms*, Theoret. Comput. Sci. **67** (1989), 55–63.

[5] ———, *Beiträge zum Entwurf schneller Spektraltransformationen*, Habilitationsschrift, Fakultät für Informatik der Universität Karlsruhe (TH), 1988.

[6] J. W. Cooley and J. W. Tukey, *An algorithm for machine calculation of complex Fourier series*, Math. Comp. **19** (1965), 297–301.

[7] P. Diaconis, *A generalization of spectral analysis with applications to ranked data*, Ann. Stat. **17** (1989), 949–979.

[8] ———, *Group representations in probability and statistics*, IMS, Hayward, CA, 1988.

[9] P. Diaconis and D. Rockmore, *Efficient computation of the Fourier transform on finite groups*, J. Amer. Math. Soc. **3**(2) (1990), 297–332.

[10] ———, *Efficient computation of isotypic projections for the symmetric group*, DIMACS Ser. Discrete Math. Theoret. Comput. Sci. **11**, L. Finkelstein and W. Kantor (eds.), 1993, 87–104.

[11] D. Elliott and K. Rao, *Fast transforms: algorithms, analyses, and applications*, Academic, New York, 1982.

[12] F. Goodman, P. de la Harpe, and V. Jones, *Coxeter graphs and towers of algebras*, Springer-Verlag, New York, 1989.

[13] G. James and A. Kerber, *The representation theory of the symmetric group*, Encyclopedia of Mathematics and its Applications, vol. 16, Addison-Wesley, Reading MA, 1981.

[14] S. Linton, G. Michler, and J. Olsson, *Fourier transforms with respect to monomial representations*, Math. Ann. **297** (1993), 253–268.

[15] I. Macdonald, *Symmetric functions and Hall polynomials*, Oxford University Press, 1979.

[16] D. Maslen, *Efficient computation of Fourier transforms on compact groups*, Max-Planck preprint MPI/95-8 (1995).

[17] ———, *The efficient computation of Fourier transforms on the symmetric group*, preprint IHES/M/96/52, and Math. Comp. (to appear).

[18] D. Maslen and D. Rockmore, *Generalized FFTs - a survey of some recent results*, DIMACS Ser. Discrete Math. Theoret. Comput. Sci., Groups and Computation, II, L. Finkelstein and W. Kantor (eds.), (1996).

[19] ———, *Separation of variables and the efficient computation of Fourier transforms on finite groups, I*, J. Amer. Math. Soc. **10** (1) (1997), 169–214.

[20] ———, *Separation of variables and the efficient computation of Fourier transforms on finite groups, II*, (preprint).

[21] D. Rockmore, *Applications of generalized FFTs*, DIMACS Ser. Discrete Math. Theoret. Comput. Sci., Groups and Computation, II, L. Finkelstein and W. Kantor (eds.), (1996).

[22] J. Serre, *Linear representations of finite groups*, Springer-Verlag, New York, 1977.

[23] R. Stanley, *Differential Posets*, J. Amer. Math. Soc. **1**(4) (1988), 919–961.

[24] R. Stanley, *Variations on differential posets*, in Invariant Theory and Tableaux (ed. D. Stanton), IMA Vol. Math. Appl. **19**, Springer, New York, 1990, 145–165.

[25] A. Vershik and S. Kerov, *Locally semisimple algebras. Combinatorial theory and the K_0 functor*, Itogi Nauki i Tekhniki, Sovremennye Problemy Matematiki, Noveishie Dostizheniya, Vol. 26 (1985), 3–56.

Department of Mathematics, Universiteit Utrecht, 3584 CD, Utrecht, Netherlands
E-mail address: `maslen@math.ruu.nl`