2002

Representations and the Symmetric Group

Elizabeth Norton
Harvey Mudd College

Data Compression on the Symmetric Group

by
Elizabeth Norton
Michael Orrison, Advisor

Advisor: ______________________________

Second Reader: ______________________________
(Shahriar Shahriari)

May 2002
Department of Mathematics

Abstract

Data Compression on the Symmetric Group

by Elizabeth Norton

May 2002

The regular representation of the symmetric group S_n is a vector space of dimension n! with many interesting invariant subspaces. The projections of a vector onto these subspaces may be computed by first considering projections onto certain basis elements in the subspace and then recombining later. If all of these projections are kept, it creates an explosion in the size of the data, making it difficult to store and work with. This is a study of techniques to compress this computed data such that it is of the same dimension as the original vector.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank my advisor, Michael Orrison, for his support and direction when working on this thesis. I would also like to thank Shariar Shariari for helping me learn the foundation for this work and his continuing support throughout the year. Additionally, I would like to thank my fourth grade teacher, Rita Fleming, without whose encouragement in my quest for knowledge I would not be here.

Chapter 1

Introduction

The study of representation theory is one way to merge the disciplines of abstract and linear algebra. It is the study of representation theory, briefly outlined in Chapter 2, that fueled the current research. The concepts expressed there, such as the construct of a module in Section 2.1, form one of the many areas of algebra.

Research, however, does not exist in a vacuum. It is only through the study of previous work on such things as the Jucys-Murphy elements of Section 3.3 and the seminormal basis defined in Section 3.4 that the current project crystallized.

At its heart, the work here is on data compression. The question addressed is "Is there redundancy? Can the data be stored without redundancy?" But knowing the question is not enough. There are also constraints on what the data actually is.

In this, the concern is not about where the data comes from. The assumption is that it exists and that there is enough of it to warrant the extra expenditure of the fixed cost to improve the efficiency of the computation before analysis. Building on the concepts in Chapter 2, the material in Chapter 3 gives the restraints on the structure of the data.

Between Chapter 3 and Chapter 4 the focus shifts. The space being examined is not original, but the study of it beginning in Chapter 4 is my own. Here the focus is on how I look at the concepts that have been presented prior in a way which illuminates them more clearly. My work has, unfortunately, lacked focus and there are two distinct ways in which I have turned my attention, the first being explored in

Sections 4.2 and 4.3 and the second, which seems to be more fruitful in the interests of understanding, appearing only in Section 4.4.

What I have found is presented in Chapter 5. To summarize, I have worked through some examples in Sections 5.1 to 5.3 and presented some conjectures in Section 5.4. Since my results are examples, the conclusion in Chapter 6 is rather short, but it summarizes where I think research should go from here.

Chapter 2

Background

Before presenting my research, I present this compact introduction to representation theory to help smooth the way. Section 2.1 is the heart of representation theory as it is presented here. Section 2.2 gives the construct within representation theory which is used the most in subsequent chapters. Section 2.3 and 2.4 present some definitions of terms which prove extremely useful in this study. Each of these will be used in the following chapter. For more information about representation theory, see [3].

2.1 Representation of a Group

Recall that a group G consists of a base set and an operation for combining two elements in the set to form a third. There are some constraints on the operation, to see these please see [3] or your favorite Abstract Algebra textbook. For a representation of a group, there is the additional requirement of a vector space V over a field F. This field is frequently the complex numbers, $\mathbb{C}$, and this is the field that I used throughout my research.

Definition 2.1.1 *Let V be a n-dimensional vector space over a field F. A* **representation** *of a group G is a function $\rho\colon G \to \mathrm{GL}(n, F)$, where $\mathrm{GL}(n, F)$ is the group of $n \times n$ invertible matrices over the field F, such that, for all g, h in G, $\rho(gh) = \rho(g)\rho(h)$ and $\rho(1) = I_n$, the identity matrix.*

This associates each element of the group with a $n \times n$ matrix, which may be thought of as an automorphism of V. For simplicity, we denote $v\rho(g)$ by vg when

$v \in V, g \in G$.

Definition 2.1.2 *A vector space V over a field F with an associated representation of the group G is called a* **module** *or an* **FG-module**.

Modules are important because they are the basic units of representation theory, and one such module is the group algebra.

2.2 Group Algebra

The group algebra is the set of all linear combinations of the group elements. A precise definition follows.

Definition 2.2.1 *The* **group algebra** *of a group G over a field F is a vector space V over F with dimension $|G|$ and basis elements $\{b_g\}$ indexed by the elements of the group, along with rules for addition and multiplication in the vector space. For μ_g and λ_g in the field F, addition of two vectors is given by*

$$\left(\sum_{g\in G} \lambda_g b_g\right) + \left(\sum_{g\in G} \mu_g b_g\right) = \sum_{g\in G} (\lambda_g + \mu_g) b_g \tag{2.1}$$

and multiplication of the same two elements would be

$$\left(\sum_{g\in G} \lambda_g b_g\right) \left(\sum_{h\in G} \mu_h b_h\right) = \left(\sum_{g,h\in G} \lambda_g \mu_h b_{gh}\right). \tag{2.2}$$

It is often easier to look at the basis elements as if they were the group elements themselves, but it is more accurate to think of them as symbols referencing the elements of the group. Since the group is not necessarily abelian, order does matter in multiplication in the group algebra. The group algebra for a group G over a field F is denoted FG. I have been working with $\mathbb{C}S_n$, where the $\mathbb{C}$ is the field of complex numbers and the group S_n is the symmetric group on n elements. For more information on the symmetric group S_n and the notation I am using, please see Section 3.1.

To illustrate this algebra better, consider the elements $v = (1+2i)[123] + (3)[213]$ and $w = 1[321]$ in $\mathbb{C}S_3$. In this case, $v + w = (1+2i)[123] + (3)[213] + 1[321]$ and $vw = (1+2i)[321] + (3)[312]$. Note that when an element in the group algebra is multiplied by an element in the group from the right, it simply switches the ordering by permuting the positions of the members of the original set $\{1, 2, \ldots, n\}$, rather than permuting the members themselves. (cf. Section 3.1)

2.3 Submodules and Reducibility

As an FG-module, the group algebra FG has some subspaces which are invariant with respect to the representation of the group. For example the one-dimensional subspace which is the span of the element of the group algebra which is the sum of one (the multiplicative identity in the field) times each of the basis elements, i.e.,

$$V = \operatorname{span}(\sum_{g \in G} g) \tag{2.3}$$

is invariant. It is invariant because multiplication by any element of the group results in merely a permutation of the group elements, and addition is commutative. This particular subspace is often referred to as the trivial subspace or the trivial representation. Formally,

Definition 2.3.1 *A subspace W of a module V is called a* **submodule** *if, for all $g \in G, w \in W$, $wg \in W$.*

Definition 2.3.2 *We say that an FG-module V is* **irreducible** *if the only invariant subspaces it contains are itself and the trivial space, $\{0\}$.*

Thus, the trivial subspace above is actually an irreducible submodule of the group algebra $\mathbb{C}S_n$. As another example, consider the group algebra of the symmetric group S_2. In this, there are two proper non-trivial submodules, $V = \operatorname{span}\{[12] + [21]\}$ and $W = \operatorname{span}\{[12] - [21]\}$. Each of these is irreducible. Again, for notation, please see Section 3.1.

2.4 Isomorphisms and Isotypics

Definition 2.4.1 *Two FG-modules V and W are* **isomorphic** *if there exists a bijection $\sigma\colon V \to W$ such that $\sigma(vg) = (\sigma(v))g$ for all $v \in V$.*

The two submodules of $\mathbb{C}S_2$ give in Section 2.3 are non-isomorphic. In $\mathbb{C}S_3$, there are three different non-isomorphic irreducible spaces.

Consider a set of two orthogonal irreducible submodules of $\mathbb{C}S_3$ which are isomorphic. One such set is

$$\begin{aligned} &\text{span}\{[123]+[213]-[321]-[231],[123]-[213]+[321]-[231]\} \quad \text{and} \\ &\text{span}\{[132]+[312]-[321]-[231],[132]-[312]-[321]+[321]\}. \end{aligned}$$

While another such set is

$$\begin{aligned} &\text{span}\{[123]+[213]-[321]-[231],[132]-[312]-[321]+[321]\} \quad \text{and} \\ &\text{span}\{[132]+[312]-[321]-[231],[123]-[213]+[321]-[231]\} \end{aligned}$$

which are also isomorphic to the first two submodules. There are, in fact, an infinite number of isomorphic submodules in the smallest module that contains all of these. Although the irreducible submodules themselves are not unique, this larger submodule, which could be considered as the sum of all irreducible submodules which are isomorphic (such as the two examples above) is always the same. Thus, it is more useful to look at this sum when the multiplicity of the irreducible is more than one.

Definition 2.4.2 *The smallest submodule of a module V which contains all submodules isomorphic to a given irreducible submodule is called an* **isotypic subspace***.*

Since there can be many nonisomorphic submodules, there are many isotypic subspaces, but for a single irreducible submodule the isotypic subspace containing it can be shown to be well-defined. See [3] for more information.

Chapter 3

Applications to the Symmetric Group

Building on the previous chapter, here we explore the specific applications to the symmetric group. Section 3.1 is a quick introduction to the symmetric group itself, including the notation used in this thesis. Section 3.2 is not obviously an application of representation theory to the symmetric group, but from the tableaux there defined some of the concepts of Sections 3.3 and 3.4 become more clear. The Jucys-Murphy elements in Section 3.3 have been at the heart of the research throughout, helping with computation regarding the seminormal basis in Section 3.4.

3.1 Symmetric Group

One fundamental group in group theory is the symmetric group on n elements.

Definition 3.1.1 *The* **symmetric group***, denoted S_n, is the set of all permutations of the set $\{1, 2, \dots, n\}$. Two permutations α_1, α_2 are multiplied to form the element $\alpha_1\alpha_2$ which is the permutation obtained by performing α_1 followed by α_2.*

There are $n!$ elements in this group, corresponding to the different orderings of the set $1, \dots, n$. There are many different notations for the elements in this group. For example, consider S_3. One way to permute the set $1, 2, 3$ is to map 1 to 2 and 2 back to 1. This can be denoted by the 2×3 matrix

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}$$

or in cycle notation as (12). In the matrix representation, the elements in the set are listed (in some order) in the top row, and each one is mapped to the element listed directly underneath it. For general S_n, this becomes a $2 \times n$ matrix. Cycle notation is a much more compact form for writing the elements. In cycle notation, each element is mapped to the next element to the right, unless there is a right parenthesis. In this case, it gets mapped back to the first element in the parenthetical grouping. If an element is not listed, as 3 is not in the permutation (12), it is mapped to itself. Take the element (13)(264) in S_6. This could also be written as

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 6 & 1 & 2 & 5 & 4 \end{pmatrix}.$$

The problem with cycle notation is that the elements are not read in the original order. The problem with the matrix notation is that it's too large. In this report, I used a different notation, which assumes the first row of the matrix is simply $1 \dots n$ and writes the second row. This way, the elements above would be [213] and [361254], respectively. The elements of S_3 would then be $[123], [213], [132], [312], [321], [231]$.

3.2 Standard Tableaux

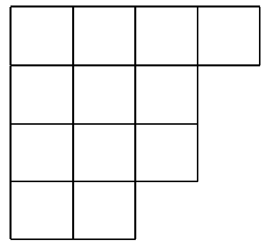
Before seeing standard tableaux, it is necessary to understand the partition of an integer and the Ferrers diagram of that partition.

Definition 3.2.1 *A* **partition** *of the integer $n > 0$ is a non-increasing list $(\lambda_1, \lambda_2, \dots, \lambda_k)$ of positive integers with $\sum_{i=1}^{k} \lambda_i = n$.*

Definition 3.2.2 *For a partition $(\lambda_1, \lambda_2, \dots, \lambda_k)$ of n the* **Ferrers diagram** *is a table of n left-aligned boxes with λ_i boxes in the i^{th} row and a total of k rows.*

Definition 3.2.3 *A* **standard tableaux** *of shape λ is a filling of the Ferrers diagram of the partition λ with the numbers in the set $\{1, 2, \dots, n\}$ such that there is an increasing sequence across each row and down each column.*

To illustrate, consider $\lambda = (4, 3, 3, 2)$. The Ferrers diagram would be



while a standard tableau would be

1	3	4	7
2	5	9	
6	8	11	
10	12		

.

It is easy to see that there are many standard tableaux of shape λ. For example, another standard tableaux of the same shape above could be obtained by switching the numbers 7 and 5.

Definition 3.2.4 *The* **content** *of a box b in a given standard tableau is defined as* $\mathrm{ct}(b) = j - i$ *if the box b is in the i^{th} row and the j^{th} column of the tableau.*

For example, $\mathrm{ct}(5) = 0$ in the standard tableaux above. Content is useful when dealing with multiple standard tableaux of the same shape. Applications of standard tableaux and their content will be explored in Sections 3.3 and 3.4.

3.3 Jucys-Murphy Elements

The Jucys-Murphy(JM) elements are specific elements of the group algebra $\mathbb{C}S_n$.

Definition 3.3.1 *The* **Jucys-Murphy(JM) element** $R_j \in \mathbb{C}S_n$, $n \geq j \geq 2$, *is given by*

$$R_j = (1, j) + (2, j) + \cdots + (j - 1, j) \tag{3.1}$$

where (i, j) represents the transposition in which i is mapped to j and vice versa, written in the cycle notation described in Section 3.1. [7]

In an algebra V, the mapping $\sigma\colon V \to V$ defined by $\sigma(v) = vw$ for any $v \in V$ and a given $w \in V$ gives a linear transformation of V. One way to write linear transformations is as matrices by picking a basis, then writing the image of the i^{th} basis element as a column vector in the i^{th} column of the matrix. For example, the image of [132] under the JM-element R_2 is [312]. One way to order S_3 is as $[123], [213], [132], [312], [321], [231]$. In this ordering, [132] is the 3^{rd} element and [312] is the 4^{th} element, so the third column of the matrix representation of R_2 should be the column vector $(0, 0, 0, 1, 0, 0)^T$.

For S_3, the JM-elements are

$$R_2 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \qquad R_3 = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}. \tag{3.2}$$

If a subspace is invariant with respect to the representation of the group, it must be invariant under transformation by any of these elements. If a subspace is invariant under transformation by all JM-elements, then it must be invariant under transformation by any linear combination of these elements.

Since I have defined a real, symmetric matrix, it is diagonalizable, and thus helpful to find the eigenvalues of the matrix.

Theorem 3.3.2 *(See [7], although this was not first proved there) For the JM-element R_k in $\mathbb{C}S_n$ the eigenvalues are all possible contents of the box k in standard tableaux of any partition λ of n.*

3.4 Seminormal Basis

The group S_n has, as a subgroup, the group S_{n-1} which are the elements that keep n fixed and only permute the first $n-1$ numbers. In this way, we can make the chain $\{1\} = S_1 \leq S_2 \leq \cdots \leq S_{n-1} \leq S_n$. For any given irreducible module V of the group S_n, this module can be decomposed into irreducible submodules when restricted to S_{n-1}. This process of restriction can be continued until the trivial group is reached.

Definition 3.4.1 *A* **seminormal basis** *of a module V of the group S_n is a basis in which, if V is decomposed into irreducible submodules as an $S_k,\ k \leq n$ module, there is a partition of the basis such that each of the blocks forms a basis for one of the irreducibles.*

What does this mean? Recall from abstract algebra that the conjugacy class of an element $g \in G$ is the set $\{h^{-1}gh | h \in G\}$. In [3], it is shown that there are the same number of non-isomorphic irreducible $\mathbb{C}S_n$-modules as there are conjugacy classes in S_n. Also, the conjugacy classes of the symmetric group would be elements of the same cycle shape. A given cycle shape can be considered a partition of the integer n. Thus, there is a correspondence between the unique (up to isomorphism) irreducible $\mathbb{C}S_n$-modules and partitions λ of n.

If the right correspondence is chosen, the irreducible module has a basis that can be indexed by the standard tableaux of shape λ. This can be shown to lead to a seminormal basis. Moreover, each of these basis vectors is an eigenvector for each R_k. Specifically,

$$R_k b = \mathrm{ct}(k) b \tag{3.3}$$

$\mathrm{ct}(k)$ is the content of the box k in the standard tableaux associated with the basis vector b. This is an application of Theorem 3.3.2 above.

By projecting onto the eigenspaces of the JM-elements, the process of restriction in Definition 3.4.1 gives rise to a method of computing projections onto the smallest members, the basis elements themselves. Unfortunately, since the irreducible modules can only be identified with the partitions up to isomorphism, the projection ends up being a projection onto equivalence classes of the basis vectors, which is why in Appendix A there are vectors with more than one independent variable. Fortunately, I am concerned with the projections onto the isotypic subspaces (see Section 2.4), which include all copies of all the isomorphic irreducibles.

Chapter 4

Methods

Up to this point, I have focused on the mathematical concepts which I used in my research. Pulling together concepts from Chapters 2 and 3, here is the actual work that I have done.

4.1 Data Compression

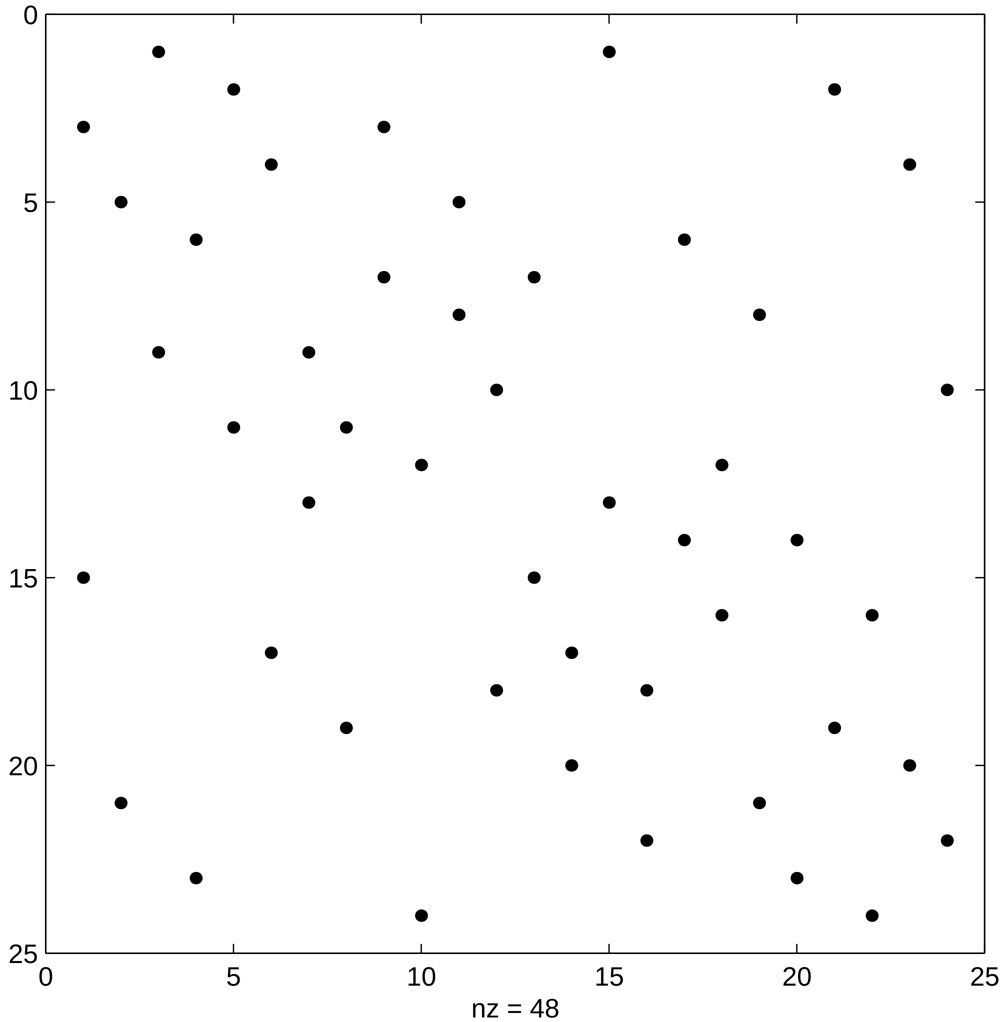
My goal with my research is, as stated in Chapter 1, to attempt to store data without redundancy. The data I want to store are the projections onto the isotypic subspaces. Thus, knowledge of the structure of the isotypic subspace would allow for that compression. The projections I have actually computed are the projections onto equivalence classes of seminormal basis elements. Since the projection onto an isotypic subspace is the sum of the projections onto the corresponding seminormal basis elements, storing these projections is just as good.

For $\mathbb{C}S_3$, these projections are listed in Table 5.1. For $\mathbb{C}S_4$, these projections are in Appendix A.

4.2 Element Order

When there isn't a truly natural ordering, mathematicians tend to pick one. One ordering for the elements of S_n, especially because of the way I have chosen to notate them, is lexicographic. Yet, in lexicographic order, there is very little structure on the elements of which I can take advantage. For example, in lexicographic order the element R_3 of S_4 looks like Figure 4.1(nz represents the number of nonzero elements),

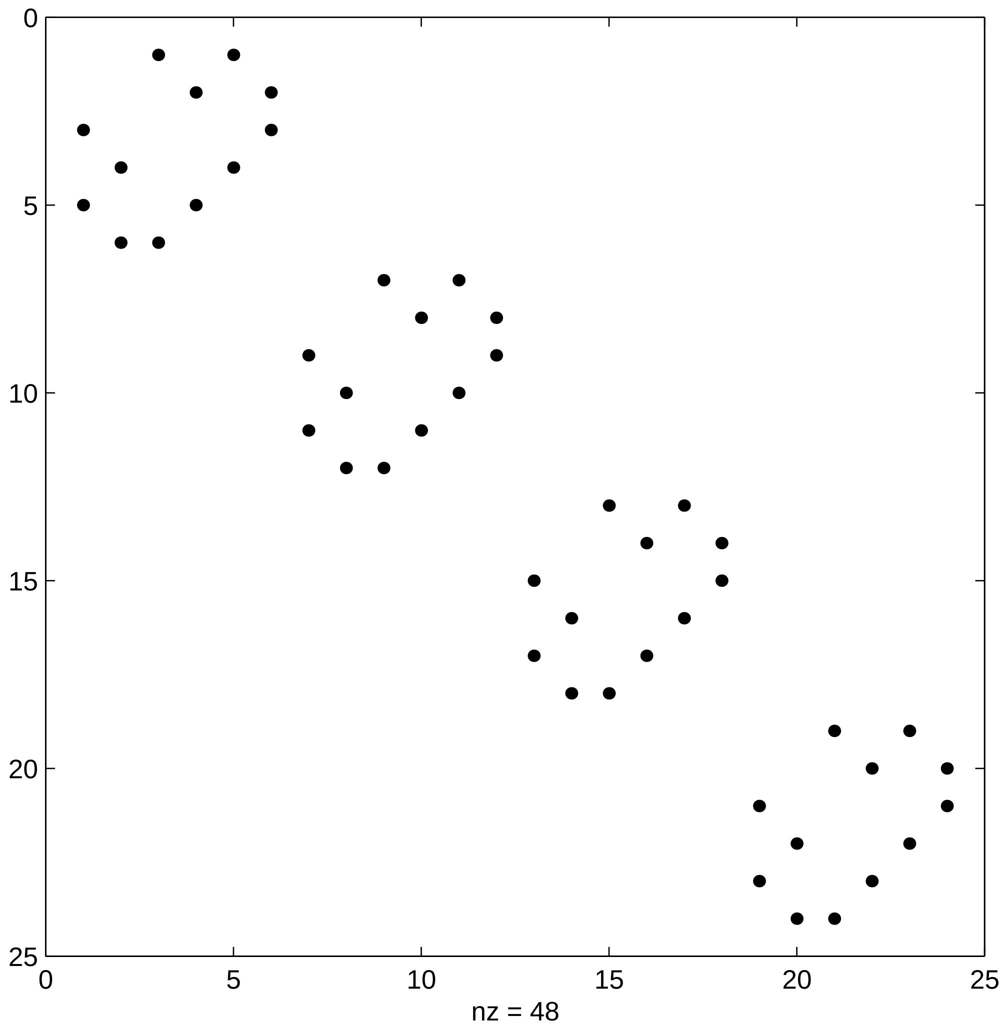
Figure 4.1: Lexicographic Ordering of R_3 in S_4



but I would prefer to see it in the order which gives Figure 4.2. Note that, in these figures, the black dots represent where there are non-zero entries in the matrix.

In Section 3.4, I noted that there is a natural way in which $\{1\} = S_1 \leq S_2 \leq \cdots \leq S_n$. To create the ordering for Figure 4.2, I took advantage of this structure in the group. Beginning with the identity permutation, I focused first on the permutations that fixed the last $n-1$ elements, which is just the identity, and then the ones where the last $n-2$ elements are fixed. Other than the identity, there is one with this

Figure 4.2: R_3 in S_4 with my ordering



property, the transposition (12), so this transposition is the second element in my ordering.

When transferring to elements that exist in S_3, I had to bring in four more elements. To choose the ordering of these, I began by looking at the ones where 3 gets mapped to 2 and then where 3 gets mapped to 1. In this way, I create blocks within the list of basis elements. As illustrated in Figure 4.2, which has 4 blocks, the interaction within one block is exactly the same as the interaction in another.

Table 4.1: Ordering on S_4

1	2	3	4
2	1	3	4
1	3	2	4
3	1	2	4
3	2	1	4
2	3	1	4
1	2	4	3
2	1	4	3
1	4	2	3
4	1	2	3
4	2	1	3
2	4	1	3
1	4	3	2
4	1	3	2
1	3	4	2
3	1	4	2
3	4	1	2
4	3	1	2
4	2	3	1
2	4	3	1
4	3	2	1
3	4	2	1
3	2	4	1
2	3	4	1

Interaction between blocks, which occurs in Figure 4.2 if the blocks considered are of two elements rather than six, is not as constrained and must be computed

For example, since the ordering on S_2 is [12] followed by [21], the second block of the ordering on S_3 would be [132] followed by [312]. This generates the ordering of S_3 I have been using since Section 3.1 and the ordering of S_4 as given in Table 4.2. To generate this example, I used the MatLab program in Appendix B.1.

With this ordering, I need only compute the structure of the JM-element R_k in the group algebra $\mathbb{C}S_k$ in order to know it's structure in the group algebra $\mathbb{C}S_n,\ n \geq k$. Specifically, R_k is a block diagonal matrix with $\frac{n!}{k!}$ blocks of size $k! \times k!$, each of which is the $k! \times k!$ matrix of R_k in $\mathbb{C}S_k$. This is the specific attribute of the ordering that I wanted to have, so I generated my order from it.

4.3 Eigenspaces

Recall from Theorem 3.3.2 that the eigenvalues of the JM-elements are all integers. Specifically, the values which can be obtained are dictated by the standard tableaux for the integer n. From Section 3.4, if a seminormal basis is chosen, subsets of the basis will form a basis for irreducible submodules when the group algebra $\mathbb{C}S_n$ is viewed as a $\mathbb{C}S_k$-module. Although when dealing with action from the right it is not possible to break down the space to the seminormal basis, it is possible to break it into subspaces which are acted on by the JM-elements in the same manner. This is done by identifying the eigenvectors which form a single standard tableau. For example, if the standard tableau is

1	2	4
3		

then I wish to find an eigenvector with eigenvalue 1 for R_2, -1 for R_3 and 2 for R_4. The structure of these vectors (for there will be three independent vectors of these sort) is given in the second column on page 31.

In computing the structure of the seminormal basis elements, it is my hope to improve the computational efficiency of algorithms (see [6]) to calculate the projections onto the isotypic subspaces, specifically algorithms that use the seminormal basis, by only having to compute certain components of the projection of the vector.

4.4 Compressing Matrices

R_2 can act as a linear transformation in $\mathbb{C}S_6$, with the only difference being the dimension of the algebra. The eigenvalues of R_2 remain $\{-1, 1\}$, but the multiplicity of each eigenvalue increases. Once the eigenspaces of R_2 are identified, repeating that work to identify that these eigenspaces are superspaces of the eigenspaces of R_3 is rather pointless, so it is useful to identify the redundancy introduced by the knowledge that the isotypic we are looking for, at least part of it, lives in an eigenspace of which the structure is known. This can be used to eliminate unnecessary parts of the matrix representation of R_j.

To make this elimination, because the matrices R_i are simultaneously diagonalizable, use the eigenvectors of R_{j-1}, with vectors corresponding to the same eigenvalue grouped together, as a change of basis on R_j. For examples, see Section 5.3, specifically (5.2).

Also, because of the block structure of the imposed ordering of the basis elements, when $j < n$ we know that the JM-element R_j will only act within the blocks, as can be seen by the block diagonal structure of R_3 in Figure 4.2. Once it is known how R_j acts in $\mathbb{C}S_j$, this can be extended to each of the blocks of size $j!$ of the basis elements of $\mathbb{C}S_n$. In this way, only the action of R_n need be considered.

Chapter 5

Results

Much of what I have obtained cannot be easily shown in a chart, but some can. I have summarized here what I can. This is presented in chronological order, beginning with my first attempts at identifying a pattern, computing by hand everything in Section 5.1. The patterns I did see have been summarized in Section 5.2. Results from the more recent study as described in Section 4.4 are given in Section 5.3. Conjectures, which are possible future areas of research, are given in Section 5.4.

5.1 Early Examples: S_3

For R_2, the possible eigenvalues are plus and minus one. For R_3, the possible eigenvalues are $\{-2, -1, 1, 2\}$, but none of these are possible with both of the eigenvalues for R_2. Table 5.1 summarizes the compression obtained by projecting onto the semi-normal basis elements of $\mathbb{C}S_3$.

Similar to these results, those for $\mathbb{C}S_4$ are included as Appendix A. These are the major results of this research.

5.2 Observations

Before even considering the compressed matrices discussed in Section 4.4, there are some observations that can be made. One observation is that, because of the chosen ordering on the elements, once I have reduced S_{n-1}, this reduction carries through to each of the n blocks of S_n. For example, note the compression in $\mathbb{C}S_3$ corresponding to the eigenvalues 1 and 2 of R_2 and R_3, respectively. This same compression occurs

Table 5.1: Compression in the Projections of S_3

Eigenvalue for R_2:	1		-1	
Eigenvalue for R_3:	2	-1	1	-2
Compression of vector:	x_1	x_2	x_4	x_6
	x_1	x_2	$-x_4$	$-x_6$
	x_1	x_3	x_5	x_6
	x_1	x_3	$-x_5$	$-x_6$
	x_1	$-(x_2+x_3)$	x_4-x_5	$-x_6$
	x_1	$-(x_2+x_3)$	$-(x_4-x_5)$	x_6

in each of the 4 blocks in the compression of $\mathbb{C}S_4$ corresponding to these same eigenvalues. This can be seen on page 29 in the first column and on page 31 in the first column. Also, the first block of $(n-1)!$ basis elements is easily seen to be entirely redundant. Note that, since $R_2, \ldots, R_{n-1}$ each only act on the elements in the 1^{st} through $(n-1)^{th}$ positions, they can only permute within blocks, as evidenced by the block diagonal structure of the matrices representing these elements. Thus, only R_n can now compress the data. R_n is the sum of $n-1$ transpositions, so multiplying a basis element in the module $\mathbb{C}S_n$ by R_n results in the sum of $n-1$ new elements. But the structure of the ordering helps to identify these elements, because each transposition exchanges the number 4 with another number. Because of the structure of the ordering chosen, the elements to which an element in the first block maps are the elements in the other blocks in exactly the same position as the element being acted upon.

For example, the element $[1324]R_4 = [4321] + [1423] + [1342]$ in $\mathbb{C}S_4$. $[1324]$ is the 3^{rd} element in the first block, and each of the new elements is the 3^{rd} element in its respective block. Note that in Table 4.2 the blocks are divided by horizontal lines.

5.3 *Reduced Matrices*

The matrices representing the JM-elements up to R_4 have been reduced as explained in Section 4.4. R_5 is in progress.

5.3.1 R_2

It is always useful to start with the most basic cases, just to make sure that one knows what is going on. Thus, the first matrix is for the linear transformation defined by $\rho\colon \mathbb{C}S_2 \to \mathbb{C}S_2$ where $\rho(x) = R_2 x$ and R_2 is the transposition (21). This matrix is

$$R_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{5.1}$$

Since this is the first element, I have no additional information, so it cannot be reduced, but it provides information that will be used in the later matrices. The eigenvalues of this matrix are $\{1, -1\}$ corresponding to the eigenvectors $\left(\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}\right)$ and $\left(\begin{smallmatrix} 1 \\ -1 \end{smallmatrix}\right)$. Knowing that R_2 in S_n is going to be a block diagonal matrix with $(n!)/2$ copies of (5.1) as the blocks along the diagonal, shows that these eigenvectors will be similar to the previous eigenvectors. Specifically, these eigenvectors are one of the above eigenvectors in a position corresponding to one of the blocks and zeros everywhere else. For example, refer to (3.2) to see the matrix representation of R_2 acting on $\mathbb{C}S_3$. The eigenvalues of this matrix are $\{-1, -1, -1, 1, 1, 1\}$ with corresponding eigenvectors

$$\begin{pmatrix} 1 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \\ -1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

5.3.2 R_3

The matrix representation of R_3 acting on $\mathbb{C}S_3$ is given in (3.2). This matrix has eigenvalues $\{-2,-1,-1,1,1,2\}$. But when we only look at how it acts in the eigenspaces of R_2, we know more about the action. Considering the eigenvectors above, it is easy to see that we only need to know where the first, third, and fifth basis vectors map in order to have complete information, but that these may map to different places depending upon which eigenspace we look at. So, when in the eigenspace corresponding to the eigenvalue -1, the reduced matrix R_3 is

$$R_3[-1] = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & -1 \\ 1 & -1 & 0 \end{bmatrix}$$

and the reduced matrix corresponding to the eigenvalue 1 is

$$R_3[1] = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}.$$

These matrices were obtained by using the eigenvectors above as a new basis for the matrix. To see this, note that they are obtained as the blocks of the block diagonal matrix R_3 after the change of basis by the eigenvectors of R_2 as follows:

$$\begin{bmatrix} 0&0&1&0&1&0 \\ 0&0&0&1&0&1 \\ 1&0&0&0&0&1 \\ 0&1&0&0&1&0 \\ 1&0&0&1&0&0 \\ 0&1&1&0&0&0 \end{bmatrix} = \begin{bmatrix} 1&0&0&1&0&0 \\ 1&0&0&-1&0&0 \\ 0&1&0&0&1&0 \\ 0&1&0&0&-1&0 \\ 0&0&1&0&0&1 \\ 0&0&1&0&0&-1 \end{bmatrix} \begin{bmatrix} 0&1&1&0&0&0 \\ 1&0&1&0&0&0 \\ 1&1&0&0&0&0 \\ 0&0&0&0&1&1 \\ 0&0&0&1&0&-1 \\ 0&0&0&1&-1&0 \end{bmatrix} \begin{bmatrix} 1&0&0&1&0&0 \\ 1&0&0&-1&0&0 \\ 0&1&0&0&1&0 \\ 0&1&0&0&-1&0 \\ 0&0&1&0&0&1 \\ 0&0&1&0&0&-1 \end{bmatrix}^{-1} \tag{5.2}$$

Since these are smaller matrices, they have fewer eigenvalues. These are, for $R_3[-1]$, $\{-2,1,1\}$ and for $R_3[1]$, $\{-1,-1,2\}$. The multiplicity of the eigenvalue 1 means that the eigenvectors will not be unique. One choice of eigenvectors is

$$\begin{pmatrix} 1 \\ -1 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}$$

for $R_3[-1]$ and, for $R_3[1]$,

$$\begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

5.3.3 R_4

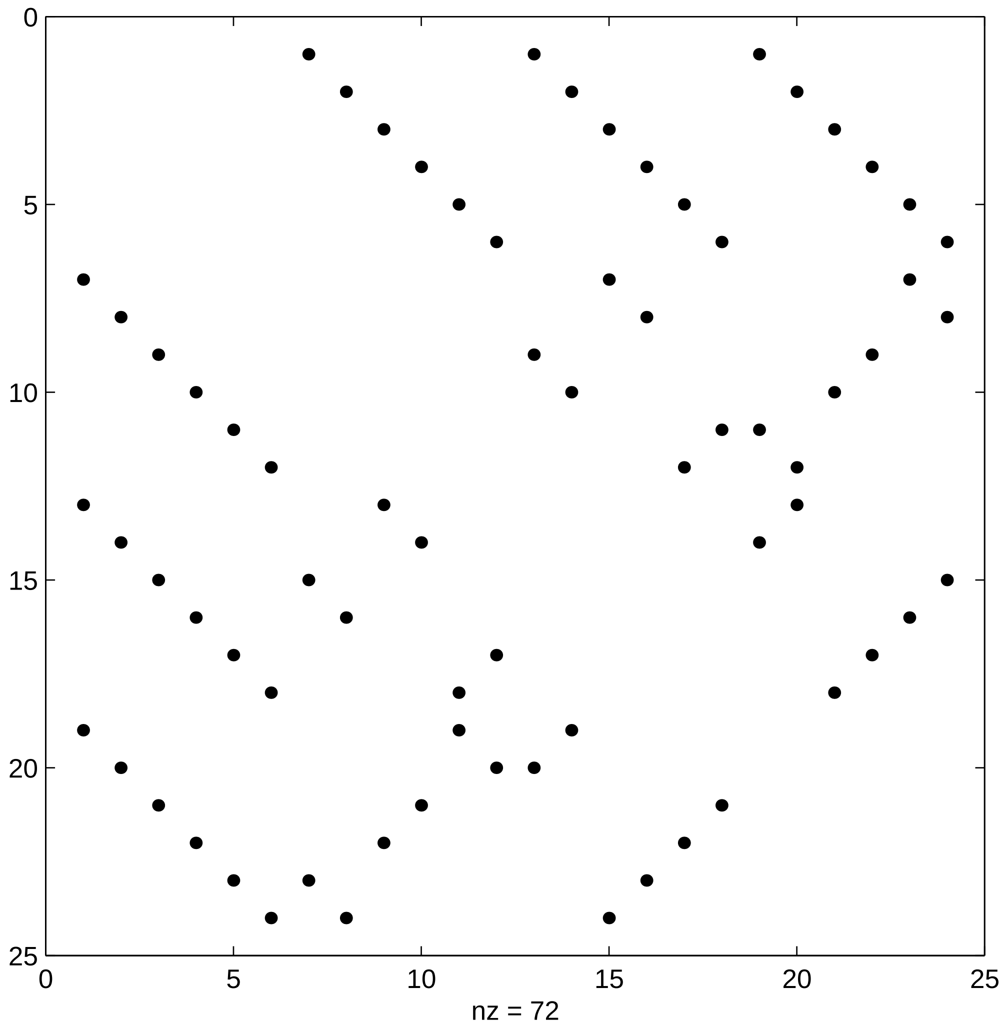
In R_4, the matrices start getting larger, so I will only list the matrices themselves and their eigenvalues. The original, 24×24 matrix, appears in Figure 5.1

A final note before I present the matrices: labelling these turned somewhat difficult, so I have opted for the label format of $R_n[\lambda_2, \dots, \lambda_{n-1}]$ where this is the reduction of the matrix representation of R_n living in the eigenspaces for R_j corresponding to the eigenvalue λ_j where $j = 2, \dots, n-1$.

$$R_4[1,2] = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}, \text{ eigenvalues: } \{-1,-1,-1,3\}$$

$$R_4[-1,-2] = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & -1 & -1 \\ 1 & -1 & 0 & -1 \\ 1 & -1 & -1 & 0 \end{bmatrix}, \text{ eigenvalues: } \{-3,1,1,1\}$$

Figure 5.1: R_4 in S_4



$$R_4[1,-1] = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & -1 & -1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & -1 & -1 \\ 1 & 0 & -1 & -1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & -1 & -1 & 0 & 0 \end{bmatrix}, \text{ eigenvalues: } \{-2,-2,-2,0,0,2,2,2\}$$

$$R_4[-1,1] = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & -1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & -1 \\ 1 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & -1 & 1 \\ 1 & 0 & 1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & -1 & 1 & 0 & 0 \end{bmatrix}, \text{ eigenvalues: } \{-2,-2,-2,0,0,2,2,2\}$$

Proving that these are the correct matrices would involve a similar change of basis as in (5.2) performed on the matrix representation of R_4 followed by a similar change of basis corresponding to the eigenvectors of R_3. After these two basis changes, the above matrices would be blocks along the diagonal.

5.4 Conjectures

Examination of these matrices has led to three conjectures, one about the size of the compressed matrix, which is not obvious, and two about the actually matrices.

Conjecture 1 *The compressed matrix $R_n[\lambda_2, \dots, \lambda_{n-1}]$ is an $nk \times nk$ matrix, where k is the multiplicity of the eigenvalue λ_{n-1} in the compressed matrix $R_{n-1}[\lambda_2, \dots, \lambda_{n-2}]$.*

I would recommend induction as an attempt to prove this.

Conjecture 2 *The compressed matrix* $R_n[1,2,\ldots,n-2]$ *exists, and is the* $n \times n$ *matrix corresponding to the adjacency matrix on the complete graph on* n *vertices, i.e., the matrix* $A = [a_{ij}]$ *where*

$$\begin{aligned} a_{ii} &= 0 \\ a_{ij} &= 1,\ i \neq j \end{aligned}$$

Given a proof of the previous conjecture, to prove the size of this matrix would involve a proof that the multiplicity of the eigenvalue $n-2$ in the matrix $R_{n-1}[1,2,\ldots,n-3]$ is 1. Again, if this is done by induction, this can be assumed to be the adjacency of the complete graph on $n-1$ vertices, and results have been obtained for such graphs. After this, the same basis change as in (5.2) would justify the result.

Conjecture 3 *The compressed matrix* $R_j[-1,-2,\ldots,-(j-2)]$ *exists, and is the* $n \times n$ *matrix* $A = [a_{ij}]$ *where*

$$\begin{aligned} a_{ii} &= 0 \\ a_{1j} &= 1,\ j \neq 1 \\ a_{i1} &= 1,\ i \neq 1 \\ a_{ij} &= -1,\ i,j \neq 1, i \neq j \end{aligned}$$

This could be proved much like Conjecture 2, but would involve more leaps of logic.

Chapter 6

Conclusion

The next step in the work is to prove the three conjectures and find the compressed matrices for R_5. If Conjecture 1 is true, then the sizes of these matrices will be

$R_5[1,2,3]$	5×5
$R_5[1,2,-1]$	15×15
$R_5[1,-1,2]$	15×15
$R_5[1,-1,0]$	10×10
$R_5[1,-1,-2]$	15×15
$R_5[-1,1,2]$	15×15
$R_5[-1,1,0]$	10×10
$R_5[-1,1,-2]$	15×15
$R_5[-1,-2,1]$	15×15
$R_5[-1,-2,-3]$	5×5

So far, compression seems to be very possible.

Appendix A

S_4 Compression

Using the ordering showed in Table 4.2, this is the compression for the group algebra $\mathbb{C}S_4$ in each of the eigenspaces for the JM-elements R_2, R_3, R_4. In each case the intersection of the eigenspaces is listed, and the eigenvalues are listed in the order corresponding to R_4, R_3, R_2, respectively. Eigenspaces corresponding to the same isotypic subspaces are listed on the same page, but the two spaces on the first page are in different isotypic subspaces.

At the bottom of each column, I've placed the succinct version of the results, showing which coefficients need to be calculated and stored, as well as which elements they correspond to. The notation for the elements is defined in Section 3.1.

eigenvalues:	$3, 2, 1$	$-3, -2, -1$
vector:	$\begin{pmatrix} x_1 \\ x_1 \end{pmatrix}$	$\begin{pmatrix} x_2 \\ -x_2 \\ -x_2 \\ x_2 \\ -x_2 \\ x_2 \\ -x_2 \\ x_2 \\ x_2 \\ -x_2 \\ x_2 \\ -x_2 \\ -x_2 \\ x_2 \\ x_2 \\ -x_2 \\ x_2 \\ -x_2 \\ -x_2 \\ x_2 \\ x_2 \\ -x_2 \\ x_2 \\ -x_2 \end{pmatrix}$
Need to store (numerical index):	1	1
corresponding elements:	e	e

$$
\begin{array}{ccc}
1,-2,-1 & -2,-1,1 & -2,1,-1 \\
\begin{pmatrix} x_3 \\ -x_3 \\ -x_3 \\ x_3 \\ -x_3 \\ x_3 \\ x_4 \\ -x_4 \\ -x_4 \\ x_4 \\ -x_4 \\ x_4 \\ x_5 \\ -x_5 \\ -x_5 \\ x_5 \\ -x_5 \\ x_5 \\ x_3-x_4-x_5 \\ -x_3+x_4+x_5 \\ -x_3+x_4+x_5 \\ x_3-x_4-x_5 \\ -x_3+x_4+x_5 \\ x_3-x_4-x_5 \end{pmatrix} &
\begin{pmatrix} x_6 \\ -x_6 \\ x_7 \\ -x_7 \\ x_6-x_7 \\ -x_6+x_7 \\ -x_6 \\ x_6 \\ -x_7-x_8 \\ x_7+x_8 \\ -x_6+x_7+x_8 \\ x_6-x_7-x_8 \\ x_7+x_8 \\ -x_7-x_8 \\ -x_7 \\ x_7 \\ 2x_7+x_8 \\ -2x_7-x_8 \\ -x_6-x_7-x_8 \\ x_6+x_7+x_8 \\ x_8 \\ -x_8 \\ -x_6-x_7 \\ x_6+x_7 \end{pmatrix} &
\begin{pmatrix} x_9 \\ x_9 \\ -2x_9+3x_{10}-3x_{11} \\ -2x_9+3x_{10}-3x_{11} \\ x_9-3x_{10}+3x_{11} \\ x_9-3x_{10}+3x_{11} \\ -3x_9+4x_{10}-6x_{11} \\ -3x_9+4x_{10}-6x_{11} \\ x_{10} \\ x_{10} \\ 3x_9-5x_{10}+6x_{11} \\ 3x_9-5x_{10}+6x_{11} \\ 2x_9-x_{10}+2x_{11} \\ 2x_9-x_{10}+2x_{11} \\ -4x_9+5x_{10}-7x_{11} \\ -4x_9+5x_{10}-7x_{11} \\ 2x_9-4x_{10}+5x_{11} \\ 2x_9-4x_{10}+5x_{11} \\ 3x_9-3x_{10}+4x_{11} \\ 3x_9-3x_{10}+4x_{11} \\ x_{11} \\ x_{11} \\ -3x_9+3x_{10}-5x_{11} \\ -3x_9+3x_{10}-5x_{11} \end{pmatrix} \\
1,7,13 & 1,3,21 & 1,9,21 \\
e,(1243),(1342) & e,(1324),(4321) & e,(1423),(4321)
\end{array}
$$

$$\begin{array}{ccc}
-1,2,1 & 2,-1,1 & 2,1,-1 \\
\begin{pmatrix}
x_{12} \\ x_{12} \\ x_{12} \\ x_{12} \\ x_{12} \\ x_{12} \\
x_{13} \\ x_{13} \\ x_{13} \\ x_{13} \\ x_{13} \\ x_{13} \\
x_{14} \\ x_{14} \\ x_{14} \\ x_{14} \\ x_{14} \\ x_{14} \\
-(x_{12}+x_{13}+x_{14}) \\ -(x_{12}+x_{13}+x_{14}) \\ -(x_{12}+x_{13}+x_{14}) \\ -(x_{12}+x_{13}+x_{14}) \\ -(x_{12}+x_{13}+x_{14}) \\ -(x_{12}+x_{13}+x_{14})
\end{pmatrix} &
\begin{pmatrix}
2x_{17}+2x_{16}-x_{15} \\ 2x_{17}+2x_{16}-x_{15} \\
5x_{16}+2x_{17}-4x_{15} \\ 5x_{16}+2x_{17}-4x_{15} \\
-7x_{16}-4x_{17}+5x_{15} \\ -7x_{16}-4x_{17}+5x_{15} \\
x_{15} \\ x_{15} \\
3x_{17}+6x_{16}-5x_{15} \\ 3x_{17}+6x_{16}-5x_{15} \\
-6x_{16}-3x_{17}+4x_{15} \\ -6x_{16}-3x_{17}+4x_{15} \\
4x_{16}+3x_{17}-3x_{15} \\ 4x_{16}+3x_{17}-3x_{15} \\
x_{16} \\ x_{16} \\
-5x_{16}-3x_{17}+3x_{15} \\ -5x_{16}-3x_{17}+3x_{15} \\
x_{17} \\ x_{17} \\
-3x_{15}+3x_{16}+x_{17} \\ -3x_{15}+3x_{16}+x_{17} \\
-2x_{17}-3x_{16}+3x_{15} \\ -2x_{17}-3x_{16}+3x_{15}
\end{pmatrix} &
\begin{pmatrix}
x_{18} \\ -x_{18} \\ -x_{20} \\ x_{20} \\ x_{18}+x_{20} \\ -x_{18}-x_{20} \\
x_{18} \\ -x_{18} \\ x_{19} \\ -x_{19} \\ x_{18}-x_{19} \\ -x_{18}+x_{19} \\
x_{19} \\ -x_{19} \\ -x_{20} \\ x_{20} \\ x_{19}+x_{20} \\ -x_{19}-x_{20} \\
x_{18}-x_{19} \\ -x_{18}+x_{19} \\ -x_{19}-x_{20} \\ x_{19}+x_{20} \\ x_{18}+x_{20} \\ -x_{18}-x_{20}
\end{pmatrix} \\
1,7,13 & 8,15,20 & 1,13,16 \\
e,(1243),(1342) & (2143),(1342),(2431) & e,(1432),(3142)
\end{array}$$

$$
\begin{array}{cc}
0,-1,1 & 0,1,-1 \\
\begin{pmatrix}
x_{21} \\ x_{21} \\ x_{22} \\ x_{22} \\ -(x_{21}+x_{22}) \\ -(x_{21}+x_{22}) \\ x_{21} \\ x_{21} \\ -(x_{21}+x_{22}) \\ -(x_{21}+x_{22}) \\ x_{22} \\ x_{22} \\ -(x_{21}+x_{22}) \\ -(x_{21}+x_{22}) \\ x_{22} \\ x_{22} \\ x_{21} \\ x_{21} \\ x_{22} \\ x_{22} \\ x_{21} \\ x_{21} \\ -(x_{21}+x_{22}) \\ -(x_{21}+x_{22})
\end{pmatrix}
&
\begin{pmatrix}
x_{23} \\ -x_{23} \\ x_{23}+x_{24} \\ -(x_{23}+x_{24}) \\ -x_{24} \\ x_{24} \\ -x_{23} \\ x_{23} \\ x_{24} \\ -x_{24} \\ -(x_{23}+x_{24}) \\ x_{23}+x_{24} \\ -x_{24} \\ x_{24} \\ -(x_{23}+x_{24}) \\ x_{23}+x_{24} \\ x_{23} \\ -x_{23} \\ x_{23}+x_{24} \\ -(x_{23}+x_{24}) \\ x_{23} \\ -x_{23} \\ x_{24} \\ -x_{24}
\end{pmatrix} \\
1,3 & 1,6 \\
e,(1324) & e,(2314)
\end{array}
$$

Appendix B

MatLab Programs

B.1 Generating the Group Elements

This is the MatLab code for generating the elements of the group in the ordering defined in Section 4.2.

```
function e = element(numb, dim)
     % ELEMENT produces the ordering of the
     % numb th element of a specific ordering of S(dim).

if dim <= 1,
  e = [1];
  % if the dimension is 1, then this must be the only element, ie, [1].
else
  fact = factorial(dim-1);
if numb > dim*fact,
  error('Sorry, I cannot compute with that input.');
end

newnumb = mod(numb, fact);            % the recursive call
last = dim - floor((numb-1)/fact);
if newnumb == 0,
  newnumb = fact;
end
prelim = element(newnumb, dim-1);

if last == dim,                       % the easy case
  e = [prelim dim];
else                   % the case in which an element has to be changed.
e = [(prelim + ismember(prelim, last).*(dim-last)) last];
end
end
```

B.2 Identifying the Group Elements

This is the MatLab code for taking the elements of the group algebra as computed in the previous code and generating where it is in the ordering.

```
function i = basisnumb(element)
     % BASISNUMB takes as input an element from the basis of S(n) and
     % computes the number i, which is where the element lies in my
     % specific ordering of S(n)
     % also see element.

if element == [1],
  % base case, in order to establish the start of the order.
  i = 1;
else
% here is the real knitty gritty.
length = size(element,2);

% I need to get the last entry in element
last = element(length);

% I need to truncate element,
  newelement = element(1:(length-1));

% replace the correct entry with the last entry
newelement = [(newelement + ismember(newelement, length).*(last-length))];

% and give the recursive call
interior = basisnumb(newelement);

% I need to add the results of the recursive call to the product of
% the size of the blocks and the last entry
i = interior + (length - last)*factorial(length-1);

end
```

B.3 JM-elements

This is the MatLab code to generate the JM-elements (see Section 3.3) in matrix form.

```
function JM = JMs(i, n)
% For the JM-element R(i) in S(n)

% initialization
sparsegen = [];
fact = factorial(n);
for j = 1:fact,
  current = element(j,n);
  for k = 1:i-1,
    temp = action(current,[k i]);
    sparsegen = [sparsegen; j basisnumb(temp) 1];
  end
end

  JM = sparse(sparsegen(:,1),sparsegen(:,2),sparsegen(:,3),fact,fact);
```

Bibliography

[1] P. Diaconis, *A generalization of spectral analysis with application to ranked data.* Ann. Statist. **17** (1989), no. 3, 949-979.

[2] P. Diaconis, and D. Rockmore. *Efficient computation of isotypic projections for the symmetric group*, Groups and computation (New Brunswick, NJ, 1991), Amer. Math. Soc., Providence, RI, 1993, pp. 87-104.

[3] G. James and M. Liebeck. *Representations and Characters of Groups.* Cambridge; New York: Cambridge University Press, 1993

[4] G. James, *The representation theory of the symmetric groups.* Berlin; New York: Springer-Verlag, 1978

[5] D.K. Maslen, et al. *Computing Isotypic Projections with the Lanczos Iteration.* Technical Report PMA-TR01-196, Dept. of Math., Dartmouth College, 2001.

[6] M. Orrison, *An eigenspace approach to decomposing representations of finite groups*, Ph.D. thesis, Dartmouth College, 2001.

[7] A. Ram, *Seminormal representations of Weyl groups and Iwahori-Hecke algebras.* Proc. London Math. Soc. (3) 75 (1997), no. 1, 99–133.