Orthogonal Decompositions of Traceless Matrix Spaces

Master's thesis by Patrik Olsson
Supervised by Victor Ufnarovski

2018 – 2019

Abstract

We study orthogonal decompositions of complex special linear Lie algebras or, in other words, linear spaces consisting of complex matrices with zero trace. The conjugacy of the component subspaces give rise to change of basis matrices with a particular form that we, for the moment, call *nice*. We prove a necessary and sufficient condition for this form, which allows us to characterize orthogonal decompositions as a finite set of matrices: namely, the change of basis matrices from the standard diagonal component subspace to each of the other n component subspaces. We develop a basic theory for nice matrices, and then present methods to construct known orthogonal decompositions in terms of the aforementioned characterization of them.

Contents

Chapter 1

Introduction

The work presented in this thesis has been done as a degree project for the master programme in computer science and engineering at the Faculty of Engineering of Lund University. The purpose has been to develop deeper knowledge on the subject of matrix theory, which is itself an advanced theoretical component of the computer graphics specialization.

To this end, the problem of constructing orthogonal decompositions of special linear spaces (consisting of complex matrices with zero trace) has been studied. In particular, this problem has been framed in a way that makes the matrix theory perspective especially relevant, namely by focusing on the change of basis matrices that connect the subspace components of an orthogonal decomposition.

While the problem of orthogonal decompositions originally comes from the subject of Lie algebras, it is not necessary to be familiar with it in order to understand or think about the problem. In fact, the presentation in this thesis is geared towards readers that are familiar with linear algebra and matrix theory. Even some finer details of these two subjects will be repeated to increase accessibility of the material.

Before this introduction chapter ends we will have described what orthogonal decompositions are, as well as provided the basics that will be used in the rest of the report. In the second chapter we will describe the special form that the change of basis matrices take in an orthogonal decomposition. Then in the third chapter, methods of constructing orthogonal decompositions in terms of these change of basis matrices will be presented.

1.1 Notation and basics

Subscript, A_i, always implies that A is a container of some sort (for example a sequence, vector or matrix, depending on the index and element types) and that we are accessing an element at position i from it. It's never used as a part of the name of a variable.

Indices may start counting from either 0 or 1 for sequences in general, but specifically for matrices and vectors they will always start counting from 1. Vectors should be interpreted as column vectors unless otherwise stated. For matrices, $A_{i,j}$ denotes the element at row i and column j in the matrix A.

Any operation (for example, exponentiation) on the same syntactic level as the subscript will apply to the accessed element, and not to the container as a whole. For example $A_{i,j}^{-1}$ means the reciprocal of an element in the matrix A, and not an element in its inverse matrix (which is instead denoted by $(A^{-1})_{i,j}$).

The imaginary unit is written with the Greek letter ι (iota), and should not be confused with i or j which are always used as index variables.

The terms 'linear space', 'vector space' and 'matrix space' are essentially synonymous, and only differ in connotation regarding what kind of elements it consists of. These spaces will be especially important:

1. The linear space of complex $n \times n$ matrices with trace zero:
   $\mathrm{sl}_n(\mathbb{C}) = \{A \in \mathbb{C}^{n\times n} \mid \mathrm{tr}\, A = 0\}$
2. The linear space of complex $n \times n$ diagonal matrices with trace zero:
   $\mathrm{sd}_n(\mathbb{C}) = \{A \in \mathrm{diag}(\mathbb{C}^n) \mid \mathrm{tr}\, A = 0\}$

Componentwise multiplication of two vectors u and v is denoted by $u \odot v$.

We will also make use of the *triangular numbers*:

$$\triangle_n = \sum_{k=1}^{n} k = \binom{n+1}{2} = \frac{n(n+1)}{2}$$

1.2 Orthogonal decompositions

In order to understand orthogonal decompositions, we first need to understand decompositions in general. The idea with a decomposition is that we want to reconstruct a large object in terms of smaller independent objects (elements). A probably familiar example is how we might decompose an integer into a product of its primes, or even just smaller factors. In this thesis we are specifically looking at decompositions of linear spaces (e.g. vector spaces and matrix spaces) by this definition:

Definition 1. A *decomposition of a linear space* L is a sequence V of subspaces of L, such that $L = \sum_i V_i$ and all subspaces in V are linearly independent in the following sense:

$$\forall v_i \in V_i.\left(\sum_i v_i = \mathbf{0} \Rightarrow \forall i.\, v_i = \mathbf{0}\right)$$

Because the independence condition is equivalent to the condition necessary and sufficient for the subspace sum $L = \sum_i V_i$ to be direct (namely that $V_i \cap (\sum_{j \neq i} V_j) = \{\mathbf{0}\}$), we will use the direct sum notation $L = \bigoplus_i V_i$ to denote a decomposition. For finite-dimensional spaces, which is the only kind we will work with in this thesis, the independence condition may be equivalently and more concisely restated as: $\dim L = \sum_i \dim V_i$.

We should now be ready to define an orthogonal decomposition. The term could apply to linear spaces in general, but we will specifically look at only the traceless matrix spaces $\mathrm{sl}_n(\mathbb{C})$ and make the definition accordingly.

Definition 2. A decomposition of $\mathrm{sl}_n(\mathbb{C})$, given by $\mathrm{sl}_n(\mathbb{C}) = \bigoplus_{i=0}^{n} V_i$, is said to be an *orthogonal decomposition* when:

1. Subspaces are conjugate:
   $V_i = S_i^{-1} V_0 S_i$
2. Subspaces are diagonalizable:
   $V_0 = \mathrm{sd}_n(\mathbb{C})$
3. Subspaces are pairwise orthogonal:
   $\forall i \neq j.\, V_i \perp V_j$

Strictly speaking, orthogonality is not yet defined for the subspaces since we have not specified $\mathrm{sl}_n(\mathbb{C})$ as an inner product space or even as a space with a bilinear form. For clarity, the following definition of orthogonality is used:

Definition 3. Two matrix subspaces V_i and V_j are *orthogonal* ($V_i \perp V_j$) when:

$$\forall A \in V_i, B \in V_j.\, \mathrm{tr}(AB) = 0$$

This is consistent with equipping the space with the bilinear form $\langle A, B \rangle = \mathrm{tr}(AB)$. Note however that this bilinear form is not an inner product, so some properties of orthogonality that can be assumed in the context of real vectors with dot product may not hold for matrices in $\mathrm{sl}_n(\mathbb{C})$ with the given bilinear form. For example $\mathrm{tr}(A^2) = 0$ does not necessarily mean that A is the zero matrix, so some non-zero matrices are actually orthogonal to themselves!

Another thing is worth pointing out here, namely that any sequence V of $n+1$ subspaces that satisfy the three listed requirements in definition 2 will in fact form a decomposition in the first place. This is not immediately obvious because the way the definition is stated it seems to imply that we must first know, as a precondition (or zeroth requirement), that the subspaces form a decomposition at all, before being able to even consider whether it is an orthogonal one.

The main reason this precondition is not really necessary is that orthogonality of subspaces (as provided by the third requirement) implies independence of them[1]. Thus they have a direct sum and must be a decomposition of something. Furthermore, the second requirement implies that V_0 has dimension $n - 1$, and the first requirement implies that all other V_i have that same dimension by conjugacy. The direct sum then has dimension $(n + 1)(n - 1) = n^2 - 1$, and the only such dimensioned common superspace is the total space $\mathrm{sl}_n(\mathbb{C})$, which must thus be the thing being decomposed, as expected.

[1]The proof is beyond the scope of this thesis, but it's not difficult to adapt a proof of the same statement regarding vectors.

1.3 Prior work

The problem of orthogonal decompositions of (simple) Lie algebras was first posed and considered in full generality in [2]. A following paper [3] focused more specifically on the problem of orthogonally decomposing Lie algebras of type A_n, and this is in essence the same problem as introduced in definition 2 because the matrix space $\mathrm{sl}_n(\mathbb{C})$ happens to be a Lie algebra of type A_{n-1}.

The papers cited above are written for a quite advanced level. For a more accessible introductory text on Lie algebras and their orthogonal decompositions, another Lund University master's thesis [1] is a recommended read.

The most important point to take from these accounts is that there are known to exist orthogonal decompositions when n is a prime power. However, even for the first example which is not a prime power, $n = 6$, no orthogonal decomposition has ever been found. In fact, there is a standing conjecture called the Winnie-the-Pooh problem that there is no orthogonal decomposition at all for this size.

Bibliography

[1] Per-Anders Andersson. The Winnie-the-Pooh Problem. Master's thesis, Lund University, 2005.

[2] A. I. Kostrikin, I. A. Kostrikin, and V. A. Ufnarovskii. Orthogonal decompositions of simple Lie algebras. *Soviet Mathematics Doklady*, 24(2), 1981.

[3] A. I. Kostrikin, I. A. Kostrikin, and V. A. Ufnarovskii. Orthogonal decompositions of simple Lie algebras (type A_n). *Proceedings of the Steklov Institute of Mathematics*, 1983.

Chapter 2

Nice matrices

The primary subject of this thesis is the form of the change of basis matrices S_i that an orthogonal decomposition will give rise to.

Let's begin by only looking at the condition $V_0 \perp S^{-1}V_0S$. In other words, we look at a special case of the third requirement for an orthogonal decomposition (with $i = 0$ and $j > 0$). By definition, this is equivalent to the condition that $\text{tr}(AB) = 0$ given any $A \in V_0$ and $B \in S^{-1}V_0S$.

Because $V_0 = \text{sd}_n(\mathbb{C})$ we know that an arbitrary A can be written as $A = \text{diag}(a)$ for some $a \in \mathbb{C}^n$ with $\text{tr}\, A = \sum_i a_i = 0$. Likewise, we also have that any B can be written as $B = S^{-1}\,\text{diag}(b)S$ for some $b \in \mathbb{C}^n$ with $\text{tr}(\text{diag}(b)) = \sum_i b_i = 0$.

We would like to expand $\text{tr}(AB) = \sum_k (AB)_{k,k}$. In elementwise description $(AB)_{i,j} = \sum_k A_{i,k}B_{k,j}$. However, only elements $A_{i,i} = a_i$ are nonzero which permits $k = i$, so that $(AB)_{i,j} = a_iB_{i,j}$. To expand further, we need $B_{i,j} = (S^{-1}\,\text{diag}(b)S)_{i,j} = \sum_k (S^{-1})_{i,k}(\text{diag}(b)S)_{k,j} = \sum_k (S^{-1})_{i,k}b_kS_{k,j}$. Putting all this together, we get the complete expansion:

$$\text{tr}(AB) = \sum_i a_iB_{i,i} = \sum_i a_i \sum_j b_j(S^{-1})_{i,j}(S)_{j,i}$$

Under what conditions on the elements of S is this trace equal to zero for any valid assignments of vectors a and b? We begin by looking only at $\sum_i a_iz_i$ with $z_i = B_{i,i}$. In general the only z such that this is zero for arbitrary a, is when $z_i = 0$. However, now we also know that $\sum_i a_i = 0$ so it is also permitted that z is elementwise constant ($z_i = c$ for some $c \in \mathbb{C}$), since we then have $\sum_i a_iz_i = c\sum_i a_i = c \cdot 0 = 0$.

We thus know the trace is zero if and only if the sum $B_{i,i} = \sum_j b_j(S^{-1})_{i,j}(S)_{j,i}$ is a constant. Let's write it as $\sum_j b_jz_j$ with $z_j = (S^{-1})_{i,j}(S)_{j,i}$. The only constant possible for this expression (for arbitrary b satisfying $\sum_i b_i = 0$) is zero, which again is when z is elementwise constant: $z_j = c_i$ (it may theoretically be a different constant for different i, since there is a separate z for each iteration). Thus it is necessary that $(S^{-1})_{i,j}(S)_{j,i} = c_i$, or equivalently, $(S^{-1})_{i,j} = c_i(S)^{-1}_{j,i}$.

However, we also know that $S^{-1}S = I$. Elementwise the left hand side is $(S^{-1}S)_{i,j} = \sum_k (S^{-1})_{i,k}(S)_{k,j} = \sum_k c_i(S)^{-1}_{k,i}(S)_{k,j}$. Comparing with the identity matrix along the diagonal we have that $\sum_k c_i(S)^{-1}_{k,i}(S)_{k,i} = \sum_k c_i = nc_i = 1$, which means that $c_i = n^{-1}$. In conclusion, the necessary and sufficient conditions we sought on S is that $(S^{-1})_{i,j} = (nS)^{-1}_{j,i}$.

We are now also interested in knowing for which conditions on S_i and S_j we satisfy the other cases of the third requirement for an orthogonal decomposition. Namely, that $S_i^{-1}V_0S_i \perp S_j^{-1}V_0S_j$ (for $i > 0, j > 0$ and $i \neq j$), which by definition is equivalent to $\mathrm{tr}(S_i^{-1}AS_iS_j^{-1}BS_j) = 0$ given any A and B in V_0.

Here we can use that $\mathrm{tr}(AB) = \mathrm{tr}(BA)$, and thus find the trace in question to be equal to $\mathrm{tr}(AS_iS_j^{-1}BS_jS_i^{-1}) = \mathrm{tr}(AT^{-1}BT)$ with $T = S_jS_i^{-1}$. So an equivalent condition is that $V_0 \perp T^{-1}V_0T$, and we already know from before how to find simple conditions on T from that.

We collect these results in a definition and a theorem.
Definition 4. An $n \times n$ matrix A is *nice* when it is invertible and its inverse satisfies:

$$(A^{-1})_{i,j} = (nA)^{-1}_{j,i}$$

Theorem 1. In any orthogonal decomposition and only in them, every change of basis matrix S_i will be nice, as well as $S_iS_j^{-1}$ when $i \neq j$.

The conclusion is that we can construct and describe orthogonal decompositions from nice matrices alone since there is an exact correspondence. We don't need to keep track of n-many subspaces each with $(n-1)$-many basis matrices, which is a quadratically increasing amount of information. It is sufficient with the linearly increasing amount of n-many nice matrices. Most importantly, we may look at the problem from just a matrix theory perspective, and do not have to think about Lie algebras or linear spaces as objects themselves.

2.1 Basic properties

While the defining property of nice matrices is a lot easier to work with than the orthogonality conditions directly, it's sometimes inconvenient when it requires expressing the entries of A^{-1} in terms of entries in A (for example using minor determinants). The following theorem gives an equivalent condition which does not require knowing an expression for the inverse matrix. After that a series of theorems will show how the niceness property is preserved by various operations, which give very powerful methods for deciding whether a complicated matrix is nice or not, by reducing it to a simpler one.
Theorem 2. A matrix A is nice if and only if, for any i and j such that $i \neq j$, the following holds:

$$\sum_k \frac{A_{i,k}}{A_{j,k}} = 0, \quad \sum_k \frac{A_{k,i}}{A_{k,j}} = 0$$

(Trivially, when $i = j$, both sums equal n for any A whether nice or not.)

Proof. When A is nice, then $(AA^{-1})_{i,j} = \sum_k \frac{A_{i,k}}{A_{j,k}} = I_{i,j}$. Likewise $(A^{-1}A)_{j,i} = \sum_k \frac{A_{k,i}}{A_{k,j}} = I_{j,i}$. Looking only at $i \neq j$, then both sums equal $I_{i,j} = I_{j,i} = 0$.

Conversely, we apparently have a matrix B such that $B_{i,j} = A^{-1}_{j,i}$ and $AB = BA = nI$ (this is just a more concise way of stating all the assumptions). This means that $n^{-1}B$ is an inverse of A. Since every matrix has only at most one inverse it must be the same as A^{-1}. We have $(A^{-1})_{i,j} = (n^{-1}B)_{i,j} = (nA)^{-1}_{j,i}$ which means that A is nice. □

Theorem 3. If a matrix A is nice, then so is its inverse A^{-1}.

Proof.

$$\begin{aligned} (A^{-1})_{i,j} = (nA)^{-1}_{j,i} &\Leftrightarrow \\ (A^{-1})^{-1}_{i,j} = nA_{j,i} &\Leftrightarrow \\ (nA^{-1})^{-1}_{i,j} = A_{j,i} &\Leftrightarrow \\ (nA^{-1})^{-1}_{i,j} = ((A^{-1})^{-1})_{j,i} & \end{aligned}$$

□

Theorem 4. If a matrix A is nice, then so is A scaled by a nonzero factor c.

Proof.

$$\begin{aligned} ((cA)^{-1})_{i,j} &= c^{-1}(A^{-1})_{i,j} = c^{-1}(nA)^{-1}_{j,i} \\ &= (ncA)^{-1}_{j,i} \end{aligned}$$

□

Theorem 5. If a matrix A is nice, then so is its transpose A^{T}.

Proof.

$$\begin{aligned} (A^{-1})_{i,j} = (nA)^{-1}_{j,i} &\Leftrightarrow \\ (A^{-1})_{j,i} = (nA)^{-1}_{i,j} &\Leftrightarrow \\ ((A^{-1})^{\mathsf{T}})_{i,j} = (nA^{\mathsf{T}})^{-1}_{j,i} &\Leftrightarrow \\ ((A^{\mathsf{T}})^{-1})_{i,j} = (nA^{\mathsf{T}})^{-1}_{j,i} & \end{aligned}$$

□

Theorem 6. If a matrix A is nice, then so is A modified by row or column exchange.

Proof. The conditions in theorem 2 don't depend on the ordering of rows and columns in A. They apply to all pairs of rows (first condition) and pairs of columns (second condition), and those complete sets of pairs are the same regardless of which order we select them. Likewise, row and column exchange may reorder the terms in the sums (over iterations k), but the sums stay the same due to commutativity. □

Theorem 7. If a matrix A is nice, then so is A modified by row or column scaling with a non-zero factor c.

Proof. Since scaling of a column is the same as a scaling of a row in the transpose matrix, through theorem 5 it is enough to prove for the case when row l in A is scaled.

The requirements in theorem 2 on the modified matrix are the following:

$$\sum_k \frac{C_i A_{i,k}}{C_j A_{j,k}} = 0, \quad \sum_k \frac{C_k A_{k,i}}{C_k A_{k,j}} = 0, \quad \text{where } C_i = \begin{cases} c, & \text{if } i = l \\ 1, & \text{if } i \neq l \end{cases}$$

These hold from the assumption that A is nice:

$$\sum_k \frac{C_i A_{i,k}}{C_j A_{j,k}} = \frac{C_i}{C_j} \sum_k \frac{A_{i,k}}{A_{j,k}} = 0 \Leftrightarrow \sum_k \frac{A_{i,k}}{A_{j,k}} = 0$$

$$\sum_k \frac{C_k A_{k,i}}{C_k A_{k,j}} = \sum_k \frac{A_{k,i}}{A_{k,j}} = 0$$

□

2.2 Examples

At this point we know a lot about nice matrices in general, but we haven't actually seen one yet. Here we introduce the two most important classes of nice matrices.

Hadamard matrices

For the case $n = 2$ it's not difficult to find an example of a nice matrix. We already know that we may normalize the first entry to 1 on each row and column due to theorem 7. Once that is done on a 2×2 matrix there is only one entry left to fill in and we find that the value -1 makes the matrix nice.

$$\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Such a matrix is an example of a Hadamard matrix: square matrices which by definition only have entries in $-1, +1$ and all rows (or equivalently: columns) are mutually orthogonal. A Hadamard matrix H will have the property that $H^{-1} = n^{-1} H^{\mathsf{T}}$, which looks very similar to that which defines nice matrices.

It's just that a nice matrix will have an inverse which is proportional to the transpose of the elementwise inverse; not just the transpose directly. However, the difference is immaterial when the entries are their own inverse which is the case for -1 and $+1$ in $\mathbb{C}$. With that we have proved:

Theorem 8. All Hadamard matrices are nice.

Fourier matrices

Hadamard matrices present some examples of nice matrices, but by no means all of them. It is known from the theory of Hadamard matrices that the size of a Hadamard matrix must be 1, 2 or a multiple of 4. It is not even known whether *all* multiples of 4 has a Hadamard matrix of that size. On the other hand, it is known that there is an orthogonal decomposition for all $n = p^m$, and thus there have to be examples of nice matrices for at least all such sizes.

A natural generalization of Hadamard matrices are Fourier matrices (in particular, the 2×2 Fourier matrix is a Hadamard matrix). Like Hadamard matrices, a Fourier matrix F will have an inverse equal to $n^{-1}{F_n}^{\mathsf{H}}$. Should these be nice also, then we have a method of constructing nice matrices of any size since Fourier matrices exist in all sizes. As it happens, it is in fact so:

Theorem 9. The Fourier matrices F_n, defined by $(F_n)_{i,j} = \varepsilon^{(i-1)(j-1)}$ with $\varepsilon = \exp(\frac{2\pi}{n}\iota)$, are nice.

Proof. We know that $F_n^{-1} = n^{-1}{F_n}^{\mathsf{H}}$. Furthermore, the complex conjugate of an element of F_n is in fact just its inverse: $\overline{\varepsilon^{(i-1)(j-1)}} = \exp(-\frac{2\pi}{n}\iota(i-1)(j-1)) = \varepsilon^{-(i-1)(j-1)} = (\varepsilon^{(i-1)(j-1)})^{-1}$. Thus we have that $(F_n^{-1})_{i,j} = n^{-1}\overline{(F_n)_{j,i}} = (nF_n)_{j,i}^{-1}$ which means F_n is nice. □

2.3 Equivalence classes

From theorems 6 and 7, we learnt that the niceness property is preserved when exchanging rows and columns, and also when scaling them by a nonzero scalar. But these operations are the same as multiplying to the left or right (affecting rows and columns, respectively) with permutation matrices and nonzero diagonal matrices. Furthermore, an arbitrary product of these kinds of matrices is always a *monomial matrix*[1]. We may thus capture both these theorems in one that states that LAR is nice if and only if A is, given monomial matrices L and R. This means that in the following definition, both A and B are nice if one of them is.

Definition 5. A matrix A is *monomial-equivalent* to B when $A = LBR$, for some monomial matrices L ("left-monomial") and R ("right-monomial"). More specifically, *left-monomial-equivalent* implies $R = I$ while *right-monomial-equivalent* implies $L = I$.

That this really is an equivalence relation is not difficult to see. Identity matrices are monomial which gives the reflexive property. Monomial matrices are invertible which gives the symmetric property. Finally, the set of monomial matrices (of a certain size) is closed under matrix product, which gives the transitive property.

With an equivalence relation in hand, it is now natural to ask: what equivalence classes does it have in the context of nice matrices?

[1] Also known as a *generalized permutation matrix*: a matrix with exactly one nonzero entry in each row and column, but it does not have to be 1 as in a standard permutation matrix.

For the case $n = 2$ we have actually already found the one and only equivalence class, where the Hadamard matrix given previously (also equal to the Fourier matrix F_2) is a representative. We first used theorem 7 to normalize the first entry to 1 on every row and column. Any matrix of any size, whether nice or not, is monomial-equivalent to such a normalized matrix, so there is no reason not to begin with this step when classifying nice matrices by monomial-equivalence. Then we found that setting the final entry to the value -1 makes the matrix nice. In fact, theorem 2 makes it clear that this is the only value which makes it nice, and consequently the equivalence class is unique.

As for $n = 3$, we again begin by normalizing the first entry on each row and column. Then we repeatedly use theorem 2 on various pairs of rows and columns to get more and more information about the remaining entries. We eventually reach the result below and the fact that $x^2 + x + 1 = 0$, which has the two solutions $x = \varepsilon$ and $x = \varepsilon^2$.

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & x & y \\ 1 & z & w \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 1 & 1 \\ 1 & x & y \\ 1 & y & w \end{pmatrix} \rightarrow$$

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & x & y \\ 1 & y & x \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 1 & 1 \\ 1 & x & -(1+x) \\ 1 & -(1+x) & x \end{pmatrix}$$

We may choose x to be any of the two solutions but we should not conclude that each choice result in a representative for one distinct equivalence class each. Note that $-(1+x)$ is the other of the two solutions, and by row exchange (alternatively column exchange) both solutions actually result in monomial-equivalent matrices. In other words, there is a unique equivalence class and not two. Furthermore, with the solution $x = \varepsilon$ it is clear that a possible representative for the equivalence class is the Fourier matrix F_3.

We have now seen that there is only one single monomial equivalence class for $n = 2$ and $n = 3$ which has the Fourier matrix F_n as a natural representative. This may lead one to entertain the idea that this will generalize to arbitrary n. It would indeed be a very useful thing, making it possible to represent any nice matrix as a pair of monomial matrices. Unfortunately, it is not possible in general. Already $n = 4$ will make a counterexample, as demonstrated below.

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & x & y & z \\ 1 & & & \\ 1 & & & \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & x & -1 & -x \\ 1 & & & \\ 1 & & & \end{pmatrix} \rightarrow$$

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & x & -1 & -x \\ 1 & -1 & & -1 \\ 1 & -x & & x \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & x & -1 & -x \\ 1 & -1 & 1 & -1 \\ 1 & -x & -1 & x \end{pmatrix}$$

Chapter 3

Constructions

We will now look at constructing orthogonal decompositions of $\mathrm{sl}_n(\mathbb{C})$ for various values of n, by applying the knowledge we have collected regarding nice matrices. The general procedure will be to construct nice matrices that satisfy the conditions of theorem 1.

3.1 Case $n = 2$

In order to construct an orthogonal decomposition for $n = 2$, we are looking for two nice matrices S_1 and S_2 such that also $S_2S_1^{-1}$ is nice.

From last chapter we know that we can write $S_1 = L_1F_2R_1$ and $S_2 = L_2F_2R_2$, where L_1, L_2, R_1 and R_2 are monomial matrices and F_2 is the Fourier matrix of size 2. Without loss of generality we may even assume that R_1 and R_2 are (invertible) diagonal matrices. Because if they had a non-identity permutation factor (there is only one such when $n = 2$) it would exchange the columns of F_2, which can also be accomplished by scaling the second row by -1, which corresponds to a modification of the opposite monomial matrix (L_1 or L_2).

With this rewrite, we have that $S_2S_1^{-1} = L_2F_2R_2R_1^{-1}F_2^{-1}L_1^{-1}$. The inverse of F_2 is again F_2 but scaled by $1/2$, and there are matrices to the sides, L_2 and L_1^{-1}, which are monomial. These factors do not affect the niceness of $S_2S_1^{-1}$, and thus it is enough to look only at when $F_2R_2R_1^{-1}F_2$ is nice.

When that matrix is nice it should also be rewritable as a monomial equivalent to F_2. So clearly we must find an invertible diagonal matrix $X = R_2R_1^{-1}$ such that $F_2XF_2 = L'F_2R'$ for some monomial matrices L' and R'. Once we have found X, we may easily decompose it by setting R_1 to an arbitrary invertible diagonal matrix and then calculate $R_2 = XR_1$.

Let $X = \mathrm{diag}(x)$ and set L' and R' such that $(L')^{-1}F_2XF_2(R')^{-1}$ is normalized in the first entry on all rows and columns.

$$\begin{pmatrix} x_1+x_2 & x_1-x_2 \\ x_1-x_2 & x_1+x_2 \end{pmatrix} \to \begin{pmatrix} 1 & 1 \\ 1 & \frac{(x_1+x_2)^2}{(x_1-x_2)^2} \end{pmatrix}$$

In order for this to equal F_2 according to the condition above, the final entry must equal -1. However, we also need to keep in mind that both $x_1 + x_2$ and $x_1 - x_2$ must be nonzero since we inverted them when performing the normalization. The complete system we need to solve is this:

$$\begin{aligned} x_1 + x_2 &\neq 0 \\ x_1 - x_2 &\neq 0 \\ (x_1+x_2)^2 + (x_1-x_2)^2 &= 0 \end{aligned}$$

The final equation is equivalent to $x_1^2 + x_2^2 = 0$ and $x_1 = 1, x_2 = \iota$ is one possible solution that also satisfies the inequations. In fact, for any $t \neq 0$ there is a solution in the form of $x_1 = t, x_2 = t\iota$, and this gives us a way to completely parameterize all orthogonal decompositions for $n = 2$. We are satisfied with the one solution though, and after arbitrarily setting $L_1 = L_2 = R_1 = I$ we can easily calculate S_1, R_2 and then S_2:

$$\begin{aligned} S_1 = L_1F_2R_1 &= IF_2I = F_2 \\ R_2 = XR_1 &= XI = X \\ S_2 = L_2F_2R_2 &= IF_2X = F_2X \end{aligned}$$

$$S_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad S_2 = \begin{pmatrix} 1 & \iota \\ 1 & -\iota \end{pmatrix}$$

3.2 Case $n = 3$

We need to find three matrices, S_1, S_2 and S_3, such that $S_1S_2^{-1}$, $S_2S_3^{-1}$ and $S_1S_3^{-1}$ are also nice. Similar to the case $n = 2$ we assume that $S_i = L_iF_3R_i$, where F_3 is the Fourier matrix of size 3 and L_i and R_i are monomial matrices. Furthermore, also like the previous case, R_i may even be assumed to be invertible diagonal matrices, because all permutations of columns in F_3 can instead be equivalently performed[1] with a modification to L_i.

We also find again that the left-monomials are completely unimportant because they do no affect the niceness of $S_iS_j^{-1} = L_iF_3R_iR_j^{-1}F_3^{-1}L_j^{-1}$. In fact, this always holds for any n and we could safely drop them entirely without losing

[1]One can easily verify this since there are only five cases, namely the $3! - 1$ different non-identity permutations of the columns. They are all equivalently realized by scaling the bottom two rows by ε and ε^2 in some order, and/or exchanging them.

generality. However, they have been very useful to allow us assume a more simple shape on the right-monomials as we have done now for both $n = 2$ and $n = 3$. Still, at the end of the day we have no reason not to set $L_i = I$.

We let $X = \operatorname{diag}(x) = R_1R_2^{-1}$, $Y = \operatorname{diag}(y) = R_2R_3^{-1}$ and note that $XY = \operatorname{diag}(x \odot y) = R_1R_3^{-1}$. Thus we are looking for X and Y such that all of the following matrices are nice: F_3XF_3, F_3YF_3 and F_3XYF_3. We will proceed by finding the solution space for X alone, which is clearly also the solution space for Y and XY alone. Then we take particular solutions for X and Y such that XY also happens to be inside the solution space.

We will be interested in the elements of F_3XF_3 in terms of x.

$$F_3XF_3 = \begin{pmatrix} x_1+x_2+x_3 & x_1+\varepsilon x_2+\varepsilon^2 x_3 & x_1+\varepsilon^2 x_2+\varepsilon x_3 \\ x_1+\varepsilon x_2+\varepsilon^2 x_3 & x_1+\varepsilon^2 x_2+\varepsilon x_3 & x_1+x_2+x_3 \\ x_1+\varepsilon^2 x_2+\varepsilon x_3 & x_1+x_2+x_3 & x_1+\varepsilon x_2+\varepsilon^2 x_3 \end{pmatrix}$$

Using theorem 2 with normalization of the fractions, we construct a polynomial system for the niceness of F_3XF_3:

$$\begin{aligned} \chi_1 = x_1+x_2+x_3 &\neq 0 \\ \chi_2 = x_1+\varepsilon x_2+\varepsilon^2 x_3 &\neq 0 \\ \chi_3 = x_1+\varepsilon^2 x_2+\varepsilon x_3 &\neq 0 \\ \chi_1^2\chi_3+\chi_2^2\chi_1+\chi_3^2\chi_2 &= 0 \\ \chi_1^2\chi_2+\chi_2^2\chi_3+\chi_3^2\chi_1 &= 0 \end{aligned}$$

Simplification of the two equations gives the following equivalent:

$$\begin{aligned} x_1^3+\varepsilon x_2^3+\varepsilon^2 x_3^3 &= 0 \\ x_1^3+\varepsilon^2 x_2^3+\varepsilon x_3^3 &= 0 \end{aligned}$$

With a couple of substitutions and simplifications we furthermore find the two equations to be equivalent to $x_1^3 = x_2^3 = x_3^3$. For those conditions alone, we have a solution precisely when the x-values are a third root of unity (power of ε) multiple to each other. The inequality conditions further require them to be not all equal nor all different. In conclusion, for every $t \neq 0$ we have solutions $x = (\varepsilon t, t, t)$ and $x = (\varepsilon^2 t, t, t)$, as well as permutations of these.

Setting $x = (\varepsilon, 1, 1)$ and $y = (1, \varepsilon, 1)$ we get $x \odot y = (\varepsilon, \varepsilon, 1)$. These are all in the solution space and thus make sure that all of F_3XF_3, F_3YF_3 and F_3XYF_3 are nice. We can now find R_1, R_2 and R_3 by setting one of them to an arbitrary invertible diagonal matrix and calculating the other two from X and Y. For example, we set $R_2 = I$ which means $R_1 = X$ and $R_3 = Y^{-1}$. Now it's only a simple matter of calculating S_1, S_2 and S_3:

$$S_1 = \begin{pmatrix} \varepsilon & 1 & 1 \\ \varepsilon & \varepsilon & \varepsilon^2 \\ \varepsilon & \varepsilon^2 & \varepsilon \end{pmatrix}, \quad S_2 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & \varepsilon & \varepsilon^2 \\ 1 & \varepsilon^2 & \varepsilon \end{pmatrix}, \quad S_3 = \begin{pmatrix} 1 & \varepsilon^2 & 1 \\ 1 & 1 & \varepsilon^2 \\ 1 & \varepsilon & \varepsilon \end{pmatrix}$$

3.3 Case n is prime

The methods we have been using up to now start to become impractical when generalized to $n > 3$. We will find that we cannot assume without loss of generality that the right-monomials are diagonal, because some permutations won't be equivalently representable with a modification to the left-monomial. This means we need to consider all remaining permutations which in worst case are factorially many. It does not help that the number of constraints that need to hold simultaneously, namely that $S_i S_j^{-1}$ should be nice for all $i > j$, is increasing quadratically. This quadratic amount acts as an exponent in the complexity measure since we need to consider all combinations of permutations from different right-monomials.

While $n = 4$ might be barely possible to do by hand if one had the patience (and luck in testing options in the right order), any larger size would almost surely be beyond even the fastest computers. Especially in worst case runtime which would happen if there is no orthogonal decomposition for a certain size n. It does not even seem like the method would be any more advantageous than a more direct approach. Each permutation would ask the computer to solve a polynomial system, but we could have provided it with a single such system by never decomposing the S-matrices into monomial equivalences in the first place.

Still, it is known that there are, at least, orthogonal decompositions for any prime power n, and we should be able to find sets of nice matrices that correspond to these known decompositions. In Per-Anders Andersson's thesis [1] a construction is presented in terms of bases for the subspaces V_i, which is more specifically proven correct for any prime n (not prime power). What follows here is an adaptation of this result which presents an equal construction in terms of nice matrices.

Andersson's construction is built on a basic kind of matrix called *nearly diagonal*, which is a diagonal matrix whose entries have been shifted, with wrap-around, a number of steps to the left.

$$\mathrm{ND}_h(a) = \begin{pmatrix} & & & a_1 & & \\ & & & & \ddots & \\ & & & & & a_h \\ a_{h+1} & & & & & \\ & \ddots & & & & \\ & & a_n & & & \end{pmatrix}$$

More precisely, its elementwise definition is:

$$(\mathrm{ND}_h(a))_{i,j} = \begin{cases} a_i, & \text{if } j = i - h \pmod{n} \\ 0, & \text{otherwise} \end{cases}$$

Then together with the fixed diagonal subspace V_0, the following subspaces V_k for $1 \le k \le n$ (with prime n), are claimed to form an orthogonal decomposition:

$$V_k = \left\langle \mathrm{ND}_h(\varepsilon^{kh}, \ldots, \varepsilon^{nkh}) \mid 1 \leq h < n \right\rangle, \text{where } \varepsilon = \exp(\frac{2\pi}{n}\iota)$$

The proof by Andersson is quite elegant, but we will not repeat it here. Instead we will discover an alternate proof when we extract, from this construction, the change of basis matrices S_k and then proceed to show that they indeed satisfy the conditions of theorem 1, when n is a prime.

One of the necessary conditions for an orthogonal decomposition is that $V_k = S_k^{-1} V_0 S_k = S_k^{-1}\,\mathrm{sd}_n(\mathbb{C}) S_k$, but this is just another way of saying that all matrices in V_k should be simultaneously diagonalizable. The columns of S_k^{-1} become the common eigenvectors, and the eigenvalues for each matrix in V_k are taken from a corresponding diagonal matrix in $\mathrm{sd}_n(\mathbb{C})$.

We defer the issue of simultaneous diagonalizability for a while, and begin with just extracting a candidate for S_k. While we could do this by taking an arbitrary matrix in V_k and calculate its eigenvectors using standard methods, there is a simpler way if we choose to use specifically the first basis matrix ($h = 1$) given above: $\mathrm{ND}_1(\varepsilon^k, \ldots, \varepsilon^{nk})$. The specific diagonalization is then:

$$\mathrm{ND}_1(\varepsilon^k, \ldots, \varepsilon^{nk}) = S_k^{-1} \Lambda S_k, \quad \Lambda = \mathrm{diag}(\lambda) \in \mathrm{sd}_n(\mathbb{C})$$

We now multiply with S_k to the left on both sides, and look at each element of this matrix equation.

$$\begin{aligned}
(S_k\,\mathrm{ND}_1(\varepsilon^k, \ldots, \varepsilon^{nk}))_{i,j} &= (\Lambda S_k)_{i,j} \Leftrightarrow \\
\sum_l (S_k)_{i,l} (\mathrm{ND}_1(\varepsilon^k, \ldots, \varepsilon^{nk}))_{l,j} &= \sum_l \Lambda_{i,l} (S_k)_{l,j} \Leftrightarrow \\
(S_k)_{i,j+1} \varepsilon^{(j+1)k} &= \lambda_i (S_k)_{i,j} \Leftrightarrow \\
(S_k)_{i,j+1} &= \lambda_i \varepsilon^{-(j+1)k} (S_k)_{i,j}
\end{aligned}$$

A prototype for a recurrence relation is becoming apparent. Given that we eventually show S_k to be nice, then knowing that the columns of S_k^{-1} are the eigenvectors we would expect the rows of S_k to be the elementwise inverse of the eigenvectors divided by n. These rows are precisely what the recurrence relation is generating.

Since any scaled version of an eigenvector is also an eigenvector, we may normalize all of their first coordinates which gives natural initial conditions, namely that $(S_k)_{i,0} = 1$. Furthermore, since the ordering of eigenvectors is not important, we are free to reorder the rows of S_k as well, or equivalently, we don't have to mind which eigenvalue is assigned to which λ_i.[2]

In order to complete the recurrence relation, we will now calculate the eigenvalues λ_i from the characteristic equation. With the exception for the top-right

[2] As an interesting sidenote, these properties of eigenvectors can be related to left-monomials in the sense that replacing any S_k with something left-monomial-equivalent will not produce a different orthogonal decomposition.

element, the matrix $\mathrm{ND}_1(\varepsilon^k, \dots, \varepsilon^{nk}) - \lambda I$ is lower triangular. This makes the Gaussian elimination method for calculating the determinant particularly suitable since in this special case it is enough with $2(n-1)$ row operations.

The row operations we perform are given by this algorithm: For each row i starting from the bottom and ending at the second row, we scale the first row by λ and then add to this the row i multiplied by the value in column i on the first row. The reader is encouraged to check that this works. Below is presented the first and final step of the Gaussian elimination.

$$\begin{pmatrix} -\lambda & & & & \varepsilon^k \\ \varepsilon^{2k} & -\lambda & & & \\ & \varepsilon^{3k} & \ddots & & \\ & & \ddots & -\lambda & \\ & & & \varepsilon^{nk} & -\lambda \end{pmatrix} \to \begin{pmatrix} (\prod_i \varepsilon^{ik}) - \lambda^n & & & & \\ \varepsilon^{2k} & -\lambda & & & \\ & \varepsilon^{3k} & \ddots & & \\ & & \ddots & -\lambda & \\ & & & \varepsilon^{nk} & -\lambda \end{pmatrix}$$

The determinant of the triangular result is $((\prod_i \varepsilon^{ik}) - \lambda^n)(-\lambda)^{n-1}$, but we also need to divide by λ^{n-1} to get the determinant of the original matrix, because $(n-1)$-many times we scaled a row with the factor λ. In conclusion, we find that the characteristic equation is equivalent to $\lambda^n = \prod_i \varepsilon^{ik} = \varepsilon^{(\sum_i ik)} = \varepsilon^{\frac{n(n+1)}{2}k}$. When n is odd ($n = 2\tilde{n}+1$), the right hand side simplifies to $\varepsilon^{n(\tilde{n}+1)k} = \varepsilon^0 = 1$, and when n is even ($n = 2\tilde{n}$) it simplifies to $\varepsilon^{\frac{n}{2}(n+1)k} = \varepsilon^{\tilde{n}nk + \frac{n}{2}k} = \varepsilon^{\frac{n}{2}k} = (-1)^k$. We capture both of these cases with $\lambda^n = (-1)^{k(n+1)}$, and thus λ_i should be each nth root of this. One way to describe these is $\lambda_i = \sqrt[n]{(-1)^{k(n+1)}}\varepsilon^{i-1}$, which is easily verified by raising it to the power of n again. For sake of presentation, we note that $\sqrt[n]{(-1)^{k(n+1)}} = \varepsilon^{\frac{1}{2}(k(n+1) \bmod 2)}$ which will be shortened to $\varepsilon^{\phi(k)}$.[3]

Now we have everything necessary to fully evaluate the recurrence relation:

$$(S_k)_{i,j} = \lambda_i^{j-1} \prod_{l=1}^{j-1} \varepsilon^{-(l+1)k} = \lambda_i^{j-1} \varepsilon^{(\sum_{l=1}^{j} -lk)} \varepsilon^k = \varepsilon^{(j-1)\phi(k)} \varepsilon^{(i-1)(j-1)} \varepsilon^{(1-\triangle_j)k}$$

The fact that each S_k is nice is not difficult to see since they are, with exception for column scaling, the Fourier matrix of size n given in theorem 9. The niceness of $S_{k_1} S_{k_2}^{-1}$ for $k_1 \neq k_2$ is a bit more difficult to deduce. The expression for it (given below) does not make it immediately obvious it is any form of nice matrix that we know of. Of course, it might not even be nice in general so we may now have to make use of the assumption that n is prime.

[3]The reader is reminded that the following reduction, while tempting, is incorrect:

$$\sqrt[n]{(-1)^{k(n+1)}} = \sqrt[n]{\varepsilon^{\frac{n(n+1)}{2}k}} \doteq \sqrt{\varepsilon^{(n+1)k}} = \sqrt{\varepsilon^{nk+k}} = \sqrt{\varepsilon^k}$$

The argument fails where there is a dot over the equality sign, because for complex values it is not generally the case that $(z^{w_1})^{w_2} = z^{w_1 w_2}$.

$$(S_{k_1}S_{k_2}^{-1})_{i,j} = \sum_{l=1}^{n}(S_{k_1})_{i,l}(S_{k_2}^{-1})_{l,j} = \frac{1}{n}\sum_{l=1}^{n}(S_{k_1})_{i,l}(S_{k_2})_{j,l} =$$

$$\frac{1}{n}\sum_{l=1}^{n}\varepsilon^{(l-1)\phi(k_1)}\varepsilon^{(i-1)(l-1)}\varepsilon^{(1-\triangle_l)k_1}\varepsilon^{-(l-1)\phi(k_2)}\varepsilon^{-(j-1)(l-1)}\varepsilon^{-(1-\triangle_l)k_2} =$$

$$\frac{1}{n}\sum_{l=1}^{n}\varepsilon^{(l-1)(\phi(k_1)-\phi(k_2))}\varepsilon^{(\triangle_l-1)(k_2-k_1)}\varepsilon^{(i-j)(l-1)}$$

Almost directly from the definition of nice matrices, in order for $S_{k_1}S_{k_2}^{-1}$ to be nice it is sufficient that:

$$n^2(S_{k_1}S_{k_2}^{-1})_{i,j}(S_{k_2}S_{k_1}^{-1})_{j,i} = n$$

Expansion of the left hand side gives:

$$n^2(S_{k_1}S_{k_2}^{-1})_{i,j}(S_{k_2}S_{k_1}^{-1})_{j,i} =$$

$$\sum_{l_1=1}^{n}\sum_{l_2=1}^{n}\varepsilon^{(l_1-1)(\phi(k_1)-\phi(k_2))}\varepsilon^{(\triangle_{l_1}-1)(k_2-k_1)}\varepsilon^{(i-j)(l_1-1)}$$

$$\varepsilon^{(l_2-1)(\phi(k_2)-\phi(k_1))}\varepsilon^{(\triangle_{l_2}-1)(k_1-k_2)}\varepsilon^{(j-i)(l_2-1)} =$$

$$\sum_{l_1=1}^{n}\sum_{l_2=1}^{n}\varepsilon^{(l_1-l_2)(\phi(k_1)-\phi(k_2))}\varepsilon^{(\triangle_{l_1}-\triangle_{l_2})(k_2-k_1)}\varepsilon^{(i-j)(l_1-l_2)}$$

We can easily see that n of these n^2 terms equal $\varepsilon^0 = 1$, namely when $l_1 = l_2$. That means that the other $n^2 - n$ terms have to sum to 0, in order for the whole sum to be n. Due to the presence of $\phi(k)$ which is sometimes one half, all terms are $(2n)$th roots of unity. However, that is only the case when k is odd and n is even, because otherwise $\phi(k) = 0$. From our assumption that n is prime we may check the only even prime separately, and then deal with odd primes where $\phi(k_1)$ and $\phi(k_2)$ are zero and hence all terms become prime nth roots of unity which greatly simplifies analysis.

The case $n = 2$ can be straightforwardly, but tediously, verified with case analysis, evaluation and comparison, but some simplifications save many calculations. For example, note that $\triangle_{l_1} - \triangle_{l_2} = 0 \pmod 2$ for all l_1 and l_2. Furthermore, without loss of generality we may assume $k_1 < k_2$ and consequently $k_1 = 1$, $k_2 = 2$ and then $\phi(k_1) - \phi(k_2) = \frac{1}{2}$. With these simplification taken into account, the terms are $(-1)^{(\frac{1}{2}+i-j)(l_1-l_2)}$ and hence on the form $(\pm\sqrt{-1})^{(l_1-l_2)}$ for all i and j. Because the multiplicative inverse of the imaginary unit is also its additive inverse, we easily see that the sum of all terms is $n = 2$ as expected.

The case of odd prime follows. In general $m > 1$ roots of unity sum to zero if and only if they are equally spaced and m is a divisor of the root degree. Since the degree is prime n, we have to identify $(n-1)$-many sets of n terms (among those where $l_1 \neq l_2$) where each set contains all roots of unity. One such set can be found along each "diagonal" $l_2 = l_1 + x \pmod{n}$ for nonzero x.

Since $\triangle_{a+b} = \triangle_a + \triangle_b + ab$, the exponents along a diagonal are $(\triangle_{l_1} - \triangle_{(l_1+x)})(k_2 - k_1) + (i-j)(l_1 - (l_1 + x)) = (-\triangle_x - xl_1)(k_2 - k_1) + -(i-j)x$. This is just a linear function in l_1 with x, k_1, k_2, i and j as parameters to the coefficients. Note that the linear coefficient is never zero, because $k_1 \neq k_2$ and $x \neq 0$. In a finite field every such linear function will be bijective, so every value on l_1 (which selects a term in the set of terms represented by x) will produce a different exponent and thus make sure the set contains all roots of unity.

At this point we have, in terms of nice matrices by theorem 1, a complete constructive proof of an orthogonal decomposition of $\mathrm{sl}_n(\mathbb{C})$ for every prime n. We summarize this result in the following theorem.

Theorem 10. For every prime n, the below given $n \times n$ matrices S_k (for $1 \leq k \leq n$) form a complete set of change of basis matrices for an orthogonal decomposition of $\mathrm{sl}_n(\mathbb{C})$.

$$(S_k)_{i,j} = \varepsilon^{(j-1)\phi(k)}\varepsilon^{(1-\triangle_j)k}\varepsilon^{(i-1)(j-1)}$$
$$\text{where } \phi(k) = \frac{1}{2}(k(n+1) \bmod 2)$$

As promised we should also find an alternate proof that Andersson's construction is correct. The remaining issue, which we deferred previously, is showing that *all* matrices in V_k, and not just $\mathrm{ND}_1(\varepsilon^k, \ldots, \varepsilon^{nk})$, are simultaneously diagonalizable with the change of basis S_k^{-1}. In other words, for all matrices $M \in V_k$, we must show that $S_k M S_k^{-1}$ is diagonal[4].

All linear combinations of simultaneously diagonalizable matrices are also simultaneously diagonalizable with the same change of basis. Thus it is enough to verify the condition for every basis matrix of V_k, namely $M = \mathrm{ND}_h(\varepsilon^{kh}, \ldots, \varepsilon^{nkh})$.

$$\begin{aligned}
&(S_k \,\mathrm{ND}_h(\varepsilon^{kh}, \ldots, \varepsilon^{nkh}) S_k^{-1})_{i,j} = \\
&\sum_{l_1,l_2} (S_k)_{i,l_1} (\mathrm{ND}_h(\varepsilon^{kh}, \ldots, \varepsilon^{nkh}))_{l_1,l_2} (S_k^{-1})_{l_2,j} = \\
&\sum_l (S_k)_{i,l+h} \varepsilon^{(l+h)hk} (S_k^{-1})_{l,j} = n^{-1} \sum_l (S_k)_{i,l+h} (S_k)^{-1}_{j,l} \varepsilon^{(l+h)hk} = \\
&n^{-1} \sum_l \varepsilon^{h\phi(k)} \varepsilon^{(\triangle_l - \triangle_{l+h})k} \varepsilon^{(i-1)(l+h-1)} \varepsilon^{-(j-1)(l-1)} \varepsilon^{(l+h)hk} = \\
&n^{-1} \sum_l \varepsilon^{h\phi(k)} \varepsilon^{-(\triangle_h + lh)k} \varepsilon^{(i-j)(l-1)} \varepsilon^{(i-1)h} \varepsilon^{(l+h)hk} = \\
&n^{-1} \varepsilon^{h\phi(k)} \varepsilon^{(h^2 - \triangle_h)k} \varepsilon^{(i-1)h} \sum_l \varepsilon^{(i-j)(l-1)}
\end{aligned}$$

It is clear that this is diagonal. When $i \neq j$, the final sum is over every nth root of unity so its total is 0, and when $i = j$ all terms are 1 and so the sum is n.

[4]Which is then necessarily in $\mathrm{sd}_n(\mathbb{C})$, because all matrices in V_k have zero trace.

3.4 Case n is prime power

A common operation to construct larger matrices from smaller ones is the Kronecker product. It's actually defined for non-square matrices in general, but for our purposes it's sufficient with square matrices. Given a matrix A of size $n \times n$ and a matrix B of size $m \times m$, the Kronecker product $A \otimes B$ is a matrix of size $nm \times nm$. We describe row indices of a Kronecker product using pairs of values (i_1, i_2), with $i_1 \in \{1, \dots, n\}$ and $i_2 \in \{1, \dots, m\}$, which can be understood as a shorthand for the 'real' index $(i_1 - 1)m + i_2$. The same thing applies to column indices (j_1, j_2). Note that in a Kronecker product of more than two operands, the row and column indices will in general be tuples.

The defining property of Kronecker product is:

$$(A \otimes B)_{(i_1,i_2),(j_1,j_2)} = A_{i_1,j_1} B_{i_2,j_2}$$

The operator is associative, bilinear and non-commutative. We will furthermore make use of the following two properties:

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$$

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$$

We will now investigate whether this can be used to construct orthogonal decompositions from smaller ones. We have at least this result:

Theorem 11. Given two nice matrices A and B, the Kronecker product of them $(A \otimes B)$ is also nice.

Proof. We let the size of A be n and the size of B be m.

$$\begin{aligned}(A^{-1} \otimes B^{-1})_{(i_1,i_2),(j_1,j_2)} = (A^{-1})_{i_1,j_1}(B^{-1})_{i_2,j_2} =\\ (nA)^{-1}_{j_1,i_1}(mB)^{-1}_{j_2,i_2} = (nA \otimes mB)^{-1}_{(j_1,j_2),(i_1,i_2)} =\\ (nm(A \otimes B))^{-1}_{(j_1,j_2),(i_1,i_2)}\end{aligned}$$

□

A reasonable idea now is that we can, maybe, construct orthogonal decompositions of prime power sizes by performing tuplewise Kronecker product of the nice matrices given by theorem 10 and then invoke theorem 1. For example, for $n = p^m$ with $m = 2$, we would use $T_{(k_1,k_2)} = S_{k_1} \otimes S_{k_2}$ as change of basis matrices, where S_k are given by theorem 10. These would be nice according to theorem 11, but can the same be said of $T_{(i_1,i_2)} T^{-1}_{(j_1,j_2)}$ for all $(i_1, i_2) \neq (j_1, j_2)$? Unfortunately not. In particular there is $T_{(i,k)} T^{-1}_{(j,k)} = S_i S_j^{-1} \otimes I$, which cannot be nice since it has elements with value zero.

Other attempts at defining $T_{(k_1,k_2)}$ in terms of matrix products, Kronecker products and the nice matrices given by theorem 10 will fail as well. In fact, theorem 11 is applicable for matrices A and B which are not necessarily of the same size. Although our current goal is only to construct orthogonal decompositions of prime power sizes, we would also get general composite sizes for free. Even if there might exist orthogonal decompositions of general composite sizes, it's at least unlikely they could be found this easily.

In [3], a construction for the prime power case ($n = p^m$) is given in terms of bases for the decomposition subspaces. The basis matrices are given as very specific combinations of m-ary Kronecker products of the matrices $J_{a,b}$ given below.

$$J_{a,b} = D^a P^b, \quad D = \operatorname{diag}(\varepsilon^0, \ldots, \varepsilon^{p-1}), \quad P = \mathrm{ND}_1(1, \ldots, 1)$$

Note here that $\varepsilon = \exp(\frac{2\pi}{p}\iota)$ (pth root of unity; not nth) which will be the case for the rest of this chapter. Furthermore, $D^p = P^p = I$ so we assume $a, b \in \mathbb{Z}_p$.

First find an irreducible m-degree polynomial $Q = \theta^m + \sum_{d=0}^{m-1} t_d \theta^d \in \mathbb{F}_p[\theta]$ to make the finite field $\mathbb{F}_{p^m} = \mathbb{F}_p[\theta]/\langle Q \rangle$. This finite field is also an m-dimensional linear space over $\mathbb{F}_p$ and we will need bases σ and τ, defined by $\sigma_i = \theta^{i-1} + \sum_{l=1}^{i-1} t_{m-l} \theta^{i-1-l}$ and $\tau_i = \theta^{m-i}$ (for $1 \le i \le m$). Then together with the fixed diagonal subspace, the below defined subspaces $\mathcal{V}_\alpha$ (with $\alpha \in F_{p^m}$) will form an orthogonal decomposition of $\mathrm{sl}_{p^m}(\mathbb{C})$:

$$\mathcal{V}_\alpha = \langle \mathcal{J}_{\alpha\beta,\beta} \mid \beta \neq 0 \in \mathbb{F}_{p^m} \rangle$$

$$\text{where } \mathcal{J}_{u,v} = \bigotimes_{i=1}^{m} J_{a_i,b_i} \text{ given } u = \sum_i a_i \sigma_i \text{ and } v = \sum_i b_i \tau_i$$

We would like to make an equivalent construction in terms of nice matrices just like we did for the prime case previously. Unfortunately, the usage of non-prime finite field arithmetic, and the difficulty of finding irreducible polynomials for constructing these finite fields, poses problems for analyzing the construction with generality. We can at least look at specific small cases though, and we will do so for $n = 4 = 2^2$ later in this section.

One thing we can recognize in this construction for the prime power case is that the matrices $J_{a,b}$ for $b \neq 0$ are, up to scaling, the same as the nearly diagonal basis matrices used in the prime case construction, denoted by $M_{k,h}$ below for the hth basis matrix in the kth subspace. In summary, the following holds:

$$M_{k,h} = S_k^{-1} \Lambda_{k,h} S_k = \mathrm{ND}_h(\varepsilon^{kh}, \ldots, \varepsilon^{pkh}) = (\varepsilon D)^{kh} P^h = \varepsilon^{kh} J_{kh,h}$$

$$1 \le k \le p, \quad 1 \le h < p$$

Here S_k are the nice matrices given by theorem 10 for $n = p$ and $\Lambda_{k,h}$ are the diagonal matrices in $\mathrm{sd}_n(\mathbb{C})$ found in the discussion about simultaneous diagonalization at the end of last section. The function $k \mapsto kh$ is bijective (modulus p) so collectively $J_{kh,h}$ will indeed denote the same matrices as $J_{a,b}$ for $b \neq 0$, just in a different order.

Likewise, the remaining $J_{a,b}$ when $b = 0$ form a basis for $\operatorname{diag}(\mathbb{C}^n)$, the space of all diagonal complex matrices (with arbitrary trace). More specifically when $a \neq 0$ and $b = 0$ they form a basis for $\mathrm{sd}_n(\mathbb{C})$ or, in other words, the default zeroth subspace of every orthogonal decomposition.

Now imagine that the construction for $n = p^2$ calls for some basis matrix of some subspace to be $J_{a_1,b_1} \otimes J_{a_2,b_2}$ with $b_1, b_2 \neq 0$ so we can rewrite it as $J_{k_1h_1,h_1} \otimes J_{k_2h_2,h_2}$. Scaling basis elements will not alter the space, so we choose to instead use $\varepsilon^{k_1h_1} J_{k_1h_1,h_1} \otimes \varepsilon^{k_2h_2} J_{k_2h_2,h_2} = M_{k_1,h_1} \otimes M_{k_2,h_2}$. We may easily calculate its diagonalization:

$$M_{k_1,h_1} \otimes M_{k_2,h_2} = S_{k_1}^{-1} \Lambda_{k_1,h_1} S_{k_1} \otimes S_{k_2}^{-1} \Lambda_{k_2,h_2} S_{k_2} = (S_{k_1}^{-1} \otimes S_{k_2}^{-1})(\Lambda_{k_1,h_1} \otimes \Lambda_{k_2,h_2})(S_{k_1} \otimes S_{k_2})$$

Furthermore, it is clear that $J_{k_1,1} \otimes J_{k_2,1}$, and thus also $M_{k_1,1} \otimes M_{k_2,1}$, have to be in different subspaces for different (k_1, k_2) pairs. Ostensibly this would imply that $S_{k_1} \otimes S_{k_2}$ are, after all, nice change of basis matrices of an orthogonal decomposition for $n = p^2$. We have already shown that this is not possible though, so there is some detail we are missing. That detail is the fact that $M_{k_1,1} \otimes M_{k_2,1}$ will potentially have eigenvalues with multiplicity larger than one. In fact, the diagonal of $\Lambda_{k_1,1} \otimes \Lambda_{k_2,1}$ will contain p copies of all pth roots of unity (sometimes scaled by ι when $p = 2$).

What this means is that the columns of $S_{k_1}^{-1} \otimes S_{k_2}^{-1}$, which are the eigenvectors, may be modified according to elementary columns operations as long as they are associated with the same eigenvalue. Furthermore we can reorder them as long as we also reorder the same columns in $\Lambda_{k_1,1} \otimes \Lambda_{k_2,1}$. This gives us a range of possible diagonalizations and we must, actually, specifically find the diagonalization that simultaneously diagonalizes all matrices in the same subspace, not just $M_{k_1,1} \otimes M_{k_2,1}$. In order to do that, we might also need to find which subspaces the other Kronecker products $J_{a_1,b_1} \otimes J_{a_2,b_2}$ should go, and that is not trivial for previously mentioned reasons.

However, as said before we could at least look at a specific case, and we will do so now. When $p = 2$ and $m = 2$, we have the irreducible polynomial $Q = \theta^2 + \theta + 1$ which lets us construct the finite field $\mathbb{F}_{2^2}$. The bases σ and τ are then given by $\sigma_1 = 1, \sigma_2 = \theta + 1, \tau_1 = \theta, \tau_2 = 1$. For each non-diagonal subspace α of the orthogonal decomposition there are $2^2 - 1 = 3$ basis matrices indexed by β (with $\beta \neq 0$). All the calculated basis matrices are presented in the tables below.

α	β	$\mathcal{J}_{\alpha\beta,\beta}$
0	1	$J_{0,0} \otimes J_{0,1}$
0	θ	$J_{0,1} \otimes J_{0,0}$
0	$\theta + 1$	$J_{0,1} \otimes J_{0,1}$
1	1	$J_{1,0} \otimes J_{0,1}$
1	θ	$J_{1,1} \otimes J_{1,0}$
1	$\theta + 1$	$J_{0,1} \otimes J_{1,1}$

α	β	$\mathcal{J}_{\alpha\beta,\beta}$
θ	1	$J_{1,0} \otimes J_{1,1}$
θ	θ	$J_{0,1} \otimes J_{1,0}$
θ	$\theta + 1$	$J_{1,1} \otimes J_{0,1}$
$\theta + 1$	1	$J_{0,0} \otimes J_{1,1}$
$\theta + 1$	θ	$J_{1,1} \otimes J_{0,0}$
$\theta + 1$	$\theta + 1$	$J_{1,1} \otimes J_{1,1}$

$$J_{0,0} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad J_{1,0} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

$$J_{0,1} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = S_2^{-1} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} S_2, \quad S_2^{-1} = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$J_{1,1} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} = S_1^{-1} \begin{pmatrix} -\iota & 0 \\ 0 & \iota \end{pmatrix} S_1, \quad S_1^{-1} = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ -\iota & \iota \end{pmatrix}$$

In order to construct the nice change of basis matrices T_α we need to simultaneously diagonalize the three basis matrices associated with α in the table (one for each β). The general idea is to parameterize all possible diagonalizations (represented by X, Y and Z below) for each basis matrix individually, in line with the discussion about multiplicity of eigenvalues, and then find instances where these diagonalizations are equal. We will do so for $\alpha = 1$ as an example.

$$J_{1,0} \otimes J_{0,1} = (I \otimes S_2^{-1}) \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} (I \otimes S_2) =$$

$$X^{-1} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} X, \quad X^{-1} = \frac{1}{2} \begin{pmatrix} x_1 & x_3 & x_5 & x_7 \\ x_1 & -x_3 & -x_5 & x_7 \\ x_2 & x_4 & x_6 & x_8 \\ -x_2 & x_4 & x_6 & -x_8 \end{pmatrix}$$

$$J_{1,1} \otimes J_{1,0} = (S_1^{-1} \otimes I) \begin{pmatrix} -\iota & 0 & 0 & 0 \\ 0 & \iota & 0 & 0 \\ 0 & 0 & \iota & 0 \\ 0 & 0 & 0 & -\iota \end{pmatrix} (S_1 \otimes I) =$$

$$Y^{-1} \begin{pmatrix} \iota & 0 & 0 & 0 \\ 0 & -\iota & 0 & 0 \\ 0 & 0 & \iota & 0 \\ 0 & 0 & 0 & -\iota \end{pmatrix} Y, \quad Y^{-1} = \frac{1}{2} \begin{pmatrix} y_4 & y_1 & y_6 & y_7 \\ y_3 & y_2 & y_5 & y_8 \\ \iota y_4 & -\iota y_1 & \iota y_6 & -\iota y_7 \\ -\iota y_3 & \iota y_2 & -\iota y_5 & \iota y_8 \end{pmatrix}$$

$$J_{0,1} \otimes J_{1,1} = (S_2^{-1} \otimes S_1^{-1}) \begin{pmatrix} -\iota & 0 & 0 & 0 \\ 0 & \iota & 0 & 0 \\ 0 & 0 & \iota & 0 \\ 0 & 0 & 0 & -\iota \end{pmatrix} (S_2 \otimes S_1) =$$

$$Z^{-1} \begin{pmatrix} -\iota & 0 & 0 & 0 \\ 0 & -\iota & 0 & 0 \\ 0 & 0 & \iota & 0 \\ 0 & 0 & 0 & \iota \end{pmatrix} Z, \quad Z^{-1} = \frac{1}{4} \begin{pmatrix} z_1 + z_2 & z_7 + z_8 & z_5 + z_6 & z_3 + z_4 \\ \iota(-z_1 + z_2) & \iota(-z_7 + z_8) & \iota(z_5 - z_6) & \iota(z_3 - z_4) \\ z_1 - z_2 & z_7 - z_8 & z_5 - z_6 & z_3 - z_4 \\ \iota(-z_1 - z_2) & \iota(-z_7 - z_8) & \iota(z_5 + z_6) & \iota(z_3 + z_4) \end{pmatrix}$$

Note how we reorder the eigenvalues for two of the Kronecker products in order to get independent diagonal matrices, which is necessary since these must form a basis of $\mathrm{sd}_n(\mathbb{C})$. Now it's only a simple matter of solving the linear system from $T_1^{-1} = X^{-1} = Y^{-1} = Z^{-1}$ with any nondegenerate solution. For example:

$$\begin{cases} x_1 = 1/2, x_2 = \iota/2, y_3 = 1/2, y_4 = 1/2, \\ z_1 = (1+\iota)/2, z_2 = (1-\iota)/2, \\ x_3 = -\iota/2, x_4 = -1/2, y_1 = -\iota/2, y_2 = \iota/2, \\ z_7 = (-1-\iota)/2, z_8 = (1-\iota)/2, \\ x_5 = 1/2, x_6 = \iota/2, y_5 = -1/2, y_6 = 1/2, \\ z_5 = (1+\iota)/2, z_6 = (1-\iota)/2, \\ x_7 = -\iota/2, x_8 = -1/2, y_7 = -\iota/2, y_8 = -\iota/2, \\ z_3 = (-1-\iota)/2, z_4 = (1-\iota)/2, \end{cases} \implies T_1^{-1} = \frac{1}{4} \begin{pmatrix} 1 & -\iota & 1 & -\iota \\ 1 & \iota & -1 & -\iota \\ \iota & -1 & \iota & -1 \\ -\iota & -1 & \iota & 1 \end{pmatrix}$$

We repeat this whole procedure for the other subspaces to also find the remaining three change of basis matrices. Actually we have found their inverses, but inverting them again is easily accomplished since they are nice. In conclusion, the following four matrices are nice change of basis matrices for the subspaces of an orthogonal decomposition when $n = 4$.

$$T_0 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}, \quad T_1 = \begin{pmatrix} 1 & 1 & -\iota & \iota \\ \iota & -\iota & -1 & -1 \\ 1 & -1 & -\iota & -\iota \\ \iota & \iota & -1 & 1 \end{pmatrix}$$

$$T_\theta = \begin{pmatrix} 1 & \iota & -1 & \iota \\ 1 & -\iota & 1 & \iota \\ 1 & -\iota & -1 & -\iota \\ 1 & \iota & 1 & -\iota \end{pmatrix}, \quad T_{\theta+1} = \begin{pmatrix} 1 & \iota & \iota & -1 \\ 1 & -\iota & \iota & 1 \\ 1 & \iota & -\iota & 1 \\ 1 & -\iota & -\iota & -1 \end{pmatrix}$$

Chapter 4

Conclusion

We have shown that it is possible to describe and work with orthogonal decompositions by their change of basis matrices (S_i) instead of their component subspaces (V_i). This gives us another perspective on the problem where matrices, rather than linear spaces, are the principal objects we deal with. Not only is this representation more compact, but it can be argued that individual matrices are more intuitive and easier to work with than linear spaces of them.

The change of basis matrices will have a form which we call *nice*. We have proved several basic properties that this form has and provided Hadamard and Fourier matrices as examples. Furthermore, we have defined an equivalence between nice matrices that only differ by monomial matrix factors, and shown that up to this equivalence all nice matrices of size 2 and 3 are the Fourier matrix of that size.

Finally we have provided constructions of orthogonal decompositions in terms of the change of basis matrices. For prime sizes ($n = p$) we have even given a general method. Designing a similar general method for prime power sizes ($n = p^m$), which would cover all known orthogonal decompositions, appeared to be much more difficult. However, we looked at a specific example of such a size (namely $n = 2^2$) which gives some insight into how the change of basis matrices for orthogonal decompositions of size $n = p^m$ can in principle be built from corresponding matrices of size $n = p$.

4.1 Acknowledgements

Some of the results given in chapter 2 are also found in [3]. This was not known at the time of writing and for this reason it is presented as independent work. The basic idea that change of basis matrices in an orthogonal decomposition will be nice was, however, presented by supervisor Victor Ufnarovski in conversation.

The construction attributed to Andersson [1] in section 3.3 is also given in an equivalent form in the original source by Kostrikin et al [3]. It's the presentation, and not the original idea, which is attributed to Andersson.

4.2 Future work

Three suggestions for future investigations have been identified.

Simplified construction for the case n is prime

The construction given in theorem 10 is quite complicated and inelegant, which can be attributed to the fact that it is just a reframing of another construction which primarily had in mind a simplicity for the basis matrices of the component subspaces. Is it possible to make a similarly neat construction in terms of the change of basis matrices? In particular, are the instances of $\phi(k)$ and $\triangle_j$ necessary or can they be replaced with something that is easier to work with?

Exhaustive search for orthogonal decompositions

In the first two sections of chapter 3 we found relatively simple constraints by exploiting symmetries in nice matrices which allowed us to find orthogonal decompositions for sizes $n = 2$ and $n = 3$. Our current tools developed in chapter 2 are, however, not enough to provide as simple constraints for larger n. For example, we can not anymore assume that right-monomials are diagonal.

With more extensive tools we might learn about more symmetries which could allow us to significantly decrease the time-complexity of an algorithm that looks in the complete search space of orthogonal decompositions and thus verifies whether one exists or not for a given size. Perhaps this could be made practical for $n = 6$ which could finally solve the Winnie-the-Pooh problem.

One thing in particular that seems to be unanswered in prior work is whether the subspaces of an orthogonal decomposition (and thus their change of basis matrices) need to be solved for simultaneously, or if they may be produced one at a time as is the case when producing orthogonal vectors. It would greatly decrease its time-complexity if it is possible to prove that a searching algorithm need not backtrack from a partial orthogonal decomposition.

General method for the case n is prime power

Although we dismissed it as too difficult, it should in principle be possible to give a general method to construct any orthogonal decomposition of prime power size in terms of change of basis matrices. We already know that the Kronecker products of $J_{\star,1}$ matrices need to be in different subspaces, but we also need to find the right diagonalization (that implicitly also simultaneously diagonalizes everything else in the same subspace).

This can perhaps be accomplished like this: The general idea is again to parameterize all possible diagonalizations for each Kronecker product. One would then solve for appropriate values to the parameters so that the conditions given by theorem 1 are satisfied. When expressing these values, it would probably be necessary to assume access to the coefficients of the irreducible polynomial Q.