Constructing Quantum Metric Spaces of Lie Type for Quantum Error Detection

By

Cassandra Hopkin
SENIOR THESIS

Submitted in partial satisfaction of the requirements for Highest Honors for the degree of

BACHELOR OF SCIENCE

in

MATHEMATICS AND COMPUTER SCIENCE

in the

COLLEGE OF LETTERS AND SCIENCE

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

_____

Greg Kuperberg

June 2025

Contents

Abstract

Error detection is necessary for any realistic model of computation. We can think of classical error detection and correction as a sphere packing problem in a classical metric space, and we present the analogous definition of quantum metric spaces for quantum error detection. Using this definition, we can find quantum codes as subspaces of quantum metric spaces, and we can use the KLV method to find subspaces that satisfy the error detection condition. Finally, we create a quantum metric space of Lie type using irreducible representations of the Lie algebra $\mathfrak{sl}(2, \mathbb{C})$, and find distance three codes using an optimized version of the KLV method.

Acknowledgments

I would like to thank my faculty mentor Greg Kuperberg for his enduring support and advice while completing this senior thesis. I would also like to thank Sanchayan Dutta, a PhD student who is also working with Professor Kuperberg, for discussing various ideas and directions for this project. I would also like to acknowledge Bella Finkel and Ian Shor's REU project reports, which have been incredibly useful references.

CHAPTER 1

Introduction: Quantum Codes in Quantum Metric Spaces

Quantum error detection and correction are vital for quantum computing to work, since qubits are vulnerable to decoherence. The problem of finding error detecting/correcting codes given a finite number of bits can be recast as finding subsets of a finite metric space. Using Kuperberg and Weaver's [**KW12**] definition of a quantum metric space, quantum codes analogously arise as subspaces. We are especially interested in quantum metric spaces of Lie type, which have symmetry properties that make it easier to find large codes. We use the two-stage construction proposed by Knill, Laflamme, and Viola [**KLV00**], which we call the KLV method, to construct quantum codes. In previous work by Finkel [**Fin22**] and Shor [**Sho22**], they were able to optimize the KLV construction for specific quantum metric spaces of Lie type with distance two. However, distance two codes have limited utility since they are only able to detect errors, not correct them. Therefore, we aim to optimize the KLV construction using similar techniques to create codes with distance three.

1.1. Error Detection and Correction

Information is subject to noise that can corrupt data, so to account for this we need to encode our data along with some form of redundancy. We call our initial string a *word*, and we call its encoded form a *codeword*. Assuming we have a fixed number of bits or qubits to work with, the number of codewords is the *size* of the code. The minimal number of errors to transform one codeword to another is the *minimum distance* of the code, which we sometimes refer to simply as the *distance* of the code. The 3-repetition code is a simple classical example to illustrate these ideas.

Example 1.1. Suppose we want to encode bit strings of length nine in such a way that we can correct single bit-flip errors. To do this, we map 0 to 000 and 1 to 111. For instance, we can encode the string 010 as 000111000. There are 2^3 binary strings of length three, so we can only make eight codewords, and this is the size of our code (notice that this is much smaller than the original space of 2^9 words of length nine). The distance of our code is three, since it takes three bit flip errors to transform one codeword to another. We call each triple in our codeword a *logical bit*. Within one logical bit, we can detect if one or two bit-flip errors occurred by checking if all of the bits in the triple are the same. If we know that at most one bit-flip occurred within each triple, then we can correct errors using majority vote (we flip the bit that does not agree with the other two). However, this means that although we can detect two errors in the same triple, if we try to use this method to correct two bit-flips we will change the string to the wrong codeword.

We next recall the definition of a (classical) metric space.

Definition 1.2 (Metric Space)**.** Let S be a set, and let $d : S \times S \to \mathbb{R}_{\geq 0}$ be a function such that for any $x, y, z \in S$ the following properties hold:

$$\begin{aligned} d(x, y) = 0 &\iff x = y \\ d(x, y) &= d(y, x) \\ d(x, z) &\leq d(x, y) + d(y, z) \end{aligned}$$

Then d is called a *metric* and (S, d) is called a *metric space*.

We can view the space of all words of length n as a metric space, where we have some notion of the distance between two words. We define an error detecting/correcting code as a subset, and define the minimum distance of the code to be the smallest distance between two points in our subset using the given metric. The following example describes the usual metric space used for classical error detection and correction.

Example 1.3 (Classical Hamming Space)**.** Let the *Hamming Distance* d_h between two words be the number of bits that differ between them. For example,

$$d_h(10000, 00101) = 3.$$

We can check that the Hamming distance is a valid metric by checking the conditions given in the definition above. If x, y, and z are any bit strings of length n then the Hamming distance between x and y is zero if and only if they are the same string, so

$$d_h(x, y) = 0 \iff x = y.$$

We also know that the number of bits that differ between x and y must be the same between y and x so

$$d_h(x, y) = d_h(y, x).$$

If x and z differ by i bits, x and y differ by j bits, and y and z differ by k bits, then x and z cannot differ by more than $j + k$ bits so we have that

$$d_h(x, z) \leq d_h(x, y) + d_h(y, z).$$

Therefore d_h is a metric. We call the space generated by this metric *classical Hamming space*.

Most classical codes arise as subsets of classical Hamming space, which allows us to think of constructing error detecting and correcting codes as a sphere packing problem. This is because if we have a fixed minimum distance d, maximizing the size of our code corresponds to choosing a maximal subset of points at least distance d apart. If we create balls of radius $d/2$ around each codeword, then this is the same as finding the optimal sphere packing in this metric space with spheres of radius $d/2$. Once we find this maximal subset, we are able to detect up to $d - 1$ errors, since those do not map sphere centers to other sphere centers. We can correct up to $(d-1)/2$ errors by assigning a corrupted word to the center of the sphere it lies in. Later, we provide the analogous definition of quantum metric spaces, where quantum codes arise as subspaces.

1.2. Quantum Information Background

Error detection and correction are especially important for quantum computing, since interactions with the environment introduce noise into a quantum system at much higher rates than for classical computing. We next review some of the foundations of quantum information theory. A *qubit* is the basic unit of information used in quantum computing, and it is the analogue of a classical bit. If we have a one qubit system, we can represent the qubit's state as a vector in $\mathbb{C}^2$, and we utilize Dirac's compact bra-ket notation as defined below.

Definition 1.4. (Bra-ket notation) We denote a vector v by $|v\rangle$, and call this a *ket*. We denote the dual vector of v by $\langle v|$ and call this a *bra*. Then the Hermitian inner product of two vectors $|v\rangle$ and $|w\rangle$ is $\langle w|v\rangle$ (which is a bra-ket!).

Let $|v\rangle$ and $|w\rangle$ be an orthonormal basis of $\mathbb{C}^2$. Then if we have one qubit, we can use bra-ket notation to represent its state $|\psi\rangle$ by $|\psi\rangle = a|v\rangle + b|w\rangle$ where $a, b \in \mathbb{C}$ and $|a|^2 + |b|^2 = 1$. We call a and b *quantum amplitudes*. We identify states that differ by a *global phase*, meaning that if $|\psi_1\rangle = a|v\rangle + b|w\rangle$ and $|\psi_2\rangle = e^{i\theta}(a|v\rangle + b|w\rangle)$ where $e^{i\theta} \in \mathbb{C}$, then $|\psi_1\rangle \sim |\psi_2\rangle$. Notice, however, that if two states differ by a *relative phase*, for example if $|\psi_1\rangle = a|v\rangle + b|w\rangle$ and $|\psi_2\rangle = a|v\rangle + e^{i\theta}b|w\rangle$ where $e^{i\theta} \neq 1$, then $|\psi_1\rangle \nsim |\psi_2\rangle$. Since the norms of the quantum amplitudes add up to 1, this means that our state vectors are actually in S^3, and when we quotient by the global phase equivalence we get $S^3/(|\psi\rangle \sim e^{i\theta}|\psi\rangle) \cong S^2$. This means that we can plot the state of a single qubit on a real unit sphere which we call the *Bloch sphere*. In general, if we have an n-qubit system, then the state space is $\mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2 = (\mathbb{C}^2)^{\otimes n}$, where $\otimes$ is the *tensor product*.

Below, we define the *Pauli matrices*, which are used throughout the quantum error correction literature. We will reference these matrices again in Section 1.5.

Definition 1.5. (Pauli Matrices)

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \qquad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

These matrices are important because they are each conceptually equivalent to a bit flip error, just in different bases. Additionally, these matrices, together with the identity error I, span the entire error space for one qubit. Below, we illustrate the effect of these operators in the $|0\rangle, |1\rangle$ basis, where $|\psi\rangle = a|0\rangle + b|1\rangle$.

$$X|\psi\rangle = a|1\rangle + b|0\rangle \qquad Z|\psi\rangle = a|0\rangle - b|1\rangle \qquad Y|\psi\rangle = ia|1\rangle - ib|0\rangle$$

Notice that we can write $Y|\psi\rangle = a|1\rangle - b|0\rangle$, by the global phase equivalence of states. Thus, in this basis, X corresponds to a bit flip error, Z corresponds to a phase flip error, and Y corresponds to a bit and phase flip error.

1.3. Review of Relevant Lie Theory

We next review some Lie theory that will allow us to later define quantum metric spaces of Lie type. We begin with the definitions of a Lie group and a Lie algebra, taken from Hall's book on the topic [**Hal15**].

Definition 1.6 (Lie Group)**.** A *Lie group* is a smooth manifold that is also a group, such that the group operation $G \times G \to G$ and the inverse map $G \to G$ are smooth.

Some specific Lie groups we are interested in are the general linear group $\mathrm{GL}(d, \mathbb{C})$, the special linear group $\mathrm{SL}(d, \mathbb{C})$, the unitary group $\mathrm{U}(d)$, and the special unitary group $\mathrm{SU}(d)$. $\mathrm{GL}(d, \mathbb{C})$ consists of all of the $d \times d$ invertible complex matrices, and the subgroup of $\mathrm{GL}(d, \mathbb{C})$ with determinant 1 is $\mathrm{SL}(d, \mathbb{C})$. Recall that a complex matrix is *unitary* if its columns are orthonormal. Then the group of $d \times d$ complex unitary matrices is another Lie group, and is called $\mathrm{U}(d)$. The subgroup of $\mathrm{U}(d)$ of matrices with determinant 1 is $\mathrm{SU}(d)$.

Definition 1.7 (Lie Algebra)**.** A finite real (or complex) *Lie algebra* is a finite dimensional real (or complex) vector space $\mathfrak{g}$ with a map $[\cdot, \cdot]$ called the *Lie bracket*, such that the following properties hold:

(1) $[\cdot, \cdot]$ is bilinear
(2) $[X, Y] = -[Y, X]$ for all $X, Y \in \mathfrak{g}$
(3) $[X, [Y, Z]] + [Y, [Z, X]] + [Z, [X, Y]] = 0$ for all $X, Y, Z \in \mathfrak{g}$

Although these definitions look dissimilar, every Lie group has a corresponding Lie algebra, which is the tangent space of the Lie group at the identity. In general, it is easier to work with the Lie algebra of a Lie group than with the group itself, due to the Lie algebra's linear structure. We are mainly concerned with matrix Lie groups and algebras, which we define below.

Definition 1.8 (Matrix Lie Group)**.** A *matrix Lie group* is a subgroup G of $\mathrm{GL}(d, \mathbb{C})$, such that for any sequence of matrices A_m in G that converge to some matrix A, we have that $A \in G$ or $A \notin \mathrm{GL}(d, \mathbb{C})$.

In fact, all of the Lie groups discussed above are matrix Lie groups! Next, we give the following definition for the Lie algebra of a matrix Lie group.

Definition 1.9 (Lie Algebra of a Matrix Lie Group)**.** Let G be a matrix Lie group. Then the Lie algebra of G, which we call $\mathfrak{g}$, is the set of all matrices X such that e^{tX} is in G for all $t \in \mathbb{R}$, where e^{tX} is the matrix exponential of X.

This definition implies that the Lie algebra of $\mathrm{GL}(d, \mathbb{C})$, which we call $\mathfrak{gl}(d, \mathbb{C})$, is all $d \times d$ complex matrices. This is because for any $X \in M(d, \mathbb{C})$, the matrix exponential e^{tX} has inverse e^{-tX} since $e^{tX}e^{-tX} = e^{\mathbf{0}} = I$ so $e^{tX} \in \mathrm{GL}(d, \mathbb{C})$. Next, we want to show that $\mathfrak{sl}(d, \mathbb{C})$, the Lie

algebra of $\mathrm{SL}(d, \mathbb{C})$, is all $d \times d$ complex matrices X such that $\mathrm{tr}(X) = 0$. To do this, we first need the following proposition.

Proposition 1.10. For any matrix $X \in M(d, \mathbb{C})$ we have that $\det(e^X) = e^{\mathrm{tr}(X)}$.

Proof. Suppose that X is diagonalizable with diagonal entries $\lambda_1, \lambda_2, ..., \lambda_d$. Then this implies that e^X is also diagonalizable with diagonal entries $e^{\lambda_1}, e^{\lambda_2}, ..., e^{\lambda_d}$. Since e^X is diagonal, we know the determinant of e^X is the product of its diagonal entries, so

$$\det(e^X) = e^{\lambda_1} \cdots e^{\lambda_d} = e^{\sum_{i=1}^{d} \lambda_i}.$$

We recognize that $\sum_{i=1}^{d} \lambda_i = \mathrm{tr(X)}$ and therefore $\det(e^X) = e^{\mathrm{tr}(X)}$. If X is not diagonalizable, we can represent it as the limit of a sequence of diagonal matrices, and since the property holds for each of the diagonal matrices, it must also hold for X. □

This proposition implies that for any $t \in \mathbb{R}$, $\det(e^{tX}) = e^{\mathrm{tr}(tX)}$. If $\mathrm{tr}(X) = 0$ then

$$e^{\mathrm{tr}(tX)} = e^0 = 1$$

so $e^{tX} \in \mathrm{SL}(d, \mathbb{C})$. We also know that if $e^{tX} \in \mathrm{SL}(d, \mathbb{C})$ for every real t, then

$$\det(e^{tX}) = 1 = e^{\mathrm{tr}(tX)}.$$

Since the trace is linear, $\mathrm{tr}(tX) = t \cdot \mathrm{tr}(X)$. We can take the derivative of $e^{t \cdot \mathrm{tr}(X)} = 1$ with respect to t to get

$$\mathrm{tr}(X) \cdot e^{t \cdot \mathrm{tr}(X)} = 0.$$

Since $e^{t \cdot \mathrm{tr}(X)} \neq 0$, we must have that $\mathrm{tr}(X) = 0$. Therefore, the matrices $X \in \mathfrak{sl}(d, \mathbb{C})$ are exactly the complex $d \times d$ matrices such that $\mathrm{tr}(X) = 0$.

Next, we are interested in the Lie algebras of $\mathrm{U}(d)$ and $\mathrm{SU}(d)$. We first show that the Lie algebra of $\mathrm{U}(d)$, denoted $\mathfrak{u}(d)$, is all complex matrices X such that $X^* = -X$. If we know that $X^* = -X$, then this implies that for any $t \in \mathbb{R}$ we have that

$$(e^{tX})^* = e^{tX^*} = e^{-tX} = (e^{tX})^{-1}$$

so e^{tX} is unitary[1]. Combining our reasoning for $\mathfrak{sl}(d, \mathbb{C})$ and $\mathfrak{u}(d)$ we get that $\mathfrak{su}(d)$ is composed of all complex matrices X such that $X^* = -X$ and $\mathrm{tr}(X) = 0$.

Another construction that is relevant is the complexification of a real Lie algebra. We recall the definition of the complexification of a real vector space, and then extend this construction to Lie groups and algebras.

Definition 1.11 (Complexification of a Real Vector Space)**.** If V is a real vector space, then the complexification of V, denoted $V_{\mathbb{C}}$, is the direct sum $V \oplus iV$. This means that $V_{\mathbb{C}}$ is the vector space of formal linear combinations

$$v_1 + iv_2$$

where $v_1, v_2 \in V$. In order to make this a complex vector space, we define $i(v_1 + iv_2) = -v_2 + iv_1$.

[1]The "*" here denotes the conjugate transpose of the matrix, which is also known as the Hermitian transpose. In the physics and QCQI literature, the dagger "†" is used in place of the star, but I have decided to use the star notation to match the later use of $*$-algebras.

Proposition 1.12. Let $\mathfrak{g}$ be a real Lie algebra, where $\mathfrak{g}_{\mathbb{C}}$ is its complexification as a real vector space. Then there is a unique way to extend the Lie bracket such that $\mathfrak{g}_{\mathbb{C}}$ is a complex Lie algebra, and we call $\mathfrak{g}_{\mathbb{C}}$ the *complexification* of $\mathfrak{g}$. Suppose that $\mathfrak{g} \subseteq \mathfrak{gl}(d, \mathbb{C})$ and that for all nonzero $X \in \mathfrak{g}$ we have that $iX \notin \mathfrak{g}$. Then the complexification of $\mathfrak{g}$ is isomorphic to the subset of $d \times d$ complex matrices that can be expressed as $X + iY$ where $X, Y \in \mathfrak{g}$.

Using the above proposition and definitions, we can find the complexifications of $\mathfrak{u}(d)$ and $\mathfrak{su}(d)$. We know that if $X \in \mathfrak{u}(d)$, then $X^* = -X$. This implies that $(iX)^* = iX \neq -iX$, so $iX \notin \mathfrak{u}(d)$ when $X \neq \mathbf{0}$. Now suppose that X is any complex $d \times d$ matrix. Then we can write

$$X = \left(\frac{X - X^*}{2}\right) + i\left(\frac{X + X^*}{2i}\right).$$

Since

$$\left(\frac{X - X^*}{2}\right)^* = \frac{X^* - X}{2} = -\left(\frac{X - X^*}{2}\right)$$

and

$$\left(\frac{X + X^*}{2i}\right)^* = \frac{X^* + X}{-2i} = -\left(\frac{X + X^*}{2i}\right)$$

we have that

$$\frac{X - X^*}{2}, \frac{X + X^*}{2i} \in \mathfrak{u}(d).$$

By the above proposition, this implies that $\mathfrak{u}(d)_{\mathbb{C}} \cong \mathfrak{gl}(d, \mathbb{C})$.

Next, we want to find the complexification of $\mathfrak{su}(d)$. If $X \in \mathfrak{su}(d)$, we know that $X^* = -X$ and $\mathrm{tr}(X) = 0$. By similar reasoning as above, we can write any traceless matrix X as

$$X = \left(\frac{X - X^*}{2}\right) + i\left(\frac{X + X^*}{2i}\right)$$

where

$$\frac{X - X^*}{2}, \frac{X + X^*}{2i} \in \mathfrak{su}(d).$$

This is because if X has trace zero, then so does X^*, so $X - (X^*)$ and $X + X^*$ both have trace zero. Therefore, by the above proposition, $\mathfrak{su}(d)_{\mathbb{C}} \cong \mathfrak{sl}(d, \mathbb{C})$. This is important for us, since we can consider every representation of $\mathfrak{su}(d)_{\mathbb{C}}$ as a representation of $\mathfrak{sl}(d, \mathbb{C})$.

Next, we discuss some relevant representation theory, specifically focusing on the irreducible representations of $\mathfrak{sl}(2, \mathbb{C})$. A (finite-dimensional) *representation* of a group G is a group homomorphism $\rho : G \to \mathrm{GL}(V)$ for some vector space V, or equivalently, it is the group action of G on V, written $G \circlearrowright V$. We next define Lie algebra homomorphisms so that we can provide an analogous definition for a representation of a Lie algebra.

Definition 1.13 (Lie Algebra Homomorphism)**.** If $\mathfrak{g}$ and $\mathfrak{h}$ are two Lie algebras, then a linear map $\phi : \mathfrak{g} \to \mathfrak{h}$ is called a *Lie algebra homomorphism* if it preserves the Lie bracket, that is, if

$$\phi([X, Y]) = [\phi(X), \phi(Y)]$$

for all $X, Y \in \mathfrak{g}$.

A finite-dimensional complex representation of a Lie algebra $\mathfrak{g}$ is a Lie algebra homomorphism $\rho : \mathfrak{g} \to \mathfrak{gl}(V)$, where V is a finite-dimensional complex vector space. Now let ρ be a finite-dimensional complex representation of $\mathfrak{g}$ acting on a vector space V. If ρ fixes a subspace $W \subseteq V$, that is, $\rho(X)w \in W$ for all $w \in W$ and for all $X \in \mathfrak{g}$, then W is called an *invariant* subspace under ρ. We say that W is a *nontrivial* invariant subspace if $W \neq V$ and $W \neq \{0\}$. A representation ρ that has no nontrivial invariant subspaces is called *irreducible*. Similarly to how the Lie bracket extends uniquely to the complexification of a matrix Lie group, we have the following proposition for the extension of a representation of $\mathfrak{g}$ to its complexification.

Proposition 1.14. Let $\mathfrak{g}$ be a real Lie algebra and let $\mathfrak{g}_{\mathbb{C}}$ be its complexification. Then every finite dimensional complex representation ρ of $\mathfrak{g}$ can be uniquely extended to a complex-linear representation of $\mathfrak{g}_{\mathbb{C}}$, which we also call ρ. Furthermore, ρ is an irreducible representation of $\mathfrak{g}_{\mathbb{C}}$ if and only if it is also irreducible as a representation of $\mathfrak{g}$.

We next present a family of irreducible representations of $\mathfrak{sl}(2, \mathbb{C})$ and a theorem which states that these in fact describe all of the irreducible representations up to isomorphism.

Example 1.15 ($\mathfrak{sl}(2, \mathbb{C})$ Acting on a Vector Space of Homogeneous Polynomials)**.** We begin with the following basis of $\mathfrak{sl}(2, \mathbb{C})$:

$$E = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \qquad F = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \qquad H = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

with commutation relations

$$[H, E] = 2E \qquad [H, F] = -2F \qquad [E, F] = H.$$

Then let $\rho : \mathfrak{sl}(2, \mathbb{C}) \to \mathfrak{gl}(\mathbb{C}[x, y]_n)$ be an $n + 1$ dimensional representation, where $\mathbb{C}[x, y]_n$ is the vector space of homogeneous polynomials of degree n in x and y. Define ρ such that:

$$\rho(E) = y\frac{\partial}{\partial x} \qquad \rho(F) = x\frac{\partial}{\partial y} \qquad \rho(H) = y\frac{\partial}{\partial y} - x\frac{\partial}{\partial x}.$$

We first check that this is a representation. We have that $\rho[H, E] = 2y\frac{\partial}{\partial x}$ and

$$\begin{aligned}
[\rho(H), \rho(E)] &= \left[y\frac{\partial}{\partial y} - x\frac{\partial}{\partial x}, y\frac{\partial}{\partial x} \right] \\
&= \left(y\frac{\partial}{\partial y} - x\frac{\partial}{\partial x} \right)\left(y\frac{\partial}{\partial x} \right) - \left(y\frac{\partial}{\partial x} \right)\left(y\frac{\partial}{\partial y} - x\frac{\partial}{\partial x} \right) \\
&= y\frac{\partial}{\partial x} + y^2\frac{\partial^2}{\partial xy} - xy\frac{\partial^2}{\partial xy} - y^2\frac{\partial^2}{\partial xy} + y\frac{\partial}{\partial x} + xy\frac{\partial^2}{\partial xy} \\
&= 2y\frac{\partial}{\partial x}.
\end{aligned}$$

We can check the other two commutation relations using similar calculations. Since ρ preserves the Lie bracket, it is a representation. We next analyze the effect of these operators on the basis elements of $\mathbb{C}[x, y]_n$. We have that

$$\rho(E)(x^{n-k}y^k) = (n-k)x^{n-k-1}y^{k+1}$$
$$\rho(F)(x^{n-k}y^k) = (k)x^{n-k+1}y^{k-1}$$

and

$$\begin{aligned}\rho(H)(x^{n-k}y^k) &= (k)x^{n-k}y^k - (n-k)x^{n-k}y^k \\ &= (2k-n)x^{n-k}y^k.\end{aligned}$$

Figure 1.1 shows the effects of $\rho(E), \rho(F)$, and $\rho(H)$ on the basis elements of $\mathbb{C}[x, y]_6$.

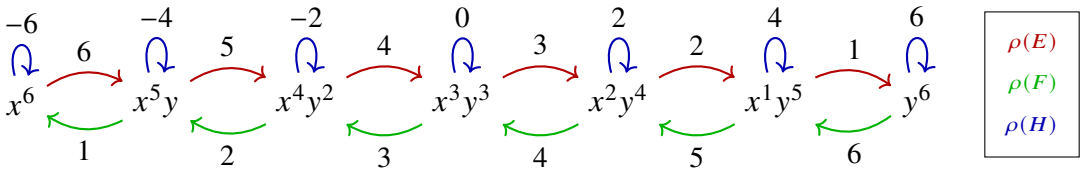
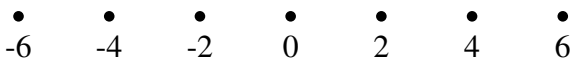


FIGURE 1.1. Weight diagram of ρ when $n = 6$. The arrows are labeled with the coefficient the basis vector picks up from the operation.

We define a *weight* of an element $x^{n-k}y^k$ to be the eigenvalue λ of $\rho(H)$, i.e.

$$\rho(H)(x^{n-k}y^k) = \lambda(x^{n-k}y^k).$$

Then $x^{n-k}y^k$ is the corresponding *weight vector*. For a given weight λ, its *weight space* is the subspace of V spanned by λ's weight vectors. We can plot each weight space labeled with its corresponding weight to create a *weight diagram* for our representation ρ. For example, if we have $n = 6$, then the weight diagram for ρ would look like the following:



Proposition 1.16. For every $n \geq 1$, the representation ρ of $\mathfrak{sl}(2, \mathbb{C})$ given above is irreducible.

PROOF. Let $W \subseteq \mathbb{C}[x, y]_n$ be a nonzero invariant subspace under ρ, which means that $\rho(X)w \in W$ for all $w \in W$ and for all $X \in \mathfrak{sl}(2, \mathbb{C})$. Let

$$w = c_0x^n + c_1x^{n-1}y + \ldots + c_{n-1}xy^{n-1} + c_ny^n$$

be a nonzero element of W, and let $c_ix^{n-i}y^i$ be the nonzero term of w such that i is smallest. Then if we repeatedly apply $\rho(E)$ to w, eventually all of the other terms of w get sent to 0. Specifically, we have that

$$\rho(E)^{n-i}w = a(c_iy^n)$$

for some coefficient a. Since W is invariant under $\rho(E)$, we know that $a(c_i y^n) \in W$, so $y^n \in W$. Then we can repeatedly apply $\rho(F)$ to y^n to get every other basis element of $\mathbb{C}[x, y]_n$. Since W contains all of the basis elements of $\mathbb{C}[x, y]_n$, we must have $W = \mathbb{C}[x, y]_n$. Therefore there are no nontrivial invariant subspaces of $\mathbb{C}[x, y]_n$ under ρ, so ρ is irreducible. □

Lastly, we provide a theorem which states that in fact, every $(n+1)-$dimensional irreducible representation of $\mathfrak{sl}(2, \mathbb{C})$ is isomorphic to the $(n+1)-$dimensional representation $\mathbb{C}[x, y]_n$ described above.

Theorem 1.17. For every integer $n \geq 0$, there is an irreducible representation of $\mathfrak{sl}(2, \mathbb{C})$ of dimension $n + 1$, and all irreducible representations of $\mathfrak{sl}(2, \mathbb{C})$ of the same dimension are isomorphic. If ϕ is an irreducible representation of $\mathfrak{sl}(2, \mathbb{C})$ of dimension $n + 1$, then it is isomorphic to $\rho : \mathfrak{sl}(2, \mathbb{C}) \to \mathbb{C}[x, y]_n$, the $(n + 1)-$dimensional representation described above.

1.4. Quantum Metric Spaces

In this section, we state the definition of a quantum metric due to Kuperberg and Weaver [**KW12**]. Let the Hilbert space $\mathcal{H}$ be the state space of our system. Note that Hilbert spaces are necessary to define quantum metrics in the infinite dimensional case, but since the only Hilbert spaces we will need will be finite-dimensional, we can just think of the state space as $\mathbb{C}^N$ for some $N \in \mathbb{N}$. The use of Hilbert spaces even in the finite-dimensional case is common in the QCQI literature. Then the linear operators on the state space $\mathcal{L}(\mathcal{H})$ form the error space of our system. Recall the definition of an algebra [**Gol95**].

Definition 1.18 (Algebra)**.** An algebra $\mathcal{A}$ is a vector space over a field F along with a bilinear multiplication operation $\mathcal{A} \times \mathcal{A} \to \mathcal{A}$ such that the following properties hold:

$$u(v + w) = uv + uw$$
$$(u + v)w = uw + vw$$
$$c(vw) = v(cw) = (cv)w$$

for all $u, v, w \in \mathcal{A}$ and $c \in F$.

$\mathcal{L}(\mathcal{H})$ is an algebra, in fact it is a $*$-*algebra*. This means that $\mathcal{L}(\mathcal{H})$ has a unary operator $*$ that is an involution, is conjugate linear, and is an anti-homomorphism. We also have the property that for $A \in \mathcal{L}(\mathcal{H})$, $A^*A = 0$ implies that $A = 0$. Now we are ready to define a quantum metric space, as given by Kuperberg and Weaver [**KW12**] [2].

Definition 1.19 (Quantum Metric Space)**.** A *quantum metric* on $\mathcal{A} \subseteq \mathcal{L}(\mathcal{H})$ is a $*$- *algebra filtration* of $\mathcal{L}(\mathcal{H})$. This means that the following properties hold:

(1) $\mathcal{V}_s \subseteq \mathcal{V}_t$ if $s < t$

[2]Conditions (3) and (4) in the definition of a quantum metric space are written in set arithmetic style, where we write $\mathcal{V}_t$ as shorthand for "for every element V in $\mathcal{V}_t$." We use this shorthand in various places in the text, but it is especially useful in sections 1.6 and 2.2.

(2) $\mathcal{V}_0 = \text{span}\{I\}$
(3) $\mathcal{V}_t = \mathcal{V}_t^*$
(4) $\mathcal{V}_s\mathcal{V}_t \subseteq \mathcal{V}_{s+t}$

$\mathcal{A}$ along with its quantum metric form a *quantum metric space*.

Since in this context we are only considering the case where $\mathcal{H}$ is finite-dimensional, we can consider integer values of t and think of $\{\mathcal{V}_t\}$ as a nested sequence $\mathcal{V}_0 \subseteq \mathcal{V}_1 \subseteq \cdots \subseteq \mathcal{V}_n$ as explained by Shor [**Sho22**].

Since the definition of a quantum metric space is very general, we are interested in working with quantum metrics with some additional structure. We next define a quantum graph metric, and then define quantum metric spaces of Lie type.

Definition 1.20 (Quantum Graph Metric)**.** Define the *edge space* $\mathcal{E}$ to be a subspace of $\mathcal{L}(\mathcal{H})$ such that $\mathcal{E} = \mathcal{E}^*$ and $\mathcal{E} \supseteq \text{span}\{I\}$. Then the corresponding *quantum graph metric* is a quantum metric with filtration $\{\mathcal{V}_t\}$ such that:

(1) $\mathcal{V}_1 = \mathcal{E}$
(2) $\mathcal{V}_t = \text{span}\{\underbrace{\mathcal{E} \cdots \mathcal{E} \cdot \mathcal{E}}_{t \text{ times}}\}$

When discussing quantum graph metrics, we use the terms "error space" and "edge space" interchangeably. We say that a quantum graph metric $\{\mathcal{V}_t\}$ is of *Lie type* if $\mathcal{E} = L \oplus \mathcal{V}_0$ where L is a self-adjoint traceless Lie algebra.

1.5. Example: Quantum Hamming Space

Building on our earlier example of classical Hamming space, we now present the analogous quantum metric space of Lie type called *quantum Hamming space*. Let $\mathcal{H} = (\mathbb{C}^2)^{\otimes n}$, that is, $\mathcal{H}$ is the state space for n qubits. Then to create a quantum metric of Lie type, we define our Lie algebra to be

$$L = \underbrace{\mathfrak{sl}(2,\mathbb{C}) \oplus \cdots \oplus \mathfrak{sl}(2,\mathbb{C}) \oplus \mathfrak{sl}(2,\mathbb{C})}_{n \text{ times}}$$

and $\mathcal{E}_1 = L \oplus \text{span}\{I\}$. Note that the Pauli matrices, given in Definition 1.5, form another basis of $\mathfrak{sl}(2,\mathbb{C})$. In addition, the matrices X, Y, Z, and I form a basis for $\mathfrak{gl}(2,\mathbb{C}) \cong M(2,\mathbb{C})$.

Therefore, we can write:

$$\mathcal{E}_t = \text{span}\{A_1 \otimes A_2 \otimes \cdots \otimes A_n \mid \text{at most t of the } A_i\text{'s} \in \{X, Y, Z\} \text{ and the rest are } I\}.$$

1.6. Quantum Error Detection

We can view quantum codes as subspaces of quantum metric spaces, by analogy with the classical case. We next want to know when we are able to detect an error E. Recall that a projection operator P is a self-adjoint idempotent operator, meaning $P = P^2 = P^*$. Then for a quantum code $\mathcal{C}$, the orthogonal projection $P_\mathcal{C}$ projects states $|\psi\rangle$ onto the subspace $\mathcal{C}$, that is, $P_\mathcal{C}|\psi\rangle \in \mathcal{C}$.

We have the following definitions of the error detection condition, the minimum distance, and the slope of a quantum code from Okada [**Oka23**]. We then provide a useful proposition for an equivalent formulation of the error detection condition, also from Okada. Lastly, we present a proposition that will be useful when constructing codes using the KLV method.

Definition 1.21 (Error Detection Condition)**.** Let $\mathcal{C}$ be a quantum code with orthogonal projection $P_{\mathcal{C}}$. $\mathcal{C}$ *detects* the error $E \in L(\mathcal{H})$ if there exists $\epsilon(E) \in \mathbb{C}$ such that $P_{\mathcal{C}} E P_{\mathcal{C}} = \epsilon(E) P_{\mathcal{C}}$.

Intuitively, this is saying that if for all $|\psi\rangle \in \mathcal{C}$ we have

$$E|\psi\rangle = re^{i\theta}|\psi\rangle \propto e^{i\theta}|\psi\rangle = |\psi\rangle$$

(the last equality holds since a state's global phase doesn't matter), then E wasn't really an error since it takes codewords to states proportional to their original state. If $E|\psi\rangle = re^{i\theta}|\psi\rangle$, then there is a linear functional ϵ such that $\epsilon(E) = re^{i\theta}$. On the other hand, if we have E such that $E|\psi\rangle \notin \mathcal{C}$ for all $|\psi\rangle \in \mathcal{C}$, then $P_{\mathcal{C}} E P_{\mathcal{C}}|\psi\rangle = 0$ and then $\epsilon(E) = 0$. Therefore, if this linear functional ϵ exists such that $P_{\mathcal{C}} E P_{\mathcal{C}} = \epsilon(E) P_{\mathcal{C}}$, then we can detect whether E was a logical error or not.

Definition 1.22 (Slope of a Quantum Code)**.** Let $\mathcal{C}$ be a quantum code of distance $d \geq 1$. The linear functional $\epsilon : \mathcal{E}_{d-1} \to \mathbb{C}$ is called the *slope* of $\mathcal{C}$.

Definition 1.23 (Minimum Distance of a Quantum Code)**.** Let $\mathcal{C}$ be a quantum code. The *minimum distance* (or just the *distance*) of $\mathcal{C}$ is the largest d such that $\mathcal{C}$ detects all errors in $\mathcal{E}_{d-1}$.

Proposition 1.24. Let $\mathcal{C}$ be a quantum code and let E be an error. Then the following two conditions are equivalent:

(1) $\mathcal{C}$ detects E
(2) For some orthonormal basis $\{|\psi_i\}_{i=1}^{\dim(\mathcal{C})}$, there is an $\epsilon(E) \in \mathbb{C}$ such that $\langle\psi_i|E|\psi_j\rangle = \epsilon(E)\delta_{ij}$ for all $i, j \in \{1 \ldots \dim(\mathcal{C})\}$

The next proposition will be particularly useful for the next section.

Proposition 1.25. Let $\mathcal{C}$ be a subspace of a Hilbert space $\mathcal{H}$ with error space $\mathcal{E}$, and let $\mathcal{B}$ be a subspace of $\mathcal{H}$ such that $\mathcal{C} \subseteq \mathcal{B} \subseteq \mathcal{H}$. In set arithmetic, if $\mathcal{C}$ detects $P_{\mathcal{B}} \mathcal{E} P_{\mathcal{B}}$, then $\mathcal{C}$ detects $\mathcal{E}$.

Proof. Since $\mathcal{C} \subseteq \mathcal{B}$, we know that $P_{\mathcal{B}} P_{\mathcal{C}} = P_{\mathcal{C}} P_{\mathcal{B}} = P_{\mathcal{C}}$. If $\mathcal{C}$ detects $P_{\mathcal{B}} \mathcal{E} P_{\mathcal{B}}$, then there exists a linear functional ϵ such that $P_{\mathcal{C}} P_{\mathcal{B}} \mathcal{E} P_{\mathcal{B}} P_{\mathcal{C}} = \epsilon(\mathcal{E}) P_{\mathcal{C}}$, which means that $P_{\mathcal{C}} \mathcal{E} P_{\mathcal{C}} = \epsilon(\mathcal{E}) P_{\mathcal{C}}$. Therefore $\mathcal{C}$ detects $\mathcal{E}$. □

In the next chapter, we discuss the KLV method for constructing quantum error detecting codes in quantum graph metric spaces, and how we can improve the KLV method to create a distance 3 code in a specific quantum metric of Lie type.

CHAPTER 2

KLV Method for Constructing Codes

2.1. Constructing Codes in Quantum Graph Metric Spaces

Knill, Laflamme, and Viola provide a general method to construct quantum codes in quantum graph metric spaces [**KLV00**]. This is a two stage construction, where we first create an intermediate code that detects some of the errors in the error space, and then find a subspace of this code to detect all of the errors in the error space. This is a type of *concatenated* code, meaning we construct our code by finding a subspace of an intermediate code (similar to Peter Shor's 9-qubit code).

2.1.1. First Stage of the KLV Method. Let $\mathcal{H}$ be the n-dimensional state space of our quantum system, and let $\mathcal{E}$ be our error space with some basis $E_1, E_2, ..., E_{\dim(\mathcal{E})}$. Let d be the desired minimum distance for our code. We want to find a set of orthonormal states that form a basis for a subspace $\mathcal{B}$, such that $\langle\psi_i|\mathcal{E}|\psi_j\rangle = \epsilon(\mathcal{E})\delta_{ij}$ for all $i, j \in \{1, ..., \dim(\mathcal{B})\}$. Then let $|\psi_1\rangle$ be any normalized state in $\mathcal{H}$. Choose $|\psi_2\rangle$ (normalized) such that $|\psi_2\rangle$ is orthogonal to $\mathcal{E}|\psi_1\rangle$, that is $|\psi_2\rangle \in (\mathcal{E}|\psi_1\rangle)^\perp$, which we can find if $\dim(\mathcal{E}) < n$. If we were able to find such a state, then make it a basis vector for $\mathcal{B}$. Then we can try to find a normalized state $|\psi_3\rangle \in (\mathcal{E}|\psi_1\rangle)^\perp \cap (\mathcal{E}|\psi_2\rangle)^\perp$, and can continue this process in this way until we can't add any more states. This results in an orthonormal basis for $\mathcal{B}$, and the remaining error space after we project to $\mathcal{B}$ is $\mathcal{E}_\mathcal{B} = P_\mathcal{B}\mathcal{E}P_\mathcal{B}$. The elements of $P_\mathcal{B}\mathcal{E}P_\mathcal{B}$ are diagonal by construction, and are therefore commutative. Using this greedy construction, we get the following lower bound for the dimension of $\mathcal{B}$:

$$\dim(\mathcal{B}) \geq \left\lceil \frac{\dim(\mathcal{H})}{\dim(\mathcal{E})} \right\rceil.$$

2.1.2. Second Stage of the KLV Method. Next, we want to find a subspace of $\mathcal{B}$ that detects $\mathcal{E}_\mathcal{B}$, and we do so using Tverberg's theorem.

Theorem 2.1 (Tverberg's Theorem)**.** Suppose we have n points in $\mathbb{R}^d$ such that $n \geq (d+1)(r-1)+1$ for some r. Then there exists a partition of the points into r parts such that the intersection of the convex hulls of the parts is nonempty.

The *convex hull* of a set of points is the smallest convex set containing all of the points. Intuitively, we can think about the convex hull in 2D space as the region obtained by stretching a rubber band around the set of points. Figure 2.1 shows an example of a Tverberg partition of ten points. Next, we use this theorem to partition the elements of $\mathcal{B}$, such that the convex hull of the parts is the slope of the code.

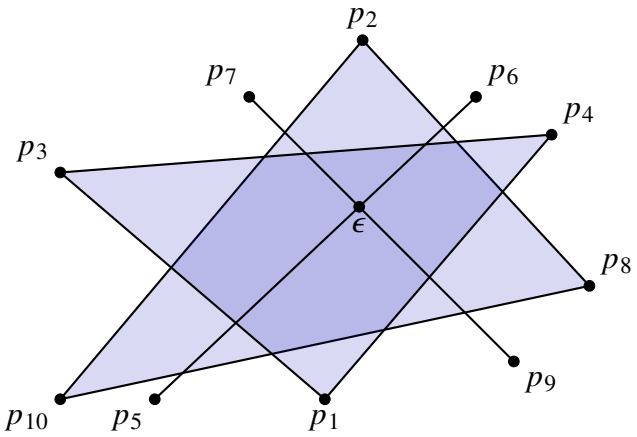


FIGURE 2.1. An example of a Tverberg partition of 10 points.

The details of the second stage of KLV were described by Bumgardner [**Bum03**], and we pull from these details in the following explanation. Since all of the errors in $\mathcal{E}_\mathcal{B}$ commute, this means that they are jointly diagonalizable, so we can choose a basis $|\psi_1\rangle, |\psi_2\rangle, ..., |\psi_k\rangle$ for $\mathcal{B}$ such that all of the errors in $\mathcal{E}_\mathcal{B}$ are diagonal. Let $E_1, ..., E_m$ be a basis for $\mathcal{E}_\mathcal{B}$. Then since each $E_l \in \mathcal{E}_\mathcal{B}$ is diagonal in this basis of $\mathcal{B}$, we have that there exist eigenvalues $\lambda_{l,i}$ such that

$$E_l|\psi_i\rangle = \lambda_{l,i}|\psi_i\rangle$$

for $1 \leq i \leq k$, and for all $E_l \in \mathcal{E}_\mathcal{B}$. The eigenvalue $\lambda_{l,i}$ is simply the ith diagonal entry of E_l.

Next, we want to construct states $|v\rangle$ such that their span satisfies the error detection condition. To do this, we first apply Tverberg's theorem to partition the indices of $|\psi_1\rangle, |\psi_2\rangle, ..., |\psi_k\rangle$ such that the convex hulls of the parts intersect at a point $\vec{\epsilon} = (\epsilon_1, ..., \epsilon_l)$. Notice that this is equivalent to partitioning the set of k eigenvalues $\{\lambda_{l,p}\}$ for a given E_l. Then for each of the parts X_j, define $|v_j\rangle$ to be the normalized linear combination of the states $|\psi_p\rangle$ in the part X_j such that

$$|v_j\rangle = \sum_{p \in X_j} \beta_{jp}|\psi_p\rangle \qquad \sum_{p \in X_j} |\beta_{jp}|^2 \lambda_{l,p} = \epsilon_l \qquad \sum_{p \in X_j} |\beta_{jp}|^2 = 1$$

for some coefficients β_{jp}. We know that we can find these coefficients because ϵ_l can be written as a convex combination of the $\lambda_{l,p}$'s.

Proposition 2.2. The span of the $|v_j\rangle$ states forms a code $\mathcal{C}$ such that $\mathcal{C}$ detects $\mathcal{E}_\mathcal{B}$.

PROOF. To prove this, we need to check that $\langle v_j|E_l|v_j\rangle = \epsilon_l$ and that $\langle v_i|E_l|v_j\rangle = 0$ when $i \neq j$, since this is equivalent to the second error detection condition given in Proposition 1.24. We first check that $\langle v_j|E_l|v_j\rangle = \epsilon_l$:

$$\begin{aligned}\langle v_j|E_l|v_j\rangle &= (\sum_{p\in X_j}\overline{\beta_{jp}}\langle\psi_p|)E_l(\sum_{p\in X_j}\beta_{jp}|\psi_p\rangle) = (\sum_{p\in X_j}\overline{\beta_{jp}}\langle\psi_p|)(\sum_{p\in X_j}\beta_{jp}E_l|\psi_p\rangle)\\ &= (\sum_{p\in X_j}\overline{\beta_{jp}}\langle\psi_p|)(\sum_{p\in X_j}\beta_{jp}\lambda_{l,p}|\psi_p\rangle) = (\sum_{p_1\in X_j}\sum_{p_2\in X_j}\overline{\beta_{jp_1}}\beta_{jp_2}\lambda_{l,p_2}\langle\psi_{p_1}|\psi_{p_2}\rangle)\\ &= \sum_{p\in X_j}|\beta_{jp}|^2\lambda_{l,p} = \epsilon_l\end{aligned}$$

We can do a similar calculation to show that $\langle v_i|E_l|v_j\rangle = 0$ when $i \neq j$, since in the third to last step the inner product will be 0 always since the partitions are disjoint. Therefore, $\mathcal{C}$ detects $\mathcal{E}_\mathcal{B}$ and we are done. □

Then this implies that

$$\dim(\mathcal{C}) \geq \frac{1}{\dim(\mathcal{E})+1}\left\lceil\frac{\dim(\mathcal{H})}{\dim(\mathcal{E})}\right\rceil.$$

2.2. Distance 3 KLV Construction for $\mathfrak{sl}(2,\mathbb{C})$

In previous work by Finkel [**Fin22**] and Shor [**Sho22**], they constructed quantum metric spaces of Lie type using the real Lie algebras $\mathfrak{su}(2)$ and $\mathfrak{su}(3)$. Since these Lie algebras act on a complex vector space, we can examine the natural action of their complexifications $\mathfrak{sl}(2,\mathbb{C})$ and $\mathfrak{sl}(3,\mathbb{C})$ on the Hilbert space. These Lie algebras are convenient for us because they have infinitely many irreducible representations that are also multiplicity-free, which makes optimizing the KLV method using the Lie algebra weight diagram more straightforward. Finkel and Shor found larger codes using an optimized form of the KLV method for distance $d = 2$. However, this distance is too small to correct errors. Therefore, we are interested in creating distance-three codes using a similar optimization.

Bumgardner analyzed the $\mathfrak{sl}(2,\mathbb{C})$ case for arbitrary distances and optimized the first stage of the KLV construction, but he did not optimize the second stage [**Bum03**]. Optimizing the second stage for distance two is easier than for higher distances, since in the distance two case the $\mathfrak{sl}(2,\mathbb{C})$ and $\mathfrak{sl}(3,\mathbb{C})$ weight diagrams match the weight diagram for $\mathcal{E}_\mathcal{B}$[3]. As a result, Finkel and Shor optimized the second stage of the KLV method directly using Lie algebra weight diagrams. We will see that the weight diagram of $\mathfrak{sl}(2,\mathbb{C})$ does not match the weight diagram of $\mathcal{E}_\mathcal{B}$ in the distance 3 case, but that the weight diagram of $\mathcal{E}_\mathcal{B}$ still has some special structure that we can use to improve the second stage. We provide an analysis for distance three codes in the quantum metric space of Lie type constructed using $\mathfrak{sl}(2,\mathbb{C})$ below.

Let $\mathcal{H}$ be an $(n+1)$-dimensional Hilbert space, and let $\mathcal{E}_2$ be the error space of distance three errors. Let $\rho : \mathfrak{sl}(2,\mathbb{C}) \to \mathcal{H}$ be an irreducible representation of $\mathfrak{sl}(2,\mathbb{C})$, which by Theorem 1.17 is isomorphic to the polynomial representation $\rho : \mathfrak{sl}(2,\mathbb{C}) \to \mathbb{C}[x,y]_n$. In this case, we have that

$$\mathcal{E}_1 = \text{span}\{I, \rho(E), \rho(F), \rho(H)\}.$$

[3]Note that the term "weight diagram" is most commonly used for weight diagrams of representations of semisimple Lie algebras. In this context, we use the term "weight diagram" to also mean the plot of the joint eigenvalues of the commutative errors in $\mathcal{E}_\mathcal{B}$.

As in Example 1.15, we can make a weight diagram for ρ and we can choose a subset of the weight spaces such that they are three error applications apart. For example, if $n = 6$ we can choose the following subset (indicated with red circles).

We next calculate the dimension of $\mathcal{E}_2$ in the distance three case. Then we discuss the lower bounds that the general KLV method guarantees, and how we can improve this by optimizing the first stage. Finally, we discuss what it means to optimize the second stage, and give an example of this in the next section.

2.2.1. Finding the Dimension of $\mathcal{E}_2$. Distance three means that we want to be able to detect all errors in

$$\mathcal{E}_2 = \text{span}\{I, \rho(E), \rho(F), \rho(H), \rho(E^2), \rho(EF), \rho(EH), \rho(F^2), \rho(FE), \rho(FH), \rho(H^2), \rho(HE), \rho(HF)\}$$

However, some of the terms in this span are redundant. Using the commutation relations between E, F, and H we can write $\rho(FE), \rho(HE)$, and $\rho(HF)$ as linear combinations of other elements already in the span:

$$\begin{aligned} [E, F] = EF - FE = H &\implies \rho(FE) = \rho(EF) - \rho(H) \\ [H, E] = HE - EH = 2E &\implies \rho(HE) = \rho(2E) + \rho(EH) \\ [H, F] = HF - FH = -2F &\implies \rho(HF) = \rho(FH) - \rho(2F). \end{aligned}$$

Therefore, we have:

$$\mathcal{E}_2 = \text{span}\{I, \rho(E), \rho(F), \rho(H), \rho(E^2), \rho(EF), \rho(EH), \rho(F^2), \rho(FH), \rho(H^2)\}$$

Since $\mathbb{C}[x, y]_n$ is an irreducible representation, we can observe that

$$\rho(EF)x^{n-k}y^k = (n - k + 1)(k)x^{n-k}y^k$$

and

$$-\frac{1}{4}\rho(H^2)x^{n-k}y^k + \frac{1}{2}\rho(H)x^{n-k}y^k + \frac{1}{4}n^2 + \frac{1}{2}n = (n - k + 1)(k)x^{n-k}y^k,$$

so

$$\rho(EF) = -\frac{1}{4}\rho(H^2) + \frac{1}{2}\rho(H) + \frac{1}{4}n^2 + \frac{1}{2}n.$$

Therefore, $\rho(EF)$ is already included in the span of the other elements. Then we can write:

$$\mathcal{E}_2 = \text{span}\{I, \rho(E), \rho(F), \rho(H), \rho(E^2), \rho(EH), \rho(F^2), \rho(FH), \rho(H^2)\}.$$

We next prove that these elements form a basis for $\mathcal{E}_2$, by showing that they are linearly independent.

Proposition 2.3. The elements listed in the span above are linearly independent when $n \geq 2$.

Proof. We can write each of the elements in the basis of monomials of degree n, ordered by decreasing powers of x and increasing powers of y. For example, if $n = 3$, then we would have the monomial basis x^3, x^2y, xy^2, y^3. Then each of the elements in the span above has a matrix form in this basis, where $I, \rho(H)$, and $\rho(H^2)$ have nonzero elements on the diagonal, $\rho(E)$ and $\rho(EH)$ have nonzero elements just below the diagonal, $\rho(F)$ and $\rho(FH)$ have nonzero entries just above the diagonal, $\rho(E^2)$ has nonzero entries just below where $\rho(E)$ does, and $\rho(F^2)$ has nonzero entries just above where $\rho(F)$ does. This means we can group the matrices into five groups where each group has disjoint support from the others. Therefore, we just need to show linear independence within each group. We can view the diagonal entries of $\rho(H^2)$ as a function of the diagonal entries of $\rho(H)$, such that

$$\rho(H^2)_{i,i} = \rho(H)_{i,i}^2.$$

We know that the monomials $1, \lambda$, and λ^2 are linearly independent as functions when evaluated on 3 or more points. When $n \geq 2$ we evaluate the function $f(\lambda) = \lambda^2$ where $\lambda = \rho(H)_{i,i}$ to get the $n + 1$ diagonal entries of $\rho(H^2)$. We need the condition $n \geq 2$, since when $n = 1$ there are only two diagonal entries and $\rho(H^2) = I$. Therefore, $\rho(H^2)$ is linearly independent from I and $\rho(H)$, since λ^2 is linearly independent from the functions $1, \lambda$ (which correspond to I and $\rho(H)$ respectively) when evaluated on $n + 1$ points. This argument also implies that I and $\rho(H)$ are linearly independent when n is at least 1.

We know that the entries of $\rho(EH)$ are given by

$$\rho(EH)_{i+1,i} = \rho(E)_{i+1,i} \cdot \rho(H)_{i,i}.$$

The nonzero elements of $\rho(EH)$ are the nonzero elements of $\rho(E)$ times a factor of λ from the entries of $\rho(H)$. Since we know that λ is linearly independent with 1 when $n \geq 2$, $\rho(EH)$'s entries are not scalar multiples of the entries of $\rho(E)$, and therefore $\rho(EH)$ and $\rho(E)$ are linearly independent. Similarly, $\rho(F)$ and $\rho(FH)$ are linearly independent when $n \geq 2$. We know $\rho(E^2)$ and $\rho(F^2)$ are linearly independent from the rest since they have different support from all the other matrices. Therefore, $I, \rho(E), \rho(F), \rho(H), \rho(E^2), \rho(EH), \rho(F^2), \rho(FH)$, and $\rho(H^2)$ are linearly independent when $n \geq 2$. □

2.2.2. KLV Lower Bounds. Now that we know that $\dim(\mathcal{E}_2) = 9$, we can use this to state the general KLV bound for the distance three $\mathfrak{sl}(2, \mathbb{C})$ case. We know that our Hilbert space has dimension $n + 1$, so our intermediate code $\mathcal{B}$ has dimension at least

$$\dim(\mathcal{B}) \geq \left\lceil \frac{n+1}{9} \right\rceil.$$

Likewise, the final code $\mathcal{C}$ has dimension at least

$$\dim(\mathcal{C}) \geq \frac{1}{10} \left\lceil \frac{n+1}{9} \right\rceil.$$

2.2.3. Optimizing Stage 1. We can choose the optimal intermediate code $\mathcal{B}$ by including every third point on the Lie algebra weight diagram. This gives us an intermediate code of size

$$p = \left\lfloor \frac{n}{3} \right\rfloor + 1.$$

Observe that this is an improvement over the KLV bound of $\lceil (n+1)/9 \rceil$ when $n \geq 3$. Now we want to find what the commutative error space $\mathcal{E}_{\mathcal{B}}$ is after choosing our intermediate subspace $\mathcal{B}$. We know that for any basis vector $b \in \mathcal{B}$, $\rho(E)$, $\rho(F)$, $\rho(E^2)$, $\rho(EH)$, $\rho(F^2)$, and $\rho(FH)$ map b to a basis vector not in $\mathcal{B}$, so for every one of these errors E we have $P_{\mathcal{B}} E P_{\mathcal{B}} = 0$. Thus, we have that:

$$\mathcal{E}_{\mathcal{B}} = \text{span}\{I, \rho(H), \rho(H^2)\}$$

Then going back to our example where $n = 6$, the weight diagram of $\mathcal{E}_{\mathcal{B}}$ looks like the parabola shown in Figure 2.2, where we plot the weight spaces (in this case these are the same as the eigenspaces) of $\rho(H)$ on the x-axis and the weight spaces of $\rho(H^2)$ on the y-axis.

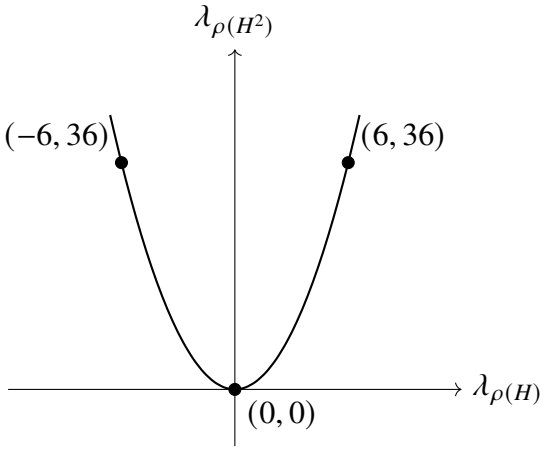


FIGURE 2.2. Plot of the weights of $\rho(H)$ and $\rho(H^2)$ in the example $n = 6$.

We can partition these points using Tverberg's theorem to get

$$\left\lfloor \frac{n-1}{3} \right\rfloor + 1 = \left\lfloor \frac{3-1}{3} \right\rfloor + 1 = 1$$

part. If we use Tverberg's theorem to find our partition in the second stage, we are guaranteed to get

$$\left\lfloor \frac{p-1}{3} \right\rfloor + 1$$

parts (where p is as defined above).

2.2.4. Optimizing Stage 2. Finkel and Shor were able to find partitions with more than $\lfloor (p/3) \rfloor + 1$ parts, thereby creating larger codes. They did this by finding *super-Tverberg* partitions, which we define below.

Definition 2.4 (Super-Tverberg partition)**.** A partition of p points in $\mathbb{R}^d$ with

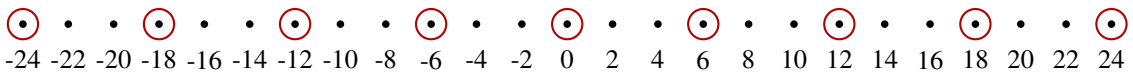
$$r > \left\lfloor \frac{p-1}{d+1} \right\rfloor + 1$$

parts such that the intersection of the convex hulls of the parts is nonempty, is called a *super-Tverberg* partition. A point in the intersection of the convex hulls of a super-Tverberg partition is called a *super-Tverberg point*.

Since Tverberg's theorem is optimal for any set of coincidence-free points, we must have some type of coincidence to find super-Tverberg points. In our case, we can use the fact that our points lie on a parabola as well as the integer lattice to find super-Tverberg points. The next section describes the smallest value of n where we can get a super-Tverberg point.

2.3. Finding a Super-Tverberg Point for $n = 24$

Let $\mathcal{H}$ be a 25-dimensional Hilbert space. Then we can make the following weight diagram for $\rho : \mathfrak{sl}(2, \mathbb{C}) \to \mathbb{C}[x, y]_{24}$ and choose our intermediate subspace $\mathcal{B}$ as indicated by the red circles.



Observe that when n is even, if we rescale the axes when plotting the weight diagram of $\mathcal{E}_{\mathcal{B}}$ so that the $x-$values of the points are consecutive, we will construct the same partition as in the original parabola. When n is odd, we get a weight diagram of the form ...-5, -3, -1, 1, 3, 5, ... This means that we can still plot the points using consecutive integer x-values, but one side of the parabola will have one more point on it than the other. We achieve this by shifting all of the weights by one, and then performing a vertical shear transformation to move the bottom of the parabola to the new 0 weight. We have not yet explored the impact this slight asymmetry will have on forming super-Tverberg partitions, instead focusing on the even case. Figure 2.3 shows a rescaled weight diagram of $\mathcal{E}_{\mathcal{B}}$ when $n = 24$.

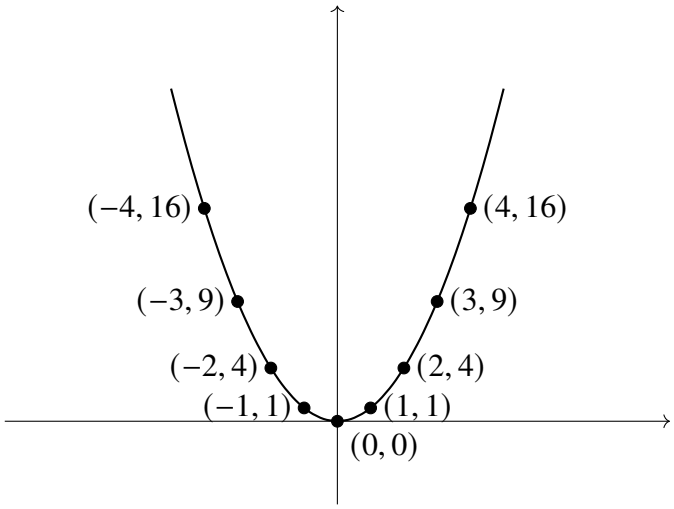


FIGURE 2.3. Plot of the weights rescaled to have consecutive x-values when $n = 24$.

Then we can make the partition shown in Figure 2.4, with the parts indicated by color.

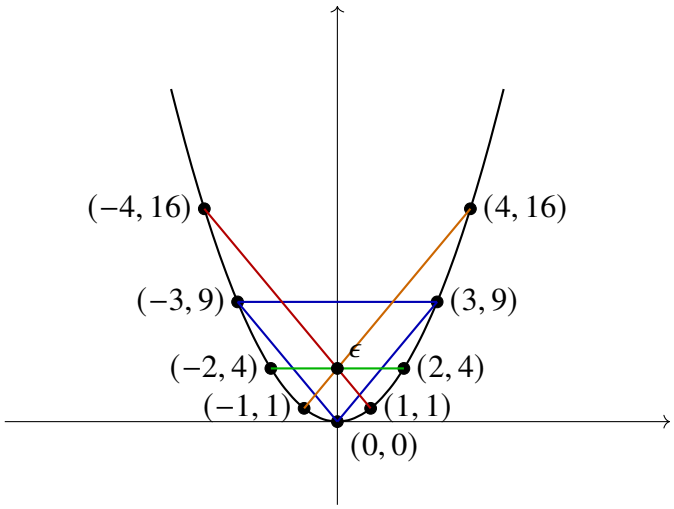


FIGURE 2.4. Super-Tverberg partition of the points in Figure 2.3.

This partition has 4 parts, which is more than $\lfloor (p-1)/3 \rfloor + 1 = \lfloor 8/3 \rfloor + 1 = 3$ parts, so ϵ indicated above is a super-Tverberg point. This results in a final code of size four, where the KLV lower bound is $(1/10)\lceil 25/9 \rceil = 3/10$ parts. We can show that this is the smallest value of n where we get a super-Tverberg point, when partitioning consecutive integer points on a parabola, using a bottom-up approach. We know that if we only have one, two, three, four, or five points on the parabola then there is a unique way to make the Tverberg partition with $\lfloor (p-1)/3 \rfloor + 1$ parts (a point, one segment, one triangle, two segments, and one triangle and one segment respectively). If we have six points, then we could either have three intersecting segments or two triangles. To know when we are able to get more than two intersecting segments, we use the following proposition.

Proposition 2.5. Suppose that two segments that connect points on the parabola $f(x) = x^2$ intersect on the y-axis at a point. Necessarily, the segments connect (a, a^2) to $(-b, b^2)$ and $(-a, a^2)$ to (b, b^2). Then another segment that connects (c, c^2) to $(-d, d^2)$ intersects both of the first two on the y-axis when $ab = cd$.

Thus, we can find numbers with multiple distinct factorizations in order to create many intersecting segments. Note that the proposition only describes when we get many intersecting segments on the $y-$axis. This works in general, since if we look for segments intersecting off the y-axis, we can first transform the parabola by a vertical shear map to make the intersection point on the new y-axis of the parabola. This proposition implies that in order to get three intersecting segments connecting six integer points on the parabola, we would need to find three pairs of integers $(a_i, b_i) \in \{\pm 1, \pm 2, \pm 3\}$ such that $a_1 b_1 = a_2 b_2 = a_3 b_3$. However, there is no way to assign $a_1, b_2, a_2, b_2, a_3, b_3$ to make this true. Therefore, the only Tverberg partition we can make uses two triangles, which is the same as $\lfloor (6-1)/3 \rfloor + 1 = 2$. For seven points on the parabola,

the best Tverberg partition we can make uses one triangle and two segments, which is also not an improvement. Lastly, if we have eight consecutive points on the parabola, then we could have four intersecting segments or two triangles and a segment. However if we have four intersecting segments, that means that we need to find four pairs of integers $(a, b) \in \{\pm 1, \pm 2, \pm 3 \pm 4\}$ such that $a_1 b_1 = a_2 b_2 = a_3 b_3$. Therefore, the first time we observe a super-Tverberg point is when we have nine points on the parabola, which first happens when we have $n = 24$. Next, we discuss the asymptotic bounds on how many extra parts we can get as n gets large.

2.4. Bounds on Improvement for Second Stage

Following this process, the best Tverberg partition we can make is the one that uses the most segments. This means that we want to know for a given number of points p, the maximum number of divisors we can obtain such that the pairs multiply to the same value. We define the following function to quantify this number of divisors.

Definition 2.6. Let

$$\alpha(p) = \max_m \#\{a \mid \exists b, 1 \leq a, b \leq p, ab = m\}.$$

This means that $\alpha(p)$ counts the maximum number of positive divisors less than p such that pairs of these divisors multiply to a common value m.

We first thought that $\alpha(p)$ might be the same function as $\text{maxNu}(p) = \max_{x \in [1..p]} \nu(x)$ (where $\nu(x)$ counts the number of positive distinct divisors of x), however after creating a simple python program to compare these values we found that $\alpha(p)$ grows faster than $\text{maxNu}(p)$. However, we know that $\alpha(p)$ is bounded above by $\text{maxNu}(p^2)$, since in the definition of $\alpha(p)$, $m \leq p^2$ (we can also see this by plotting in python as in Figure 2.5).

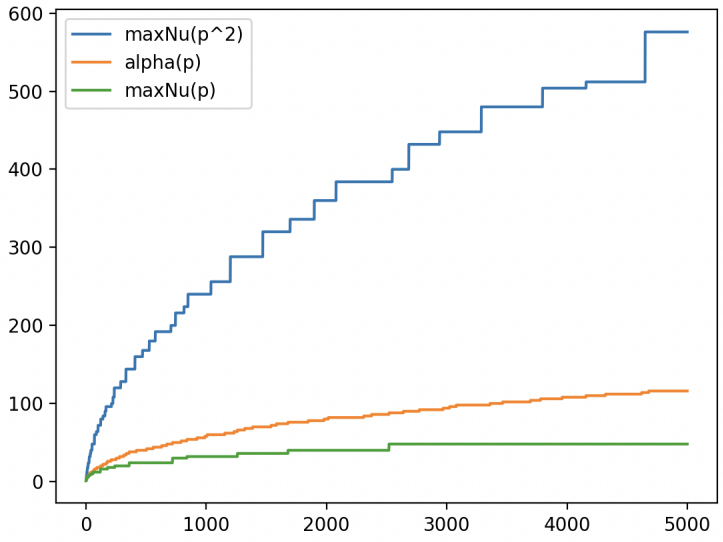


FIGURE 2.5. Plotted values of $\text{maxNu}(p)$, $\alpha(p)$, and $\text{maxNu}(p^2)$ for $p \in [1..5000]$.

Since $\nu(n) = o(n^\epsilon)$ for all $\epsilon > 0$ [**HW09**, Thm. 315], we also get $\nu(n^2) = o(n^{2\epsilon}) = o(n^\epsilon)$ for all $\epsilon > 0$. Therefore $\alpha(p)$ is sublinear, so the growth of the number of extra parts we can get in a super-Tverberg partition is at most sublinear. We have yet to explore whether we are always able to form a Tverberg partition using triangles for the remaining points, but if this is true, then we have found a tight bound for the growth of the number of extra parts, and therefore how much larger we can make our final code. We conjecture that even if we are not able to make a Tverberg partition using only triangles for the remaining points, the growth of the number of extra parts is unbounded.

Conjecture 2.7. For every $s \in \mathbb{N}$, there exists some n large enough such that we can get s extra parts.

We think this is true since as n grows, the number of intersecting segments we can create connecting integer points on the parabola is unbounded. However, we are not sure if we will always be able to partition the remaining points into triangles, or into parts that are nearly all triangles, such that the parts all intersect at the segments' intersection point. Since we are uncertain on this point, it may turn out that the number of extra parts s is upper bounded by some integer M.

2.5. Conclusion

Overall, we have shown that in the $\mathfrak{sl}(2, \mathbb{C})$ case with distance three, we can optimize the first and second stages of the KLV method for at least some values of n. Many questions remain for future research. We don't know if it is always possible to make a Tverberg partition with approximately $\alpha(p)$ segments and $(p - 2\alpha(p))/3$ triangles. We also want to find the exact asymptotic growth of the number of extra parts that we can get using this method, and for which values of n we are guaranteed to get a super-Tverberg point. More broadly, we also want to explore whether we can generalize this method to optimize the second stage in the $\mathfrak{sl}(2, \mathbb{C})$ case for higher distances.

Bibliography

[Bum03] Christopher Bumgardner, *Codes in W*-metric spaces: Theory and examples*, UC Davis, 2003, arXiv:1205.4517.

[Fin22] Bella Finkel, *Quantum error detection in boxes*, UC Davis Pure and Applied Math REU, 2022.

[Gol95] Jonathan Golan, *Foundations of linear algebra*, Springer, 1995.

[Hal15] Brian Hall, *Lie groups, Lie algebras, and representations: An elementary introduction*, Springer, 2015.

[HW09] G. H. Hardy and Edward M. Wright, *An introduction to the theory of numbers*, Oxford University Press, 2009, IBSN: 9780199219865.

[KLV00] Emanuel Knill, Raymond Laflamme, and Lorenza Viola, *Theory of quantum error correction for general noise*, Phys. Rev. Lett. **84** (2000), no. 11, 2525–2528, arXiv:quant-ph/9908066.

[KW12] Greg Kuperberg and Nick Weaver, *A von Neumann algebra approach to quantum metrics*, Mem. Amer. Math. Soc. **215** (2012), no. 1010, 1–80, arXiv:1005.0353.

[Oka23] Rui Okada, *A quantum analog of Delsarte's linear programming bounds*, UC Davis, 2023, arXiv:2502.14165.

[Sho22] Ian Shor, *Quantum error detection and Lie theory*, UC Davis Pure and Applied Math REU, 2022.