Equiangular Tight Frames for Quantum Key Distribution

An Honors Thesis for the Department of Mathematics.

Benjamin Maloy

Tufts University, 2022.

Abstract

Emerging quantum technologies make it possible to isolate single quantum states (for example, spin states of photons). It is possible to generate random binary strings by sharing quantum states; this topic is known as *quantum key distribution* (QKD). Due to quantum states collapsing upon measurement, it is possible to detect eavesdropping between the sharing of these quantum states. This thesis focuses on the linear algebra of some modern QKD schemes. Namely, if Alice and Bob want to generate a random binary string to use as a key in a classical encryption algorithm, they can agree on two sets of *equiangular tight frames* that are related intimately. The primary goal of this thesis is to provide an introduction to QKD with frames, and analyze the properties of some frames that have already been proposed for QKD.

Contents

1 Introduction

The purpose of this thesis is to introduce the field of *Quantum Key Distribution* (QKD), and the relevant linear algebra and frame theory. One distinct advantage that QKD schemes have over classical key sharing schemes (such as Diffie-Hellman key exchange), is that they allow for the detection of eavesdropping. First, I will introduce one of the original QKD schemes, the BB84 algorithm. Next, I will introduce frame theory, which will be important for more complex QKD schemes. Once frames have been introduced, I will focus on *Equiangular Tight Frames* (ETFs), which are the specific kinds of frames used in all of the remaining QKD schemes. Section 2 will then introduce quantum computing and a representation technique for qubits on the Bloch Sphere. Next, I will focus on ETFs with *Companion Frames*, which are the foundation of 2 single qubit QKD schemes (the *trine* and *tetrahedron* schemes). Note that the Hilbert space for 1 qubit is $\mathbb{C}^2$. The rest of the thesis explores how to construct ETF QKD schemes with more than 1 qubit (in higher dimensional Hilbert spaces); one way is to select columns from the *Discrete Fourier Transform Matirx*.

2 Quantum Computing and the Bloch Sphere

For a more detailed introduction to quantum computing refer to the textbook, [9]. Note that we still adopt the convention that a column vector, $x \in \mathbb{C}^d$ can be written:

$$|\mathbf{x}\rangle = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$$

The warning at the beginning of Section 3, where I point out the distinction between our norm now and the norm in Dirac notation, still holds.

In classical computing, information is represented in binary as bits; with ones and zeros. A quantum bit (or qubit) can be any linear combination of 2 basis states. We can represent one qubit as a unit vector in $\mathbb{C}^2$ with the standard basis

$$\{|\mathbf{0}\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, |\mathbf{1}\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}\}$$

Hence, a one qubit state is given by

$$|\psi\rangle = \alpha|\mathbf{0}\rangle + \beta|\mathbf{1}\rangle$$

where $\alpha, \beta \in \mathbb{C}$ and $|\alpha|^2 + \beta|^2 = 1$. Note that α and β cannot both be zero. Furthermore, α and β are each given by 2 real variables, so the state seems to be represented by 4 real variables. Fortunately, we can reduce the possible quantum states to a 2-dimensional space, allowing us to visualize and use our physical intuition.

From quantum theory, we know that each state, $|\psi\rangle$ must be a solution to the Schrodinger equation. Multiples of the same state are considered equivalent solutions to the Schrodinger equation. Hence, states that differ only by multiplication by a complex number are considered equivalent states. Furthermore, normalization is preserved if we multiply by a complex number of unit norm. Therefore, if we

impose the restriction that $|\beta| \neq 0$,

$$\begin{aligned}
|\psi\rangle &= \frac{\overline{\beta}}{|\beta|}|\psi\rangle \\
&= \frac{\overline{\beta}}{|\beta|}\alpha|\mathbf{0}\rangle + \frac{\overline{\beta}}{|\beta|}\beta|\mathbf{1}\rangle \\
&= \frac{\overline{\beta}}{|\beta|}\alpha|\mathbf{0}\rangle + \frac{|\beta|^2}{|\beta|}|\mathbf{1}\rangle \\
&= \frac{\overline{\beta}\alpha}{|\beta|}|\mathbf{0}\rangle + |\beta||\mathbf{1}\rangle
\end{aligned}$$

To write $|\psi\rangle$ in a more suggestive form, consider the exponential representation of the coefficients:

$$\begin{aligned}
\alpha &= |\alpha|e^{i\phi_\alpha} \\
\beta &= |\beta|e^{i\phi_\beta}.
\end{aligned}$$

Hence, $\overline{\beta}\alpha = |\beta||\alpha|e^{i(\phi_\alpha - \phi_\beta)}$. If we let $-\phi = \phi_\alpha - \phi_\beta$, we are left with

$$\begin{aligned}
|\psi\rangle &= \frac{|\beta||\alpha|e^{-i\phi}}{|\beta|}|\mathbf{0}\rangle + |\beta||\mathbf{1}\rangle \\
&= |\alpha|e^{-i\phi}|\mathbf{0}\rangle + |\beta||\mathbf{1}\rangle.
\end{aligned}$$

It is standard to have the $e^{i\phi}$ term as a coefficient to $|\mathbf{1}\rangle$, so we use the fact that a state, $|\psi\rangle$, is equivalent to its product with a unit norm complex number. Hence,

$$\begin{aligned}
|\psi\rangle &= e^{i\phi}|\psi\rangle \\
&= |\alpha||\mathbf{0}\rangle + e^{i\phi}|\beta||\mathbf{1}\rangle.
\end{aligned}$$

We can simplify further by parameterizing the normalization restriction that $|\psi\rangle$

has unit norm. That is, for any $|\psi\rangle$, $|\alpha|^2 + |\beta|^2 = 1$; therefore, we can write

$$|\alpha| = \cos(\theta)$$
$$|\beta| = \sin(\theta).$$

We can now represent $|\psi\rangle$ just in terms of the parameters ϕ, the phase difference between β and α, and a new parameter, θ:

$$|\psi\rangle = \cos(\theta)|\mathbf{0}\rangle + e^{i\phi}\sin(\theta)|\mathbf{1}\rangle$$

The Bloch sphere is a visualization of these states in $\mathbb{R}^3$ using these angles θ and ϕ. Notice that if $\beta = 0$, then $|\psi\rangle = |\mathbf{0}\rangle$, the north pole of the Bloch sphere. The 2 poles of the Bloch sphere are given by $|\mathbf{0}\rangle$ and $|\mathbf{1}\rangle$, so to avoid redundant terms, we substitute θ by $\frac{\theta}{2}$. Therefore, a qubit can be written as a vector on the Bloch sphere as

$$|\psi\rangle = \cos(\frac{\theta}{2})|\mathbf{0}\rangle + e^{i\phi}\sin(\frac{\theta}{2})|\mathbf{1}\rangle$$

where $\theta \in [0, \pi)$ and $\phi \in [0, 2\pi)$.

To further impose physical restraint, we require all transformations on the Bloch sphere to be unitary matrices. Physically, this means that all "quantum gates" that can act on qubits must be unitary transformations. Unitary matrices have the special property that they preserve the norm of vectors they are acting on.

Definition 2.1 (Unitary Matrix)**.** A matrix U is unitary if its inverse is its conjugate transpose. That is, if

$$UU^* = U^*U = I.$$

Unitary matrices are important, because they conserve the square of the inner

product; that is, $||\psi\rangle|^2 = |U|\psi\rangle|^2$, if U is unitary.

3 Introduction to Quantum Key Distribution

3.1 BB84

The BB84 algorithm was first proposed in 1984 by Bennett and Brassard [2]. The algorithm describes how to use polarized photons to share a random binary string, and is an early example of quantum key distribution. The randomly generated key would then be used by some classical encryption algorithm to encrypt an actual message.

Suppose that Alice and Bob want to share a random binary string to use for encryption purposes. They do this by sending a stream of polarized photons and conducting measurements to determine the spin state of the photon. For example, Alice could polarize a photon through a vertical polarization filter, and Bob could conduct a measurement to determine if the photon is spin-up or spin-down along the vertical axis. Hence, the vertical axis is part of the *measurement basis*. We can also measure the spin state on a horizontal axis, as well as 2 different diagonal axes.

The vertical and horizontal axes together form the rectilinear basis (row 1 and 2 of *Table 1*), and the others the diagonal basis (row 3 and 4 of *Table 1*). One noteworthy property of measuring the spin states of photons is that if Bob picks a measurement basis that is different than the basis that Alice chose to polarize the particle, Bob's measurement will result in a random direction along that polarization axis.

An outline of the sharing scheme is given by the following:

1. Bob and Alice agree on which directions mean zero and one for each mea-

surement basis. For this outline, assume:

Basis	Orientation	Binary Value
Rectilinear	$\leftrightarrow$	0
Rectilinear	$\updownarrow$	1
Diagonal	$\nearrow\swarrow$	0
Diagonal	$\nwarrow\searrow$	1

Table 1: Orientation to binary digit conversion

2. Alice chooses a random sequence of photon orientations to send to Bob.

3. For each photon, Bob randomly selects a measurement basis and checks the orientation to obtain a binary digit (bit).

4. Bob announces which measurement basis he used for each photon, and Alice tells him which photons they both used the same basis for.

5. All bits that do not correspond to agreeing measurement bases are discarded.

6. Of the remaining bits, Bob announces a subset so Alice can confirm that no eavesdropping took place (more on this later).

7. If Bob's announced bits correspond to Alice's original photon string, the remaining bits that were not announced publicly are Alice and Bob's shared key.

An example of generating a shared key is shown in Figure 1.

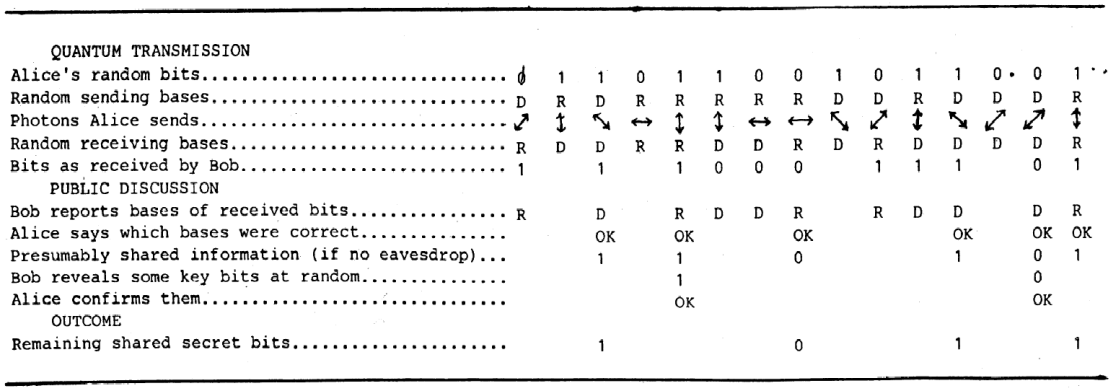

Figure 1: An example key generation given on page 177 of the journal containing [2]

Notice that in step 6, some of the shared key is lost to Bob publicly announcing bits to make sure there is no eavesdropping. If a third person, Eve, was conducting her own spin-state measurements and sending back her own random string of photons, Alice and Bob would disagree on many bits that should correspond. Eve cannot send back the photon she receives, because once a measurement is conducted, the original state cannot be recovered. Hence, this collapse of the state upon measurement allows Bob and Alice to confirm that no one was listening to their message.

3.2 Motivation for Frame Theory

Together, the diagonal and the rectilinear measurement basis form a set of *mutually unbiased bases*. This means that if you conduct a measurement in the wrong basis, you get no information about the original state. The reason these bases have this quality is because the overlap of any vector in one basis has the same angle between each of the vectors in the other basis. It turns out, this is the key feature of quantum key distribution. The BB84 algorithm can be extended to algorithms that use 2 *frames*, which will be introduced in the following chapter, to determine the

orientation of the photons that Alice and Bob share. One key distinction is that now Bob and Alice each have their own distinct measurement frames that allow them to confidently share keys and determine if there is an eavesdropper. Frame based key distribution schemes are the topic of section 5.

4 Frame Theory Background

For a more robust introduction to frames, see the textbook [3], which informed much of this section.

4.1 Introduction to Frames

We first start by defining our Hilbert space as $\mathbb{C}^d$ with the standard inner product:

$$\langle x, y \rangle := y^* x$$

where $x, y \in \mathbb{C}^d$ and * denotes the conjugate transpose. Note that this notation is distinct from the Dirac notation that is common is physics (where $\langle x|y\rangle := x^* y$).

Definition 4.1 (Frame)**.** A set $F = \{f_j\}_{j=1}^N \subseteq \mathbb{C}^d$ is a *frame* if there exist finite $A, B > 0$ such that for all $x \in \mathbb{C}^d$,

$$A||x||^2 \leq \sum_j |\langle x, f_j \rangle|^2 \leq B||x||^2$$

A frame, F, is called *tight* if $A = B$; that is, if

$$\sum_j |\langle x, f_j \rangle|^2 = A||x||^2$$

for all $x \in \mathbb{C}^d$. A frame, F, is called *equiangular* if

$$|\langle f_j, f_k \rangle|^2 = c$$

when $j \neq k$.

Definition 4.2 (Spanning Set)**.** A set $F = \{f_j\}_{j=1}^N \subseteq \mathbb{C}^d$ is a *spanning set* if for all $x \in \mathbb{C}^d$, there exist some set of coefficients, $\{c_j \in \mathbb{C}\}$ such that

$$x = \sum_{j=1}^{N} c_j f_j$$

Theorem 4.1. *A frame is a spanning set.*

Proof. Suppose that the set $F = \{f_j\}_{j=1}^N \subseteq \mathbb{C}^d$ is a frame. Let $P = Span(F)$. If F is not a spanning set of $\mathbb{C}^d$, then P is a proper subset of $\mathbb{C}^d$. This means that the orthogonal complement of P, $P^\perp$, is not just the zero vector. So let $y \in (P^\perp - \{\mathbf{0}\})$ (that is, y is nonzero). But $P^\perp \subset \mathbb{C}^d$, by the definition of orthogonal complement, so $y \in \mathbb{C}^d$. y being in $P^\perp$ means that it is orthogonal to every vector in P, including the vectors in F that span P. Hence,

$$\sum_j |\langle y, f_j \rangle|^2 = 0.$$

Then, because F is a frame, the definition of frame implies the existence of finite $A > 0$ such that,

$$A||y||^2 \leq \sum_j |\langle y, f_j \rangle|^2 = 0.$$

But y is nonzero, which contradicts the frame definition. Therefore, there is no such $y \in (P^\perp - \{\mathbf{0}\})$, and $P^\perp$ only contains the zero vector. So there is no vector in $\mathbb{C}^d$ that is perpendicular to every vector in P, the span of F. This means that F is a

spanning set of $\mathbb{C}^d$. $\square$

It will be useful to define a matrix, D, whose columns are the entries in a set of vectors, $F = \{f_j\}_{j=1}^N \subseteq \mathbb{C}^d$; that is, let

$$D = [f_1 \ f_2 \ \dots \ f_N].$$

Definition 4.3 (Gramian)**.** The *Gramian* of a set of vectors $F = \{f_j\}_{j=1}^N \subseteq \mathbb{C}^d$, G, is a matrix whose entries are given by

$$G = D^*D$$

Definition 4.4 (Frame Operator)**.** The *Frame Operator* is given by

$$S = DD^*$$

Notice that the entries of the Gramian are given by

$$G_{j,k} = \langle f_j, f_k \rangle.$$

Also note that the Gramian is $N \times N$.

The frame operator is $d \times d$. Note that $Sx = \sum_{j=1}^N \langle x, f_j \rangle f_j$. We can see this from the case where $N = d = 2$. Let

$$f_1 = \begin{bmatrix} f_{1,0} \\ f_{1,1} \end{bmatrix}, f_2 = \begin{bmatrix} f_{2,0} \\ f_{2,1} \end{bmatrix}$$

Then,

$$\begin{aligned}
Sx &= DD^*x \\
&= D\left(\begin{bmatrix} \overline{f_{1,0}} & \overline{f_{1,1}} \\ \overline{f_{2,0}} & \overline{f_{2,1}} \end{bmatrix} \begin{bmatrix} x_1 x_2 \end{bmatrix}\right) \\
&= D \begin{bmatrix} \overline{f_{1,0}}x_1 + \overline{f_{1,1}}x_2 \\ \overline{f_{2,0}}x_1 + \overline{f_{2,1}}x_2 \end{bmatrix} \\
&= D \begin{bmatrix} \langle x, f_1 \rangle \\ \langle x, f_2 \rangle \end{bmatrix} \\
&= \begin{bmatrix} f_{1,0} & f_{2,0} \\ f_{1,1} & f_{2,1} \end{bmatrix} \begin{bmatrix} \langle x, f_1 \rangle \\ \langle x, f_2 \rangle \end{bmatrix} \\
&= \begin{bmatrix} \langle x, f_1 \rangle f_{1,0} + \langle x, f_2 \rangle f_{2,0} \\ \langle x, f_1 \rangle f_{1,1} + \langle x, f_2 \rangle f_{2,1} \end{bmatrix} \\
&= \begin{bmatrix} \langle x, f_1 \rangle f_{1,0} \\ \langle x, f_1 \rangle f_{1,1} \end{bmatrix} + \begin{bmatrix} \langle x, f_2 \rangle f_{2,0} \\ \langle x, f_2 \rangle f_{2,1} \end{bmatrix} \\
&= \langle x, f_1 \rangle f_1 + \langle x, f_2 \rangle f_2 \\
&= \sum_{j=1}^{N} \langle x, f_j \rangle f_j
\end{aligned}$$

Hopefully, it is clear how this can be extended to arbitrary N and d.

Corollary 4.2. *The frame operator for any frame $F = \{f_j\}_{j=1}^N \subseteq \mathbb{C}^d$, S, is invertible.*

Proof. Suppose that S is not invertible and that the null space contains more than just the zero vector. Then, for some $x \in (\mathbb{C}^d - \{\mathbf{0}\})$,

$$Sx = 0.$$

However, S is $d \times d$, which means that $Sx \in \mathbb{C}^d$. Hence, $\langle Sx, x\rangle = \sum_{j=1}^{N} |\langle x, f_j\rangle|^2 = 0$, which violates the lower bound frame condition. Therefore, there is no such $x \in (\mathbb{C}^d - \{\mathbf{0}\})$, and so the kernel of S is trivial. Hence, S, is invertible. □

Corollary 4.3. *The Gramian for the frame $F = \{f_j\}_{j=1}^{N} \subseteq \mathbb{C}^d$, G, is not invertible.*

Proof. We noted before that $G = D^*D$ where D is $d \times N$. $N > d$, so D must have linearly dependent columns. Therefore, D is not one-to-one and must have a nonzero vector in its kernel. That is, there is some $x \in (\mathbb{C}^N - \mathbf{0})$ such that

$$Dx = \mathbf{0}.$$

Hence,

$$\begin{aligned} Gx &= D^*Dx \\ &= D^*\mathbf{0} \\ &= \mathbf{0} \end{aligned}$$

for a nonzero x, so G also has a nontrivial kernel. Therefore, G is not invertible. □

Lemma 4.4. *The Gramian and frame operator for the frame $F = \{f_j\}_{j=1}^{N} \subseteq \mathbb{C}^d$ have the same trace.*

Proof. The trace of the product of 2 matrices does not depend on the order of multiplication. So $Tr(AB) = Tr(BA)$, and by setting $A = D^*$ and $B = D$, we see that $Tr(G) = Tr(S)$. □

Lemma 4.5. *In fact, the Gramian and frame operator for the frame $F = \{f_j\}_{j=1}^{N} \subseteq \mathbb{C}^d$ have the same eigenvalues, except for 0.*

This is true for any pair AB and BA, so long as both products are square. The larger of the 2 has the same eigenvalues as the smaller one, except with additional zero eigenvalues. That is, if AB is $N \times N$ and BA is $d \times d$ where $N > d$, $N - d$ of AB's eigenvalues are 0.

Definition 4.5 (Frame Potential)**.** For a given frame, $F = \{f_j\}_{j=1}^N \subseteq \mathbb{C}^d$, the frame potential, P_F, is given by

$$P_F = \sum_{j,k=1}^{N} |\langle f_j, f_k \rangle|^2.$$

The following lemma will be useful in proving Theorem 4.7.

Lemma 4.6. *The trace of the Gramian squared is equal to the frame potential; that is,*

$$Tr(G^2) = \sum_{j,k=1}^{N} |\langle f_j, f_k \rangle|^2$$

Proof. By the definition of trace,

$$Tr(G^2) = \sum_{j=1}^{N} \langle G^2 e_j, e_j \rangle$$

where e_j is the standard basis for $\mathbb{C}^d$. First let's write out what $G^2 e_j$ is:

$$\begin{aligned} G^2 e_j &= G G e_j \\ &= G \begin{bmatrix} G_{1j} \\ G_{2j} \\ \vdots \\ G_{Nj} \end{bmatrix} \\ &= \begin{bmatrix} \sum_{k=1}^N G_{1k} G_{kj} \\ \sum_{k=1}^N G_{2k} G_{kj} \\ \vdots \\ \sum_{k=1}^N G_{Nk} G_{kj} \end{bmatrix}. \end{aligned}$$

Then $\langle G^2 e_j, e_j \rangle = e_j^* G^2 e_j$ will select the jth-row from this column vector. Hence,

$$\begin{aligned} \langle G^2 e_j, e_j \rangle &= \sum_{k=1}^N G_{jk} G_{kj} \\ &= \sum_{k=1}^N \langle f_j, f_k \rangle \langle f_k, f_j \rangle \\ &= \sum_{k=1}^N \langle f_j, f_k \rangle \overline{\langle f_j, f_k \rangle} \\ &= \sum_{k=1}^N |\langle f_j, f_k \rangle|^2. \end{aligned}$$

Therefore,

$$\begin{aligned} Tr(G^2) &= \sum_{j=1}^N \langle G^2 e_j, e_j \rangle \\ &= \sum_{j=1}^N \sum_{k=1}^N |\langle f_j, f_k \rangle|^2. \end{aligned}$$

□

Now that we have a solid understanding of frames, I will introduce a useful set of criteria for determining if a frame is tight. Then we will begin exploring the use of frames in QKD.

4.2 Tight Frame Criteria

In this section I present a set of 4 equivalent conditions for a frame being tight. This allows us more ways to prove that a frame is tight, which will be useful when we are looking for frames that can be used for QKD schemes.

Theorem 4.7. *Suppose that $F = \{f_j\}_{j=1}^N \subseteq \mathbb{C}^d$ is a set of unit norm vectors in $\mathbb{C}^d$. The following are equivalent:*

1. $x = \frac{d}{N} \sum_{k=1}^N \langle x, f_k \rangle f_k$, *for all* $x \in \mathbb{C}^d$
2. $||x||^2 = \frac{d}{N} \sum_{k=1}^N |\langle x, f_k \rangle|^2$, *for all* $x \in \mathbb{C}^d$
3. $\sum_{j,k=1}^N |\langle f_j, f_k \rangle|^2 = \frac{N^2}{d}$
4. F *is a tight frame.*

Proof. First note that condition 2 is the definition of a tight frame from Section 4.1, where the tight bound is $\frac{N}{d}$ (that is, $A = B = \frac{N}{d}$). Therefore, if we show that the other 3 are equivalent, we have shown that all 4 conditions prove that F is a tight frame. We will show that $1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 1$.

- $1 \Rightarrow 2$:

$$\begin{aligned}
||x||^2 &= \langle x, x\rangle && \text{standard definition of norm in } \mathbb{C}^d \\
&= \Big\langle x, \frac{d}{N}\sum_{k=1}^{N}\langle x, f_k\rangle f_k\Big\rangle && \text{statement 1 above} \\
&= \frac{d}{N}\sum_{k=1}^{N}\big\langle x, \langle x, f_k\rangle f_k\big\rangle && \text{linearity of inner product} \\
&= \frac{d}{N}\sum_{k=1}^{N}\overline{\langle x, f_k\rangle} f_k^* x && \text{definition of inner product} \\
&= \frac{d}{N}\sum_{k=1}^{N}\overline{\langle x, f_k\rangle}\langle x, f_k\rangle \\
&= \frac{d}{N}\sum_{k=1}^{N}|\langle x, f_k\rangle|^2 && \text{definition of complex norm}
\end{aligned}$$

- $2 \Rightarrow 3$: We will plug some $f_j \in F$ in for x in statement 2:

$$||f_j||^2 = \frac{d}{N}\sum_{k=1}^{N}|\langle f_j, f_k\rangle|^2.$$

But f_j is a unit vector, hence

$$\sum_{k=1}^{N}|\langle f_j, f_k\rangle|^2 = \frac{N}{d}.$$

So if we sum over all $f_j \in F$ (of which there are N), we find

$$\sum_{j,k=1}^{N}|\langle f_j, f_k\rangle|^2 = \frac{N^2}{d}.$$

- $3 \Rightarrow 1$: Let $D = \begin{bmatrix} f_1 & f_2 & \dots & f_N \end{bmatrix}$, a dxN matrix. Then we have $G = D^*D$,

the Gramian, and let $S = DD^*$, a dxd matrix where for $x \in \mathbb{C}^d$,

$$Sx = \sum_{k=1}^{N} \langle x, f_k \rangle f_k.$$

From Lemma 4.4, we know that $Tr(G) = Tr(S)$. Furthermore, because each f_k has unit norm,

$$Tr(G) = \sum_{k=1}^{N} \langle f_k, f_k \rangle = N.$$

Then, if we write the trace of S as the sum of its eigenvalues, $\{\lambda_1, \lambda_2, \ldots, \lambda_d\}$, and equate it to the trace of G, we find,

$$Tr(S) = \sum_{m=1}^{d} \lambda_m = N.$$

We will finish the proof by applying the Cauchy-Schwarz inequality with the vectors $\theta, \phi \in \mathbb{R}^d$

$$\theta = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_d \end{bmatrix}, \phi = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}.$$

So we have that

$$\langle \phi, \phi \rangle = d,$$

$$\langle \theta, \phi \rangle = \sum_{m=1}^{d} \lambda_m = N,$$

and

$$\begin{aligned}
\langle \theta, \theta \rangle &= \sum_{m=1}^{d} \lambda_m^2 \\
&= Tr(S^2) && \text{the sum of the squared eigenvalues of } S \\
&= Tr(G^2) && \text{might need separate lemma for this} \\
&= \sum_{j,k=1}^{N} |\langle f_j, f_k \rangle|^2 && \text{lemma 1.1.} \\
&= \frac{N^2}{d} && \text{statement 3}
\end{aligned}$$

So because the equality condition of the *Cauchy-Schwarz Inequality*,

$$|\langle \theta, \phi \rangle|^2 \leq \langle \theta, \theta \rangle \langle \phi, \phi \rangle,$$

only holds if θ is a linear combination of ϕ. By plugging in the above values into the inequality we do see that

$$|\langle \theta, \phi \rangle|^2 = \langle \theta, \theta \rangle \langle \phi, \phi \rangle,$$

so θ must be a linear combination of ϕ. Then because $\langle \theta, \theta \rangle = \frac{N^2}{d}$, each entry of θ and each eigenvalue of S is $\frac{N}{d}$. Hence, $S = \frac{N}{d} I$. Therefore,

$$Sx = \frac{N}{d} x = \sum_{k=1}^{N} \langle x, f_k \rangle f_k.$$

Equivalently,

$$x = \frac{d}{N} \sum_{k=1}^{N} \langle x, f_k \rangle f_k.$$

□

5 Companion Frames for Key Distribution

For the remainder of the thesis, we will be concerned with unit norm *Equiangular Tight Frames* (ETFs). That is, $F = \{f_j\}_{j=1}^N \subseteq \mathbb{C}^d$ is an ETF if it satisfies the definition of a frame and is *tight* and *equiangular*, as outlined by Definition 4.1. The unit norm and equiangular conditions amount to the equation:

$$|\langle f_j, f_k \rangle|^2 = \begin{cases} 1, & \text{if } j = k \\ c, & \text{if } j \neq k. \end{cases}$$

With an ETF, we can understand some bounds on the angle between any 2 vectors in the frame. Specifically, following [6], we introduce the *maximal frame correlation*:

Definition 5.1. Let $F = \{f_j\}_{j=1}^N \subseteq \mathbb{C}^d$ be a unit norm frame. The *maximal frame correlation*, $\mathscr{M}$, for this frame is given by

$$\mathscr{M}(F) = \max\{|\langle f_j, f_k, \rangle|\}_{j,k=1}^N$$

An interesting theorem is given in [6, pg. 4], and tells us that

$$\mathscr{M}(F) \geq \sqrt{\frac{N-d}{d(N-1)}},$$

where equality holds when F is an ETF and $N \leq d^2$. In the following examples, the ETFs we work with will have $N \leq d^2$.

The rectilinear and diagonal bases used in the BB84 algorithm are mutually unbiased. This means that if Bob uses the wrong measurement basis, he gets no information about what the signal was. Therefore, Alice announces which of Bob's

measurement basis choices are correct, and then those measured bits are used to generate a key. To ensure that someone is not intercepting Alice's signal and re-sending their own (a so-called intercept-resend attack), Alice and Bob will publicly compare a few bits for which they used the same measurement basis. If many bits do not agree, it means that someone is intercepting Alice's bits and sending their own random photon. This process of sharing some bits is called *sifting*. The interceptor cannot send Alice's signal with certainty, because once they conduct a measurement, the original signal is lost (this is the collapse of the wave function in action). While this allows for the detection of eavesdropping, the publicly announced bits cannot be used to generate a key. The sifting rate for ETF key distribution schemes is discussed in [4] and [1], and is shown to be smaller for the ETF key distribution schemes than for complete basis schemes (like BB84).

The set of Bob's signals are called a *companion frame* to Alice's frame.

Definition 5.2 (Companion Frame)**.** For an ETF $F = \{f_j\}_{j=1}^N \subseteq \mathbb{C}^d$, another ETF, $G = \{g_j\}_{j=1}^N \subseteq \mathbb{C}^d$, is a *companion ETF* for F if

$$|\langle g_j, f_k\rangle|^2 = \begin{cases} 0, & \text{if } j = k \\ c, & \text{if } j \neq k \end{cases}$$

For the remainder of this section, let F be the set of signals that Alice can send, and G be Bob's companion frame. We are interested in finding a map from F to G. While it is apparently possible to use non-unitary gates in quantum circuits (see [7]), here we want to find a unitary matrix such that

$$|\langle U f_j, f_k\rangle|^2 = \begin{cases} 0, & \text{if } j = k \\ c, & \text{if } j \neq k. \end{cases}$$

This matrix, U, will act on Alice's signals and yield G.

For the ETF scheme, Alice will send some signal in her frame, f_j. Let the size of Alice's frame be N. Bob will then select some vector in the companion frame to Alice's frame, g_k, and conduct a measurement from a set of Positive Operator-Valued Measures (POVM).

For a precise definition of POVMs, see [8]. For our purposes, it is enough to know that a POVM is a set of transformations that correspond to measurements on qubits.

The POVM that Bob will use is

$$\{G_j = \frac{d}{N} g_j g_j^*\}$$

Then if G is a companion frame for F, Bob will measure

$$\begin{aligned} G_j f_k &= \frac{d}{N} g_j g_j^* f_k \\ &= \frac{d}{N} g_j \langle f_k, g_j \rangle \\ &= \frac{d}{N} g_j \overline{\langle g_j, f_k \rangle}. \end{aligned}$$

So Alice's signal is projected onto Bob's. Specifically,

$$G_j f_k = \begin{cases} |\mathbf{0}\rangle, \text{ if } j = k \\ \frac{d\overline{c}}{N} g_j, \text{ if } j \neq k. \end{cases}$$

The important quality here is that if Bob does not get the zero state, then he does not know which f_k with $k \neq j$ that Alice sent. Hence, if Alice sends f_k and Bob measures with G_j, Bob announces $N-2$ possible indices in $B = \{1, 2, \ldots, N\} - \{j\}$.

If $k \in B$, then Alice announces failure (essentially, Bob guessed that the signal was not f_k, but it was). Otherwise, Alice announces success and they know that the signal was f_k. Any eavesdropper on the public channel, though, only knows that the signal is either f_j or f_k. Alice and Bob will have agreed on some kind of mapping from indices to bits beforehand, that will be used to generate the random binary string to be used as a key.

5.1 Trine

One of many papers by Joseph Renes on quantum key distribution, [4], describes 2 ways to find a frame with a companion on the Bloch sphere (that is, for single qubits). The first has the desired quality of requiring just one unitary matrix to transform between Alice and Bob's frames, and is referred to as the trine scheme. [1]

Let Alice's frame, F, be given by the columns of the following matrix:

$$\begin{bmatrix} f_1 & f_2 & f_3 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & 1 \\ 1 & \omega & \omega^2 \end{bmatrix},$$

where $\omega = e^{\frac{2\pi i}{3}}$.

You can show that $F = \{f_j\}_{j=1}^3$ is a tight frame by picking one of the equivalent conditions in Theorem 4.7, and that F is equiangular by finding $|\langle f_j, f_k \rangle|^2 = c$ where c is constant for $j \neq k$. It turns out that

$$\sum_{j,k=1}^{N} |\langle f_j, f_k \rangle|^2 = 3 + \sum_{j \neq k} c$$

[1] trine means a group of three; in astrology, it refers to when the relative position of 2 planets is 120°.

where $c = \frac{1}{4}$, and is found in the code in Appendix A. Hence,

$$\begin{aligned}\sum_{j,k=1}^{N} |\langle f_k, f_k \rangle|^2 &= 3 + 6 * \frac{1}{4} \\ &= \frac{9}{2} \\ &= \frac{N^2}{d}.\end{aligned}$$

So criteria number 3 in Theorem 4.7 is satisfied, proving that F is an ETF.

The companion frame is constructed by rotating each f_j by 180°. Hence,

$$U = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

We can show that U is unitary by showing that its complex conjugate, U^*, is also its inverse:

$$UU^* = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I$$

$$U^*U = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I$$

Then G is given by the columns of the following matrix:

$$\begin{bmatrix} g_1 & g_2 & g_3 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & 1 \\ -1 & -\omega & -\omega^2 \end{bmatrix},$$

The code in Appendix A shows that $G = \{g_j\}_{j=1}^{3}$ is a companion with

$$|\langle g_k, f_j \rangle|^2 = \begin{cases} 0, & \text{if } j = k \\ \frac{3}{4}, & \text{if } j \neq k. \end{cases}$$

Hence, G is a companion frame to F, and this scheme can be used by Alice and Bob to generate a random key.

G is shown with the dotted lines and F with the solid lines on the Bloch Sphere in figure 2.

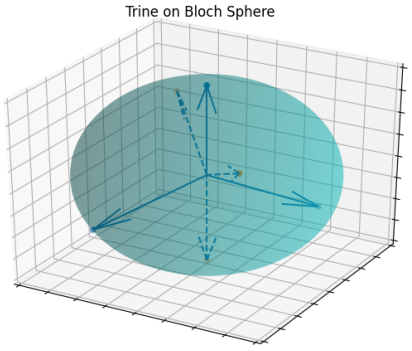

Figure 2: Alice and Bob's frames for the trine scheme

There is a special connection between U and a specific matrix, the *Discrete Fourier Transform Matrix*, which will be discussed further in section 6.

5.2 Tetrahedron

Renes introduces another Bloch sphere quantum key distribution scheme in [4] that uses a frame that includes 4 vectors. One key discovery we made is that there is no single unitary matrix to map from Alice's to Bob's set of signals. This fact is stated and a loose proof is outlined in [1]. We present a more detailed proof below, after

setting up the tetrahedron frame.

Here we want to find an ETF with 4 elements in $\mathbb{C}^2$. Because we are looking for a tight frame, we can use Theorem 4.7 to search for such a frame; namely, $\sum_{j,k=1}^{N} |\langle f_j, f_k \rangle|^2 = \frac{16}{2} = 8$. Expanding out further:

$$\begin{aligned} 8 &= \sum_{j=1}^{4} |\langle f_j, f_j \rangle|^2 + \sum_{j \neq k} |\langle f_j, f_k \rangle|^2 \\ &= 4 + \sum_{m=1}^{12} c \\ &= 4 + 12c \end{aligned}$$

So $c = |\langle f_j, f_k \rangle|^2 = \frac{1}{3}$. Now that we know the overlap of each vector in F, we can construct the general F by applying 2 of the Pauli matrices,

$$X = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, T = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

to some arbitrary 2 dimensional vector. The only restriction on this vector is that it has unit norm; hence, we can write our vector with just 2 degrees of freedom (r and θ):

$$f_0 = \begin{bmatrix} r \\ \sqrt{1-r^2} e^{i\theta} \end{bmatrix}.$$

For simplification, let $R = \sqrt{1-r^2}$. Then we can try applying each of the 2 Pauli

matrices above, and then combinations of the 2:

$$\begin{aligned} f_1 &= Tf_0 \\ &= \begin{bmatrix} Re^{i\theta} \\ r \end{bmatrix} \\ f_2 &= Xf_0 \\ &= \begin{bmatrix} r \\ -Re^{i\theta} \end{bmatrix} \\ f_3 &= TXf_0 \\ &= \begin{bmatrix} -Re^{i\theta} \\ r \end{bmatrix} \end{aligned}$$

So Alice's ETF is given by the columns of the following matrix:

$$\begin{bmatrix} f_1 & f_2 & f_3 & f_4 \end{bmatrix} = \begin{bmatrix} r & Re^{i\theta} & r & -Re^{i\theta} \\ Re^{i\theta} & r & -Re^{i\theta} & r \end{bmatrix}.$$

Then by imposing the equiangular condition ($|f_j, f_k|^2 = \frac{1}{3}$) we find that

$$r = \sqrt{\frac{1}{2} + \sqrt{\frac{1}{12}}}, \theta = \frac{\pi}{4}.$$

F is shown on the Bloch Sphere in Figure 3.

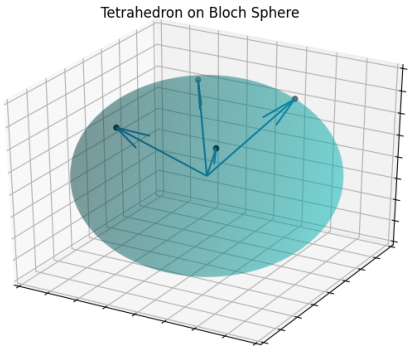

Figure 3: Alice's frame for the tetrahedron scheme

Now comes the matter of finding a single unitary matrix to yield a companion frame. The first condition to check (which we will find fails), is that $|\langle Uf_j, f_j\rangle|^2 = 0$. We will slightly simplify and check the equivalent condition that $\langle Uf_j, f_j\rangle = 0$. Generally, we can write

$$U = \begin{bmatrix} a & b \\ c & d \end{bmatrix},$$

and then apply U to each f_j:

$$Uf_0 = \begin{bmatrix} ar + bRe^{i\theta} \\ cr + dRe^{i\theta} \end{bmatrix}$$

$$Uf_1 = \begin{bmatrix} aRe^{i\theta} + br \\ cRe^{i\theta} + dr \end{bmatrix}$$

$$Uf_2 = \begin{bmatrix} ar - bRe^{i\theta} \\ cr - dRe^{i\theta} \end{bmatrix}$$

$$Uf_3 = \begin{bmatrix} -aRe^{i\theta} + br \\ -cRe^{i\theta} + dr \end{bmatrix}$$

Next, take the inner product of each Uf_j with the corresponding f_j:

$$\begin{aligned}
\langle Uf_0, f_0\rangle &= \begin{bmatrix} r & Re^{-i\theta} \end{bmatrix} \begin{bmatrix} ar + bRe^{i\theta} \\ cr + dRe^{i\theta} \end{bmatrix} \\
&= ar^2 + brRe^{i\theta} + crRe^{-i\theta} + dR^2 \\
\langle Uf_1, f_1\rangle &= \begin{bmatrix} Re^{-i\theta} & r \end{bmatrix} \begin{bmatrix} aRe^{i\theta} + br \\ cRe^{i\theta} + dr \end{bmatrix} \\
&= aR^2 + brRe^{-i\theta} + crRe^{i\theta} + dr^2 \\
\langle Uf_2, f_2\rangle &= \begin{bmatrix} r & -Re^{-i\theta} \end{bmatrix} \begin{bmatrix} ar - bRe^{i\theta} \\ cr - dRe^{i\theta} \end{bmatrix} \\
&= ar^2 - brRe^{i\theta} - crRe^{-i\theta} + dR^2 \\
\langle Uf_3, f_3\rangle &= \begin{bmatrix} -Re^{-i\theta} & r \end{bmatrix} \begin{bmatrix} -aRe^{i\theta} + br \\ -cRe^{i\theta} + dr \end{bmatrix} \\
&= aR^2 - brRe^{-i\theta} - crRe^{i\theta} + dr^2
\end{aligned}$$

Recall from the definition of companion frame that we also require $|\langle Uf_j, f_k\rangle|^2 = c$ when $j \neq k$, where c is some constant in order for $G = UF$ to be a companion frame for F. We would have to check that next if we found that $|\langle Uf_j, f_j\rangle|^2 = 0$. Now we can set up a matrix equation, $Ax = b$, where b is the zero vector,

$$A = \begin{bmatrix} r^2 & rRe^{i\theta} & rRe^{-i\theta} & R^2 \\ R^2 & rRe^{-i\theta} & rRe^{i\theta} & r^2 \\ r^2 & -rRe^{i\theta} & -rRe^{-i\theta} & R^2 \\ R^2 & -rRe^{-i\theta} & -rRe^{i\theta} & r^2 \end{bmatrix}, x = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}.$$

So we can only find a non-trivial x if the determinant of A is zero. Alas, the determinant of A is $0.77i$. The code to check this determinant is in Appendix B. Therefore, the only such matrix that satisfies

$$|\langle Uf_j, f_j\rangle|^2 = 0$$

is the matrix of all 0s. This is not a unitary matrix, which proves the following theorem:

Theorem 5.1. *For the tetrahedron ETF, F, there is no unitary U such that $|\langle Uf_j, f_j\rangle|^2 = 0$ where $j = 1,2,3,4$.*

Then the following corollary of theorem 5.1 comes from the definition of a companion frame:

Corollary 5.2. *For the tetrahedron ETF, F, there is no unitary U such that $G = UF = \{Uf_j\}_{j=1}^N \subseteq \mathbb{C}^d$ is a companion ETF for F.*

Note that it may be possible to construct G from a unitary matrix acting on a permutation of the columns of F; this is discussed in [1].

5.3 Extend to Higher Dimension

We are interested in whether or not this trend of not being able to find a single unitary matrix extends to higher dimensions. Following [5], we can generate ETFs for $d = 3, N = 9$. Similar with the tetrahedron, we will find some f_0, and then

generate a frame by multiplying by the matrices

$$X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \omega & 0 \\ 0 & 0 & \omega^2 \end{bmatrix}, \; T = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

A family of suitable f_0 is given by [5], and a subset of these are from the set

$$\{f_0 = \begin{bmatrix} r_0 & r_+ e^{i\theta_1} & r_- e^{i\theta_2} \end{bmatrix}^T : \theta_1, \theta_2 \in \{\frac{\pi}{3}, \pi, \frac{5\pi}{3}\}, \frac{1}{\sqrt{2}} < r_0 \leq \sqrt{\frac{2}{3}}\}$$

where

$$r_\pm = \frac{r_0}{2} \pm \frac{1}{2}\sqrt{2 - 3r_0^2}$$

We then find that

$$\{X^m T^n f_0\}_{m,n=0}^3$$

does form an equiangular frame, from the code in Appendix C. Specifically,

$$|\langle f_j, f_k \rangle|^2 = \begin{cases} 1, & \text{if } j = k \\ \frac{1}{4}, & \text{if } j \neq k. \end{cases}$$

Furthermore, the matrix equation $Ax = b$ where b is the zero vector, x contains the entries of an arbitrary 3×3 matrix, and A is the coefficients found by solving $\langle U f_j, f_j \rangle = 0$, does seem to have nontrivial solutions. This is demonstrated by the code in Appendix C. It means that it is at least possible for there to be some unitary matrix that maps $\{X^m T^n f_0\}_{m,n=0}^3$ directly to a companion frame.

6 DFT Eigenvectors and Companion Frame Conjecture

6.1 The Discrete Fourier Transform Matrix

The Discrete Fourier Transform (DFT) is a mapping from a signal in one state space to frequency space. That is, the signal could be a function of time, and specific points of the signal would be mapped to a function of the frequency. This type of transform is important to signal processing. We, however, are only concerned with the mathematical properties of the DFT matrix.

Definition 6.1. Let ω be a primitive root of unity ($\omega = e^{\frac{2\pi i}{N}}$). The $N \times N$ DFT matrix is given by

$$W = \left[\frac{\omega^{rc}}{\sqrt{N}}\right]_{r,c=0}^{(N-1)}$$

(Note that $\omega^j \equiv \omega^{j \bmod N}$.)

For example, the 5×5 DFT matrix is given by

$$W = \frac{1}{\sqrt{5}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \omega^4 \\ 1 & \omega^2 & \omega^4 & \omega & \omega^3 \\ 1 & \omega^3 & \omega & \omega^4 & \omega^2 \\ 1 & \omega^4 & \omega^3 & \omega^2 & \omega \end{bmatrix}$$

The DFT matrix is very interesting for our study of equiangular tight frames (ETFs) and companion ETFs. First of all, we can construct an ETF, $F = \{f_j\}_{j=1}^N \subseteq \mathbb{C}^d$, where each f_j is the j^{th} column of W with the final row removed (and then scaling to unit length). For example, for we can find an ETF $F = \{f_j\}_{j=1}^5 \subseteq \mathbb{C}^4$,

where each f_j is given by the columns of

$$F = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \omega^4 \\ 1 & \omega^2 & \omega^4 & \omega & \omega^3 \\ 1 & \omega^3 & \omega & \omega^4 & \omega^2 \end{bmatrix}.$$

Furthermore, it is possible when $d+1$ is prime to use a special eigenvector of the DFT matrix, u, as the diagonal of a unitary matrix to find a companion for F. u is given in [1] and shown below:

$$u = \begin{bmatrix} 0 & (\frac{1}{d+1}) & (\frac{2}{d+1}) & \dots & (\frac{d}{d+1}) \end{bmatrix}^T$$

where $(\frac{a}{d+1})$ denotes the Legendre symbol. That is,

$$(\frac{a}{d+1}) = \begin{cases} 1, \text{ if } n \text{ is a quadratic residue mod } d+1 \\ -1, \text{ if } n \text{ is not a quadratic residue mod } d+1 \end{cases}$$

(note that $a \in [1, d+1)$ for u). It is proven in [1] that u is the only eigenvector of W of the form

$$\begin{bmatrix} 0 & \pm 1 & \pm 1 & \dots & \pm 1 \end{bmatrix}^T$$

By assigning all values of u after the first as the diagonal of a matrix U, $G = \{g_j = Uf_j\}_{j=1}^N$ is a companion for F. Specifically, it was conjectured in [1] that this construction works for any d where $d+1$ is prime, and not for any d where $d+1$ is composite. We confirmed this numerically up to $d = 10000$ with the code shown in Appendix D.

Appendix

A Checking Trine Scheme Code

Below is the python code to confirm that F is an ETF and G is a companion in the trine scheme in section 5.1.

```
# code to check that the trine (from DFT) is an ETF and that rotating by 180
# degrees yields a companion frame
import math
import cmath
import sympy
from sympy.abc import a, b, c, d

D = 2 # dimension of vectors
N = 3 # number of vectors

# this theta yields the cube roots of unity
theta = (2*math.pi) / 3
#normalization factor
n = 1 / math.sqrt(2)
# x1+i*y1 = e^(i*theta), cube root of unity
x1 = n*math.cos(theta)
y1 = n*math.sin(theta)
# x2+i*y2 = e^(2*i*theta), cube root of unity
x2 = n*math.cos(2*theta)
y2 = n*math.sin(2*theta)
```

```
# the columns of F form our trine
F = sympy.Matrix([[n, n, n],
                  [n, complex(x2,y2), complex(x1,y1)]])
conj_F = sympy.Matrix([[n, n, n],
                  [n, complex(x2,-y2), complex(x1,-y1)]])



# first verify that we have an ETF
print("checking that F is an ETF:")
for j in range(0, N):
    for k in range(0, N):
        f1 = F.col(j)
        f2 = conj_F.col(k)

        dot = f1[0]*f2[0]+f1[1]*f2[1]
        norm = abs(dot).expand(complex=True)
        norm = norm**2

        # we get numerical rounding errors around 1
        if (j==k) and (1 - norm < 0.0000001):
            norm = 1

        print(f"(f_{k}^*)(f_{j}) = {norm}")

# U for finding companion frame
```

```
U = sympy.Matrix([[1,0], [0,-1]])

# then see if UF is a companion
print("\nchecking that UF is a companion for F:")
for j in range(0, N):
    for k in range(0, N):
        dual = U*F.col(j)
        f2 = conj_F.col(k)

        dot = dual[0]*f2[0]+dual[1]*f2[1]
        norm = abs(dot).expand(complex=True)
        norm = norm**2

        # we get numerical rounding errors around 0
        if (norm < 0.0000001):
            norm = 0

        print(f"(f_{k}^*)(Uf_{j}) = {norm}")
```

B Checking Determinant Code

Below is the python code to find the determinant of the matrix A reference in section 5.2.

```
import math
import cmath
import numpy

N = 4 # number of vectors

# precalculated values to get ETF with overlap = 1/3
r = math.sqrt(0.5 + math.sqrt(1/12))
R = math.sqrt(1 - r**2)

# get the complex part of the frame vectors
theta = math.pi / 4
t_x = r*R*math.cos(theta)
t_y = r*R*math.sin(theta)

# T = e^{i(pi)/4}
T = complex(t_x, t_y)
c_T = T.conjugate()

A = numpy.zeros((N, N), dtype=complex)

A[0,:] = [r**2, T, c_T, R**2]
```

```
A[1,:] = [R**2, c_T, T, r**2]

A[2,:] = [r**2, -T, -c_T, R**2]

A[3,:] = [R**2, -c_T, -T, r**2]


print("determinant of A:")

# use numpy's determinant function from the

# linear algebera package

print(numpy.linalg.det(A))
```

C Checking d=3, N=9 ETF for Companion

Below is the python code that checks if the set $\{X^m T^n f_0\}_{m,n=0}^{3}$ from section 5.3 is an equiangular unit norm frame, and if it might have a companion, based off of the

$$\langle U f_j, f_j \rangle = 0$$

condition.

```
# code to see if the trends in the tetrahedron continue to 3 dimensions
import math
import cmath
import numpy
import sympy
from sympy.abc import a, b, c, d, e, f, g, h, i

D = 3
N = 9 # number of vectors

# functions we want to use from libraries
pi  = math.pi
cos = math.cos
sin = math.sin
mult = numpy.matmul
sqrt = math.sqrt

# values to get an initial vector f0 to generate a frame
```

```
# see reference [5]
thetas = [pi / 3, pi, (5*pi) / 3]

bottom = (1 / sqrt(2)) + 0.0001
top = sqrt(2/3)
for r in numpy.linspace(bottom, top, 100):
    for th1 in thetas:
        for th2 in thetas:
            r_plus  = (r/2) + (1/2)*sqrt(2-3*(r**2))
            r_minus  = (r/2) - (1/2)*sqrt(2-3*(r**2))

            t_x = r_plus*cos(th1)
            t_y = r_plus*sin(th1)

            u_x = r_minus*cos(th2)
            u_y = r_minus*sin(th2)

            T = complex(t_x, t_y)
            c_T = T.conjugate()
            U = complex(u_x, u_y)
            c_U = U.conjugate()

            f0 = [r, T, U]

            # now we want to construct the rest of the frame
            phi = 2*pi / 3
```

```
om_x = cos(phi)
om_y = sin(phi)
omega = complex(om_x, om_y)
omega_sq = omega**2

X = numpy.zeros((D, D), dtype=complex)
X[0,:] = [1, 0, 0]
X[1,:] = [0, omega, 0]
X[2,:] = [0, 0, omega_sq]

T = numpy.zeros((D, D), dtype=complex)
T[0,:] = [0, 0, 1]
T[1,:] = [1, 0, 0]
T[2,:] = [0, 1, 0]

I = numpy.zeros((D, D), dtype=complex)
I[0,:] = [1, 0, 0]
I[1,:] = [0, 1, 0]
I[2,:] = [0, 0, 1]

# F will hold the frame in the columns
F = numpy.zeros((D, N), dtype=complex)

Xs = [I, X, mult(X, X)]
Ts = [I, T, mult(T, T)]
for j in range(0, D):
```

```
    for k in range(0, D):
        col = j*D + k
        F[:, col] = mult(mult(Xs[j], Ts[k]),f0)

# conj F has conjugate of each entry in F
conj_F = numpy.zeros((D, N), dtype=complex)
for j in range(0, D):
    for k in range(0, N):
        conj_F[j, k] = F[j, k].conjugate()

# first check the F is a unit norm equiangular frame
for j in range(0, N):
    for k in range(0, N):
        f1 = F[:, j]
        f2 = conj_F[:, k]

        dot = f1[0]*f2[0]+f1[1]*f2[1]+f1[2]*f2[2]

        # abs() does sqrt of a complex number times its conjugate
        norm = abs(dot)
        norm = norm**2

        # we get numerical rounding errors around 1
        if (j==k) and (1 - norm < 0.0000001):
            norm = 1
```

```
        # uncomment below if you want to confirm that F is an ETF
        # print(f"(f_{k}^*)(f_{j}) = {norm}")

# We are hoping we can find such a U that is unitary
U = sympy.Matrix([[a,b,c], [d,e,f], [g,h,i]])

A = numpy.zeros((N, N), dtype=complex)
for j in range(0, N):
    # each row of coefficients in U is distributed to this list
    row_addends = [F[0,j], F[1,j], F[2,j]]
    f2 = conj_F[:, j]

    # we multiply each term with f^*
    row1_addends = [f2[0]*r1 for r1 in row_addends]
    row2_addends = [f2[1]*r1 for r1 in row_addends]
    row3_addends = [f2[2]*r1 for r1 in row_addends]

    for r1 in range(0, len(row_addends)):
        A[j,r1] = row1_addends[r1]
        A[j,r1+3] = row2_addends[r1]
        A[j,r1+6] = row3_addends[r1]

print("determinant of A:")
# use numpy's determinant function from the
# linear algebera package
det = numpy.linalg.det(A)
```

```
# again, it looks like we are getting some numerical errors close to 0
if det.real < 0.0000000001 and det.imag < 0.0000000001:
    det = 0
print(det)
```

D Finding Companion Code

Below is the C++ code that checks if the method described in section 6 does yield companions for d where $d+1$ is prime, and not if $d+1$ is composite.

```
//code for finding companions for FUNTF of dimn d (with d+1 prime)
//constructed out of the (d+1)x(d+1) DFT matrix, W
#include<iostream>
#include<vector>
#include<math.h>
#include<complex>

// includes legendre() function to compute the legendre symbol for 2 arguments
// and the prime() function that returns true if the arg is prime
#include "legendre.h"

// check if there is a companion frame for d+1 vectors using an eigenvector
// from the DFT matrix as the diagonal entries in a diagonal unitary matrix
void check_for_companion(int d) {
/*
--> step 1
 make F, an equiangular frame from the columns of W with the 1st row and
 column removed
 scale all the vectors by 1/2 (need to confirm)
*/
    int p = d+1;
    double tau = 2*M_PI;
```

```
    std::complex<double> omega = std::polar(1.0, tau/p);

    int i, j, k; // iterator

    std::complex<double> *F = new std::complex<double>[p*d];

    //double factor = 0.5; // sqrt(5) term is dropped
    double factor = 1;
    // the first column of W is all 0, don't use this
    for (i = 0; i < p; i++) {
        for (j = 1; j <= d; j++) {
            // bad spatial locality, but works
            F[i + p*(j-1)] = factor * std::pow(omega, (i*j)%p);
        }
    }

/*
--> step 2
 construct the eigenvector of W,
 v = [0, leg(1, p), leg(2, p), ..., leg(p-1, p)]
 where legendre() is the legendre symbol
 we can ignore the first 0
 NOTE: it is proven that such an eigenvector is the only eigenvector of W
 of the form [0, +-1, ..., +-1]*
*/
    double *v = new double[d];
```

```
 for (i = 0; i < d; i++) {
     v[i] = legendre(i+1, d+1);
 }

//--> step 4
//construct the companion frame G = {Uf : each f in F}
//where U is a diagonal d x d matrix whose diagonal is v

 std::complex<double> *G = new std::complex<double>[p*d];

 for (i = 0; i < p; i++) {
     for (j = 0; j < d; j++) {
         G[i + p*j] = F[i + p*j]*(double)v[j];
     }
 }

 //--> step 5
 //confirm that norm(G[k], F[l])^2 = c if k is not equal to l and 0 if k=l
 //NOTE: the examples in the paper both have c = (d+1)/(d^2)

 double overlap;
 double constant_overlap;
 bool nonzero_found = false;
 std::complex<double> c;
```

```
for (i = 0; i < p; i++) {
    for (k = 0; k < p; k++) {
        c = 0;
        for (j = 0; j < d; j++) {
            c += G[i + p*j] * conj(F[k + p*j]);
        }
        overlap = abs(c);
        overlap *= overlap;

        // we are getting numerical errors for overlap very clsoe to 0
        if (overlap < 0.000001) overlap = 0;

        if ((!nonzero_found) && (i != k)) {
            constant_overlap = overlap;
            nonzero_found = true;
        }

        // again, numerical erros imply the need for a close to zero check
        if ((abs(overlap - constant_overlap) > 0.000001) && (i != k)) {
            if (prime(p)) {
                std::cout << p << " is prime and " << d <<
                "does not have a companion";
            }

            //uncomment below if you want more details about
            //the overlap in the output
```

```
                /*
                std::cout << d << " DOES NOT HAVE A COMPANION. overlap: "
                          << overlap << ", not equal to constant: "
                          << constant_overlap << std::endl;
                */
                return;
            }
        }
    }

    if (!prime(p))
        std::cout << d << " is not prime but does have a companion" << std::endl;

    //uncomment below if you want to see whenever there is a companion for d
    /*
    std::cout << d << " DOES HAVE A COMPANION."
    */

    delete [] v;
    delete [] F;
    delete [] G;
}
```

E References

[1] R. BALU, P. J. KOPROWSKI, K. A. OKOUDJOU, J. S. PARK, AND G. SIOPSIS, *Equiangular quantum key distribution in more than two dimensions*, Journal of Physics A: Mathematical and Theoretical, (2019).

[2] C. BENNETT AND G. BRASSARD, *Quantum cryptography: Public key distribution and coin tossing*, International Conference on Computers, Systems, and Signal Processing, (1984).

[3] D. HAN, K. KORNELSON, D. LARSON, AND E. WEBER, *Frames for Undergraduates*, The American Mathematical Society, 2007.

[4] J. M. RENES, *Spherical code key distribution protocols for qubits*, Physics Review A, (2004).

[5] J. M. RENES, R. BLUME-KOHOUT, A. J. SCOTT, AND C. M. CAVES, *Symmetric informationally complete quantum measurements*, Journal of Mathematical Physics, 45 (2003), pp. 2171–2180.

[6] T. STROHMER AND R. W. HEATH, *Grassmannian frames with applications to coding and communication*, Applied and Computational Harmonic Analysis, 14 (2003), pp. 257–275.

[7] H. TERASHIMA AND M. UEDA, *Nonunitary quantum circuit*, International Journal of Quantum Information, 3 (2005).

[8] R. TUMULKA, *POVM (Positive-Operator-Valued Measure)*, Springer, 01 2009, pp. 480–484.

[9] N. YANOFSKY AND M. MANNUCCI, *Quantum computing for computer scientists*, Cambridge Press, 2008.