Endoscopic Codes For Unitary Groups Over The Reals

By Dmitry Rubanovich

A thesis submitted to the

Graduate School-Newark

Rutgers, The State University of New Jersey

for the degree of

Philosophy Doctor

Graduate Program in Mathematical Sciences

Written under the direction of

Professor Diana Shelstad

and approved by

Newark, New Jersey

October, 2009

ABSTRACT OF THE DISSERTATION

Endoscopic Codes For Unitary Groups Over The Reals

By DMITRY RUBANOVICH

Dissertation Director:

Diana Shelstad

Transfer factors, originally defined by Langlands and Shelstad for the transfer of orbital integrals, play a central role in the theory of endoscopy. Spectral transfer factors, for the dual transfer of traces, have been defined for real groups by Shelstad. The theory shows that for discrete series representations of unitary groups the spectral transfer factors determine a bijection between the representations in a packet and certain binary words. The binary word thus associated to a representation may be called its *endoscopic code*. Such a code is difficult to calculate from the definition by transfer factors. Low dimensional examples suggest that there is an alternative approach, directly in terms of the Harish-Chandra data of the representation, which provides fast calculation of spectral transfer factors.

This thesis presents a new direct construction of the endoscopic code of a discrete series representation of any unitary group directly from its Harish-Chandra data and, conversely, identifies a discrete series representation from any particular given endoscopic code. An explicit algorithm is given and implemented in Mathematica™ .

Acknowledgement

I would like to thank Rutgers University, Newark, for the support during the studies and research that lead to this dissertation. I would also like to thank the Department of Mathematics and Computer Science at Rutgers, Newark, for this opportunity.

Lastly, while it is a given that every PhD student's work stems from the work of their adviser, full credit would not be given if I did not also thank Professor Shelstad both for her ability to bring clarity to a number of immeasurably sophisticated topics and for her patience with me at times when providing such clarity must had seemed an insurmountable task.

Thank you.

Contents

1 Introduction

Endoscopic codes are binary words related to the tranfer factors which arise in the theory of endoscopy. Transfer factors were originally defined by Langlands and Shelstad for the transfer of orbital integrals in [LS87]. Spectral transfer factors, for the dual transfer of traces, have been defined for real groups in [She09]. Their definition is similarly complicated.

We study discrete series representations of unitary groups over the field of real numbers. The unitary groups $U(p, q)$, which are described in Section 2.2, will be clustered into K-groups. Suppose $n = p + q$ is fixed. The simple case is that of n odd. In this case, there is exactly one copy of each real form $U(p, q)$ in the K-group. In the case of n even, there are two K-groups and two copies of each real form in the K-group containing them, with the exception of the quasi-split form $U(\frac{n}{2}, \frac{n}{2})$. The idea of considering several groups together is due to Vogan. The version we use is due to Kottwitz, as reported by Arthur in [Art99].

Each discrete series representation of a K-group of unitary groups has an endoscopic code defined using the theory of transfer factors ([She08]). The coding is unique for n odd, and is unique module obvious modifications for n even. In this thesis we give a new construction of the code directly in terms of the Harish-Chandra data of the representation. We give an explicit algorithm and implement it in Mathematica$^{\text{TM}}$.

Discrete series representations have various concrete realizations. We will identify them by their characters, using Harish-Chandra's existence and uniqueness results in "Discrete series for semisimple Lie groups II." [HC66].

The references to the details of these results appear in the text as references to [Wal98].

A particular character is determined by Harish-Chandra's data comprised of integers $m_1 > m_2 > \ldots > m_n$, with m_i's parities opposite to the parity of n (which fixes the infinitesimal character); a particular real form $U(n-j, j)$; and a particular coset element in the quotient $\Omega/\Omega^j_{\mathbb{R}}$ of complex Weyl group by the real Weyl group of that form.

The construction of the codes is based on the fact that the multiplicative group of diagonal matrices with ± 1 on diagonals is isomorphic to the bit-wise "exclusive or" operation on binary strings. The codes are constructed by adding a binary string that is unique to a coset $\omega\Omega^j_{\mathbb{R}}$ for a fixed j and a string that provides a unique twist which is distinct for all n, j. The construction of the strings is different for each K-group.

A general application is the calculation of transfer factors. As a particular application of our method, we prove a separation lemma for discrete series L-packet of the real unitary group $U(n-1, 1)$.

2 Unitary Groups

2.1 Classical Unitary Group

There is a number of equivalent ways to define the classical unitary group of $n \times n$ matrices. Probably the most natural way in which it arises is as the self-adjoint invariant of the standard hermitian inner product on the n-dimensional inner space. That is as the set of all matrices $\mathbf{U}$ such that $< v\mathbf{U}, w\mathbf{U} > = < v, w >$ for all $v, w \in \mathbb{C}^n$. Defining more precisely what the standard hermitian inner product $<\cdot,\cdot>$ means, gives $<a,b>=a^t\overline{b}$. Rewritting the definition of self-adjoint in this manner, says that the unitary group is all matrices $\mathbf{U}$ that satisfy $v^t w = v\mathbf{U}^t\overline{(w\mathbf{U})}$ for all $v, w \in \mathbb{C}^n$. Which immediately gives the definition of unitary matrices as those which satisfy

$$\mathbf{U}^t\overline{\mathbf{U}} = \mathbf{I} \tag{0}$$

Let v be an eigenvector of $\mathbf{U}$ with eigenvalue λ. Then

$$0 < \quad < v, v > = < v\mathbf{U}, v\mathbf{U} > = < \lambda v, \lambda v > = \lambda\overline{\lambda} < v, v >$$

So

$$\lambda\overline{\lambda} = 1, \tag{0}$$

for all eigenvalues λ of $\mathbf{U}$.

It follows immediately from (2.1) that all elements of the unitary group must be diagonalizable. So every element of an n-dimensional unitary group, $\mathbb{U}(n)$, can be written as

$$\mathbf{U} = \mathbf{E}\begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}\mathbf{E}^{-1} = \mathbf{E}\begin{pmatrix} e^{i\theta_1} & & \\ & \ddots & \\ & & e^{i\theta_n} \end{pmatrix}\mathbf{E}^{-1},$$

where the rows of $\mathbf{E}$ are orthonormal eigenvectors of U. So $\mathbf{E}$ is unitary. Let $\mathbf{D}$ denote the diagonal matrix in this decomposition. Then we have the following:

$$1 = \det(\mathbf{U}^t\overline{\mathbf{U}}) = \det(\mathbf{E}\mathbf{D}\mathbf{E}^{-1}\mathbf{E}^t\overline{\mathbf{D}}\mathbf{E}^{-1}) = \det(\mathbf{D}^t\overline{\mathbf{D}})$$

The first equality is a polynomial condition on entries of elements of $\mathbb{U}(n)$. It is, however, satisfied not only by the elements of $\mathbb{U}(n)$, but by all of

$$\mathbb{SL}(n,\mathbb{R})\mathbb{U}(n) = \{n \times n \textit{ diagonal matrices in } \mathbb{SL}(n,\mathbb{R})\}\mathbb{U}(n)$$

We want to define $\mathbb{U}(n)$ as an algebraic group. Following [GW98], we define

$$\mathbf{J} = \left[\begin{array}{c|c} & -\mathbf{I} \\ \hline \mathbf{I} & \end{array}\right]$$

and associate $\mathbb{M}(n, \mathbb{C})$ with the elements $\mathbf{A}$ of $\mathbb{M}(2n, \mathbb{R})$ such that $\mathbf{AJ} = \mathbf{JA}$. Which, of course, yields the usual

$$\left[\begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline -\mathbf{B} & \mathbf{A} \end{array}\right].$$

as the $\mathbb{M}(2n, \mathbb{R})$ equivalent of $\mathbf{A} + i\mathbf{B}$. The advantage of this association is that it allows complex conjugation to be treated as a permutation of matrix elements. So it allows for polynomials not only in $\mathbb{C}[x_{11}, x_{12}, \ldots, x_{nn}]$ but also in $\mathbb{C}[x_{11}, x_{12}, \ldots, x_{nn}, \overline{x_{11}}, \overline{x_{12}}, \ldots, \overline{x_{nn}}]$ to be used in defining algebraic groups. In particular, (2.1) can now be regarded as an *algebraic* definition of $\mathbb{U}(n)$.

[GW98, sec 1.4.1] also shows that under this association Lie algebra of any Lie group in $\mathbb{GL}(n, \mathbb{C})$ is the same as the Lie algebra of the Lie subgroup of $\mathbb{GL}(2n, \mathbb{R})$ associated to it. That is, if we call the association map r:

$$r : \mathbf{A} + i\mathbf{B} \to \left[\begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline -\mathbf{B} & \mathbf{A} \end{array}\right],$$

then $\mathrm{Lie}(r(G)) = r(\mathrm{Lie}(G))$ for any Lie group G in $\mathbb{GL}(n, \mathbb{C})$.

2.2 $\mathbb{U}(p, q)$

2.2.1 Real Forms

We continue following definitions in [GW98]. A linear algebraic group $G \subset \mathbb{GL}(n, \mathbb{C})$ is said to be **defined over** $\mathbb{R}$ if the polynomials which generate it as a variety produce points in $\mathbb{R}$ when these polynomials are evaluated at $\mathbb{GL}(n, \mathbb{R})$ (that is when $\mathbf{B}$ above is 0 or, equivalently, when all the polynomials have real coefficients).

The $G_{\mathbb{R}} = G \cap \mathbb{GL}(n, \mathbb{R})$ is the **$\mathbb{R}$-rational points of** G.

Define

$$\mathrm{Aff}(G) = \bigcup_{j \geq 0} \det{}^{-j}\, \mathbb{C}[x_{11}, x_{12}, \ldots, x_{nn}, \overline{x_{11}}, \overline{x_{12}}, \ldots, \overline{x_{nn}}],$$

where $x_{j,k}$ are matrix entries of elements of G. Let τ be a *group homomorphism* (and not necesserily a *Lie* group homomorphism) of G. Then τ is called a **complex conjugation** of G if $f^{\tau}(\cdot) \stackrel{def}{=} \overline{f(\tau(\cdot))}$ is in $\mathrm{Aff}(G)$ for all $f \in \mathrm{Aff}(G)$. A subgroup H of G is called a **real form** of G if H is fixed by *some* complex conjugation of G.

For example τ defined by $\tau : \mathbf{g} \mapsto {}^t\overline{\mathbf{g}}^{-1}$ (which happens to be identity on $\mathbb{U}(n)$) is a complex conjugation of $\mathbb{U}(n)$ because for any $f \in \mathrm{Aff}(\mathbb{U}(n))$, $\overline{f(\tau(\mathbf{g}))} = \overline{f}(\overline{\mathbf{g}})$ and the definition of Aff allows for polynomials in both matrix entries *and their complex conjugates* ($\overline{f}$ here stands for polynomial with coefficients conjugate to the coefficients of f).

2.2.2 Compact and Non-compact Forms

Let G be a *linear* group. Let $\{b_\alpha\}$ be a set of basis of the linear space on which G operates. Then **matrix coefficient** $a_{\alpha\beta}$ of $\mathbf{A} \in G$ is $< b_\alpha \mathbf{A}, b_\beta >$.

A group is compact if it is topologically closed and all matrix coefficients are bounded.

A **compact form** is a real form that is compat as a linear group. $\mathbb{U}(n)$ above is an example of a compact form. A **non-compact form** is a linear form that is not compact as a group.

In what follows, we define complex conjugation $\boldsymbol{\sigma}$ as $\boldsymbol{\sigma}(\mathbf{A}) = \mathbf{I}_{p,q}\overline{{}^t\mathbf{A}^{-1}}\mathbf{I}_{p,q}$, where

$$\mathbf{I}_{p,q} = \left[\begin{array}{c|c} \mathbf{I}_p & \\ \hline & -\mathbf{I}_q \end{array}\right]$$

and $\mathbf{I}_p$,$\mathbf{I}_q$ are identity matrices of indexed sizes. Which p and q are used for a particular $\boldsymbol{\sigma}$ will be clear from the context. In cases where it will not be clear (or more than one type of $\boldsymbol{\sigma}$ is used), they'll be labeled.

We label the real form corresponding to the complex conjugation $\boldsymbol{\sigma}_j$ as $\mathbb{U}(n-j,j)$. It happens to be an example of a non-compact form. To see this, consider a matrix that has elements a, b, c, d in positions $(i,i), (i,k), (k,i), (k,k)$, 1's in other places on the diagonal and 0's everywhere else:

$$\mathbf{A} = \begin{bmatrix} 1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & 1 & & & & & & \\ & & & a & \cdots & b & & & \\ & & & \vdots & \ddots & \vdots & & & \\ & & & c & \cdots & d & & & \\ & & & & & & 1 & & \\ & & & & & & & \ddots & \\ & & & & & & & & 1 \end{bmatrix}$$

and the effect of $\boldsymbol{\sigma}_j$ (where $i \leq n-j < k \leq n$). To simplify the calculations, let $G = \mathbb{SL}(n,\mathbb{C})$ rather than $\mathbb{GL}(n,\mathbb{C})$. This is justified because showing that $\mathbb{SL}(n,\mathbb{C}) \cap \mathbb{U}(n-j,j)$ is unbounded will surely show that $\mathbb{U}(n-j,j)$ is unbounded. Because the only entries effected will be a, b, c, d, we can, without loss of generality, consider $n = 2, j = 1$. So $\mathbf{I}_{n-j,j} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ and $\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$. Calculating $\boldsymbol{\sigma}(\mathbf{A})$ gives

$$\boldsymbol{\sigma}(\mathbf{A}) = \begin{bmatrix} \overline{d} & \overline{c} \\ \overline{c} & \overline{d} \end{bmatrix}$$

Elements of the real form are the matrices that satisfy $\boldsymbol{\sigma}(\mathbf{A}) = \mathbf{A}$. This restriction yields the relation $a = \overline{d}$ and $b = \overline{c}$. So

$$\mathbb{SU}(1,1) \equiv \mathbb{SL}(2,\mathbb{C}) \cap \mathbb{U}(1,1) = \left\{ \begin{bmatrix} a & b \\ \overline{b} & \overline{a} \end{bmatrix} \in \mathbb{SL}(2,\mathbb{R}) \mid a\overline{a} - b\overline{b} = 1 \right\}$$

$a = \cosh(\theta)$, $b = \sinh(\theta)$ satisfy $a\overline{a} - b\overline{b} = 1$ for all $\theta \in \mathbb{R}$. And sinh is unbounded. So $\mathbb{U}(n-j,j)$ (for $j > 0$), which contains elements of the form

$$\mathbf{A} = \begin{bmatrix} 1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & 1 & & & & & & \\ & & & \cosh(\theta) & \cdots & \sinh(\theta) & & & \\ & & & \vdots & \ddots & \vdots & & & \\ & & & \sinh(\theta) & \cdots & \cosh(\theta) & & & \\ & & & & & & 1 & & \\ & & & & & & & \ddots & \\ & & & & & & & & 1 \end{bmatrix}$$

is non-compact.

2.2.3 Complex Form

A **complex form** of a Lie group G is a Lie group G' such that the *complex* Lie algebra $\mathfrak{g}'$ of G' is equal to the complexification of the *real* Lie algebra $\mathfrak{g}$ of G. So $\mathfrak{g}' = \mathfrak{g}_{\mathbb{C}} = \mathfrak{g} \oplus i\mathfrak{g}$.

In case of $G = \mathbb{U}(p, q)$,

$$\mathfrak{g} = \left[\begin{array}{ccccc|ccccc} ia_1 & b_{12} & & & & & & & & \\ -\overline{b_{12}} & ia_2 & & & & & & & & \\ & & \ddots & & & & & C & & \\ & & & ia_{p-1} & b_{p-1,p} & & & & & \\ & & & -\overline{b_{p-1,p}} & ia_p & & & & & \\ \hline & & & & & ia_{p+1} & b_{p+1,p+2} & & & \\ & & & & & -\overline{b_{p+1,p+2}} & ia_{p+2} & & & \\ & & -{}^t\overline{C} & & & & & \ddots & & \\ & & & & & & & & & ia_{p+q} \end{array}\right]$$

As a real vector space it has 1 imaginary dimension for each diagonal entry and 1 complex dimension for each entry above the diagonal. As a real vector space $i\mathfrak{g}$ adds 1 real dimension for each diagonal entry and 1 complex dimension for each entry above the diagonal. For a total of $(p+q)^2$ complex dimensions. So it's must take up all of $\mathfrak{gl}$. Thus the complex form of $\mathbb{U}(n-j, j)$ is $\mathbb{GL}(n, \mathbb{C})$ for all j. Given a real form G, [She08] refers to G as **real points** of G and to complex form as **complex points** of G.

2.3 Maximal Compact and Diagonal Subgroups

$\mathbb{U}(n-j) \times \mathbb{U}(j)$ is a maximal compact subgroup of $\mathbb{U}(n-j, j)$. Because $\boldsymbol{\sigma}$ preserves the real points of the diagonal subgroup of $\mathbb{GL}(n, \mathbb{C})$, it preserves the diagonal subgroup of each $\mathbb{U}(n-j, j)$ and each $\mathbb{U}(n-j) \times \mathbb{U}(j)$. Thus all $\mathbb{U}(n-j, j)$ share the compact Cartan subgroup $T(\mathbb{R}) = \{diag(e^{i\theta_1}, \ldots, e^{i\theta_n})\}$.

2.4 $\Omega / \Omega_{\mathbb{R}}^j$

The **complex Weyl group**, Ω, is the the group of permutation of roots of the *complex form* (which, again, coincides for all $\mathbb{U}(n-j, j)$) $\mathbb{GL}(n, \mathbb{C})$. This is the group of all permutations of $n \times n$ matrices, S_n. The **real Weyl group**, $\Omega_{\mathbb{R}}^j$, is the group of permutation of roots of the *real form* $\mathbb{U}(n-j, j)$ by a matrix in $\mathbb{U}(n-j, j)$. It also happens to be the Weyl group of $\mathbb{U}(n-j) \times \mathbb{U}(j)$:

$$\Omega_{\mathbb{R}}^j = \left[\begin{array}{c|c} S_{n-j} & \\ \hline & S_j \end{array} \right].$$

2.5 Roots and Coroots

As basis for the simple root and coroot system, we pick

$$t_j : diag(t_1, t_2, \ldots, t_n) \mapsto t_j$$

and

$$z_k : diag(z_1, z_2, \ldots, z_n) \mapsto z_k.$$

Then $< t_j, z_k >= \delta_{jk}$. The **standard** simple positive root system is comprised of $t_1 - t_2, t_2 - t_3, \ldots, t_{n-1} - t_n$. For any $\mathbb{U}(n-j) \times \mathbb{U}(j)$, the $t_i - t_j$ will be called a **compact root** if $t_i - t_j$ lies in the copy of $\mathbb{U}(n-j)$ or the copy of $\mathbb{U}(j)$. Roots which are not compact roots will be called **noncompact roots**.

Compact Roots:

$$\left[\begin{array}{ccccc|c} \ddots & & & & & \\ & t_i & & & & \\ & & \ddots & & & \\ & & & -t_j & & \\ & & & & \ddots & \\ \hline & & & & & 0 \end{array}\right] \quad \left[\begin{array}{c|ccccc} 0 & & & & & \\ \hline & \ddots & & & & \\ & & t_i & & & \\ & & & \ddots & & \\ & & & & -t_j & \\ & & & & & \ddots \end{array}\right]$$

Noncompact Roots:

$$\left[\begin{array}{ccc|ccc} \ddots & & & & & \\ & t_i & & & & \\ & & \ddots & & & \\ \hline & & & \ddots & & \\ & & & & -t_j & \\ & & & & & \ddots \end{array}\right]$$

3 Discrete Series Representations of Unitary Groups

3.1 Discrete Series

A **discrete series** representation of a topological group G is one of irreducible smooth representations of G which are square-integrable modulo the center of G (with respect to an appropriate Haar measure). A represenation is square-integrable in the sense that all its matrix coeffiecients are square-integrable.

Since discrete series representation determines a representation on $G/Z(G)$, it can be used, together with a character on the center $Z(G)$ of G, to construct a representation on G.

Discrete series representations of a reductive group can be classified by classifying all equivalent irreducible represenations. *Finite dimensional* irreducible discrete series representations of $\mathbb{GL}(n, \mathbb{C})$ arise as representations on compact form $\mathbb{U}(n)$ while the *infinite dimensional* ones arise as representations on noncompact forms $\mathbb{U}(n-j, j)$.

3.2 Compact Real Form

3.2.1 Weyl's Unitary Trick

Using Weyl's unitary trick, it can be shown that finite dimensional representations of $\mathbb{GL}(n, \mathbb{C})$ correspond to finite dimensional representations of $\mathbb{U}(n)$ ([Kna05]).

Because Weyl's unitary trick requires the compact real form to be simply-connected, this requires considering $\mathbb{GL}(n, \mathbb{C})$ as $Z\mathbb{SL}$ and $\mathbb{U}(n)$ as $Z_0\mathbb{SU}$, where Z is the group of scalar matrices of $\mathbb{GL}$ and Z_0 is the group of scalar matrices of $\mathbb{U}$. $\mathbb{SU}(n, \mathbb{C})$ is a compact real form of $\mathbb{SL}(n, \mathbb{C})$. [Hal04] shows construction of highest weight representations of $\mathfrak{su}(2, \mathbb{C})$ and $\mathfrak{su}(3, \mathbb{C})$. The $\mathfrak{su}(3, \mathbb{C})$ case shows the inductive step necessery to construct $\mathfrak{su}(n, \mathbb{C})$.

Clearly, Z is complexification of Z_0:

$$Z = \left\{e^{\theta_1 + i\theta_2}\mathbf{I}\right\} \qquad Z_0 = \left\{e^{\theta}\mathbf{I}\right\}$$

Let T denote a compact connected abelian Lie subgroup of $\mathbb{U}$, it is a **torus** of $\mathbb{U}$. We'll, further, denote its Lie algebra by $\mathfrak{t}$ and the complexification of $\mathfrak{t}$

by $\mathfrak{t}_{\mathbb{C}}$. Because $\mathfrak{t}_{\mathbb{C}}$ is also the Cartan subalgebra of $\mathfrak{sl}$, we may at times also denote it by $\mathfrak{h}$.

3.2.2 Harish-Chandra's Isomorphism

Harish-Chandra's isomorphism allows for a basis-independent calculaton of a infinitesmal character on an element of $\mathfrak{h}$. The full construction with proofs is shown in [Kna05, pp 300-313], but we'll show the calculation here. This construction works for an arbitrary *complex* semisimple Lie algebra $\mathfrak{g}$ with Cartan subalgebra $\mathfrak{h}$. So, in particular, it works for the $\mathfrak{h}$ we are using.

Given a Lie algebra $\mathfrak{a}$, let $U(\mathfrak{a})$ denote its universal algebra. Define $\mathcal{H} = U(\mathfrak{h})$. $\mathcal{H}$ is the symmetric algebra of $\mathfrak{h}$. Let $\mathcal{H}^W$ denote the orbits of Weyl action on $\mathcal{H}$.

There are two facts that simplify classification of characters of an irreducible finite-dimensional represenation of $\mathbb{U}(n)$.

First, by Schur's Lemma, every element of the center of must act by a scalar. Denote the center of $U(\mathfrak{g})$ by $Z(\mathfrak{g})$. Extend action of representation π of $\mathfrak{g}$ to action on all of $U(\mathfrak{g})$. For any element $z \in Z(\mathfrak{g})$, $\pi(zu) = \chi(z)\pi(u)$. χ is the **infinitesmal character** of π. Because they must agree on the center, $\pi(z) = \chi(z)\mathbf{I}$.

Second, each element of the compact connected group $\mathbb{SU}(n, \mathbb{C})$ is conjugate to an elment of T by [Kna05, thm 4.36]. Since $\operatorname{tr}\pi(\cdot)$ is invariant under the operation of taking a complex conjugate of the argument, all equivalent irreducible finite-dimensional representations of $\mathbb{SU}(n, \mathbb{C})$ can be classified by classifying irreducible finite-dimensional representations of T. Because we have a choice of conjugates as candidates for T, we may as well pick the one that simplifies the calculations – the one that coincides with the diagonal subgroup of $\mathbb{SU}$.

Let λ be the highest weight of representation of $\mathfrak{g}$ with highest weight vector v_λ (ie, the module $U(\mathfrak{g})v_\lambda$). We can extend action of $\beta \in \mathfrak{h}^*$ to action on $\mathcal{H}$ by $< \beta, H_1 \otimes H_2 \otimes \ldots \otimes H_i > = < \beta, H_1 >< \beta, H_2 > \ldots < \beta, H_i >$. Which makes β "plug-in" values into elements of $\mathcal{H}$. So, in particular, $H_1^{p_1} H_2^{p_2} \ldots H_i^{p_i}$ acts by $\lambda(H_1)^{p_1}\lambda(H_2)^{p_2} \ldots \lambda(H_i)^{p_i}$ on v_λ.

Define Δ to be a root system with respect to Lie group $\mathfrak{g}$ and Cartan subgroup $\mathfrak{h}$. Let Δ^+ be a collection of positive roots of Δ. Let E_α denote a root vector of root α with respect to this root system. Define $\mathcal{P} = \sum_{\alpha\in\Delta^+} U(\mathfrak{g})E_\alpha$ and $\mathcal{N} = \sum_{\alpha\in\Delta^+} E_{-\alpha}U(\mathfrak{g})$. Finally, denote the half-sum of positive roots by $\delta = \frac{1}{2}\sum_{\alpha\in\Delta^+} \alpha$.

[Kna05] shows that $U(\mathfrak{g}) = \mathcal{H} \oplus (\mathcal{P} + \mathcal{N})$. Denote the projection on $\mathcal{H}$ by

$$\gamma' : U(\mathfrak{g}) \to \mathcal{H}.$$

Let $\tau(\mathbf{H}) = \mathbf{H} - \delta(\mathbf{H})\mathbf{I}$. $\gamma = \tau \circ \gamma'$ is The Harish-Chandra's isomorphism

$$\gamma : Z(\mathfrak{g}) \to \mathcal{H}^W.$$

And infinitesmal character can be calculated as $\chi_\lambda(z) = \lambda(\gamma(z))$. The infinitesmal character of the representation $U(\mathfrak{g})v_\lambda$ is $\chi_{\lambda+\delta}$.

3.2.3 $\mathfrak{su}(3)$ example of Harish-Chandra Isomorphism

$$H_1 = \begin{bmatrix} 1 & & \\ & -1 & \\ & & 0 \end{bmatrix} \qquad H_2 = \begin{bmatrix} 0 & & \\ & 1 & \\ & & -1 \end{bmatrix}$$

$$X_1 = \begin{bmatrix} 0 & 1 & \\ & 0 & \\ & & 0 \end{bmatrix} \quad X_2 = \begin{bmatrix} 0 & & \\ & 0 & 1 \\ & & 0 \end{bmatrix} \quad X_3 = \begin{bmatrix} 0 & & 1 \\ & 0 & \\ & & 0 \end{bmatrix}$$

$$Y_1 = \begin{bmatrix} 0 & & \\ 1 & 0 & \\ & & 0 \end{bmatrix} \quad Y_2 = \begin{bmatrix} 0 & & \\ & 0 & \\ & 1 & 0 \end{bmatrix} \quad Y_3 = \begin{bmatrix} 0 & & \\ & 0 & \\ 1 & & 0 \end{bmatrix}$$

Using the adjoint relations calculated in [Hal04, ch 5.2], we calculate the Killing form $B(M,N) = \mathrm{tr}(\mathrm{ad}\, M\, \mathrm{ad}\, N)$ to be

	H_1	H_2	X_1	X_2	X_3	Y_1	Y_2	Y_3
H_1	12	-6						
H_2	-6	12						
X_1						6		
X_2							6	
X_3								6
Y_1			6					
Y_2				6				
Y_3					6			

We calculate dual basis such that $B(\tilde{M}_i, M_j) = \delta_{ij}$ to be

$$\tilde{H}_1 = \frac{1}{9}H_1 + \frac{1}{18}H_2 \qquad \tilde{H}_2 = \frac{1}{18}H_1 + \frac{1}{9}H_2$$

$$\tilde{X}_1 = \frac{1}{6}Y_1 \qquad \tilde{X}_2 = \frac{1}{6}Y_2 \qquad \tilde{X}_2 = \frac{1}{6}Y_3$$

$$\tilde{Y}_1 = \frac{1}{6}X_1 \qquad \tilde{Y}_2 = \frac{1}{6}X_2 \qquad \tilde{Y}_2 = \frac{1}{6}X_3$$

Finally, we can calculate the **Casimir element** Ω as in [Kna05, p293]:

$$\Omega = \sum_{i,j} B(M_i, M_j)\tilde{M}_i\tilde{M}_j = \frac{H_1^2}{9} + \frac{H_2H_1}{18} + \frac{H_1H_2}{18} + \frac{H_2^2}{9} + \sum_{i=1}^{3} \frac{Y_iX_i}{6} + \sum_{i=1}^{3} \frac{X_iY_i}{6}$$

using $X_iY_i = [X_i, Y_i] + Y_iX_i$ (for $i = 1, 2$), $H_1H_2 = H_2H_1$, $[X_3, Y_3] = H_1 + H_2$ and making the choice that the positive root vectors are X_i

$$= \frac{H_1^2}{9} + \frac{H_1H_2}{9} + \frac{H_2^2}{9} + \frac{H_1}{3} + \frac{H_2}{3} + \underbrace{\sum_{i=1}^{3} \frac{Y_iX_i}{6}}_{\text{in } \mathcal{N}+\mathcal{P}}$$

So $\gamma'(\Omega) = \frac{H_1^2}{9} + \frac{H_1H_2}{9} + \frac{H_2^2}{9} + \frac{H_1}{3} + \frac{H_2}{3}$.

$$\delta = \frac{\alpha_1 + \alpha_2 + \alpha_3}{2}$$

gives $\delta(H_1) = 1$, $\delta(H_2) = 1$. So

$$\tau(\gamma'(\Omega)) = \frac{(H_1 - 1)^2}{9} + \frac{(H_1 - 1)(H_2 - 1)}{9} + \frac{(H_2 - 1)^2}{9} + \frac{(H_1 - 1)}{3} + \frac{(H_2 - 1)}{3}$$

$$= \frac{1}{9}(H_1^2 + H_1H_2 + H_2^2 - 3)$$

The Weyl group is generated by $\omega_1(H_1) = -H_1$, $\omega_2(H_2) = -H_2$, which are reflections across the line perpendicular to the respective roots in the $\mathfrak{su}(3, \mathbb{C})$ root-system diagram. From the same diagram it can be read off that $\omega_1(H_2) = H_1 + H_2$ and $\omega_2(H_1) = H_1 + H_2$.

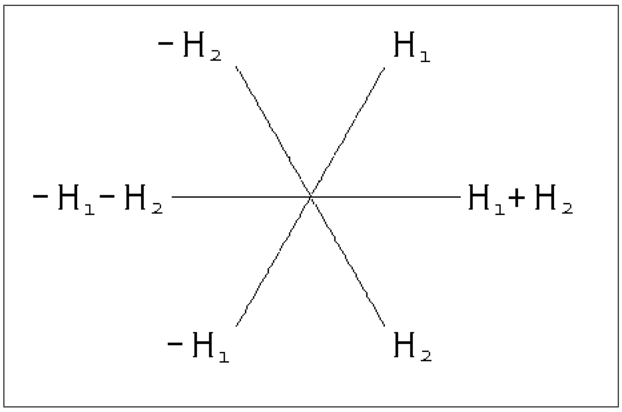

Figure 1: $\mathfrak{gl}(3, \mathbb{C})$ root system

An algebraic calculation shows that $\omega_i(H_1^2+H_1H_2+H_2^2) = H_1^2+H_1H_2+H_2^2$. So $\gamma(\Omega) = \tau(\gamma'(\Omega))$ is, indeed, in $\mathcal{H}^W$.

Because $Z(\mathfrak{g})$ is generated by Ω, the fact that Harish-Chandra's map is an isomorphism shows that $\mathcal{H}^W$ is $\mathbb{C}[H_1^2 + H_1H_2 + H_2^2]$.

3.2.4 Reparametrization

Because any element of the center of $\mathbb{U}(n)$ is a scalar multiple of a central element of $\mathbb{SU}(n)$, adding additional parameter and normalizing allows for an n-parameter *highest weight* representation of the center of $\mathbb{U}(n)$. [GW98, theorem 5.2.1] does this construction.

The highest weight character can be parametrized by tuples $(m_1', m_2', \dots, m_n')$ such that $m_i' \in \mathbb{Z}$ and $m_1' \geq m_2' \geq \dots \geq m_n'$. So that

$$\chi'(\mathbf{t}) = t_1m_1' + t_2m_2' + \cdots + t_nm_n'$$

for $\mathbf{t} \in \mathfrak{t}_{\mathbb{C}}$.

Using the Harish-Chandra's Isomorphism [She08] calculates that the highest weight representation parametrized by $(m_1', m_2', \dots, m_n')$ corresponds to the ininitesmal character $\chi(m_1, m_2, \dots, m_n)$, where

$$\chi(m_1, m_2, \dots, m_n)(\mathbf{t}) = t_1(\frac{m_1}{2}) + t_2(\frac{m_2}{2}) + \cdots + t_n(\frac{m_n}{2})$$

and

$$m_1 = 2m_1' + n - 1, m_2 = 2m_2' + n - 3, \dots, m_n = m_n' - n + 1,$$

where $m_1 > m_2 > \dots > m_n$ and (clearly) m_i's have parity opposite of n's.

To elaborate how this calculation arises out of the Harish-Chandra's isomorphism, it's the image of the $\chi_{\lambda+\delta}(z)$. So $\frac{n-2i+1}{2}$ arises as $\delta(t_i)$. The easiest way to see why is to produce a list of positive roots (because $\delta(t_i)$ is a half-sum of positive roots). We can use any positive root system. Let's use the one generated by base root system $t_i - t_{i+1}$:

$$t_1 - t_2,\ t_2 - t_3, \dots,\ t_{n-1} - t_n$$

Then the positive roots are all sums of the form $t_i - t_j$ (where $i < j$). Summing over all of them

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} t_i - t_j$$

we can rearrange as

$$\sum_{i=1}^{n-1} t_i \sum_{j=i+1}^{n} (-t_j) = \sum_{j=1}^{n} (n-j)(t_j) - (j-1)(t_j) = \sum_{j=1}^{n} (n-2j+1)t_j$$

Or even more cocretely, the first sum adds the rows first and the last sum adds the columns first in the following enumeration of roots:

$$\begin{array}{cccccc}
t_1 & -t_2 & & & & \\
t_1 & & -t_3 & & & \\
 & & \ldots & & & \\
t_1 & & & & & -t_n \\
 & & & & & \\
 & t_2 & -t_3 & & & \\
 & t_2 & & -t_4 & & \\
 & & \ldots & & & \\
 & t_2 & & & & -t_n \\
 & & & & & \\
 & & \ldots & & & \\
 & & & & & \\
 & & & t_{n-2} & -t_{n-1} & \\
 & & & t_{n-2} & & -t_n \\
 & & & & & \\
 & & & & t_{n-1} & -t_n
\end{array}$$

and δ is half of this sum.

3.2.5 Weyl Character Formula

We will adapt the notation $\omega_j = \omega^{-1}(j)$, where $1 \leq j \leq n$ and ω is some element of S_n. notation

Since character is a class function, it is sufficient to define it on diagonal entries. The formula below is only well-defined for group elements with distinct eigenvalues, but all entries that don't have distinct eigenvalues are contained in a set of measure zero.

$$\Theta^*(diag(e^{i\theta_1}, e^{i\theta_2}, \ldots, e^{i\theta_n})) = \frac{\sum_{\omega \in S_n} sign(\omega) e^{i(m_{\omega_1}\theta_1 + m_{\omega_2}\theta_2 + \ldots + m_{\omega_n}\theta_n)/2}}{\prod_{j<k}(e^{i(\theta_j - \theta_k)/2} - e^{i(\theta_k - \theta_j)/2})} =$$

$$= \frac{\sum_{\omega \in S_n} sign(\omega) e^{i((m_{\omega_1} - n + 1)\theta_1 + (m_{\omega_2} - n + 3)\theta_2 + \ldots + (m_{\omega_n} + n - 1)\theta_n)/2}}{\prod_{j<k}(1 - e^{-i(\theta_j - \theta_k)})} \qquad (0)$$

The last expression is well-defined because the eigenvalues are distinct and all $m_{\omega_i} - n + 2i - 1$ values are even because $m_{\omega_i} - n + 2i - 1 = 2m'_{\omega_i} + n - 2i + 1 - n + 2i - 1 = 2m'_{\omega_i}$.

3.2.6 n=2 example

The calculation in this section is due to Diana Shelstad. Any diagonal element, u of $\mathbb{U}(2)$ can be decomposed as $u = ds$, where $s \in \mathbb{SU}(2)$ and d is a diagonal element such that $\det(d) = \det(u)$ and $d \notin \mathbb{SU}(2)$ unless $u \in \mathbb{SU}(2)$. So

$$d = \begin{bmatrix} e^{i\lambda_1} & \\ & e^{i\lambda_2} \end{bmatrix}$$

and

$$s = \begin{bmatrix} e^{i\theta} & \\ & e^{-i\theta} \end{bmatrix}$$

The standard result in representation theory of $\mathbb{SU}(2)$ shows that representations of $\mathbb{SU}(2)$ are parametrized by an integer m (ie, the "highest weight"). A trace of an image of representation, π_m, is the sum of all eigenvalues of that image. In other words, it is (the Weyl Character Formula):

$$e^{im\theta} + e^{i(m-1)\theta} + \ldots + e^{-im\theta} = \frac{e^{i(m+1)\theta} + e^{-(m+1)\theta}}{e^{i\theta} - e^{-i\theta}} \qquad (0)$$

And element of the center of $\mathbb{U}(2)$ is of the form

$$\begin{bmatrix} e^{i\psi} & \\ & e^{i\psi} \end{bmatrix}$$

Because a represenation must send $\mathbf{I}$ to 1, the only representations on the center are homomorphisms

$$\begin{bmatrix} e^{i\psi} & \\ & e^{i\psi} \end{bmatrix} \mapsto e^{il\psi}, \tag{0}$$

for some $l \in \mathbb{Z}$. Evaluating Weyl Character Formula at $\theta + \pi$, we get

$$\frac{e^{i(m+1)(\theta+\pi)} + e^{-(m+1)(\theta+\pi)}}{e^{i(\theta+\pi)} - e^{-i(\theta+\pi)}} = e^{i\theta m}\frac{e^{i(m+1)\theta} + e^{-(m+1)\theta}}{e^{i\theta} - e^{-i\theta}} = (-\mathbf{I})^m\frac{e^{i(m+1)\theta} + e^{-(m+1)\theta}}{e^{i\theta} - e^{-i\theta}}$$

Which has to be consistent with

$$\begin{bmatrix} e^{i(\theta+\pi)} & \\ & e^{-i(\theta+\pi)} \end{bmatrix} = \begin{bmatrix} e^{i\theta} & \\ & e^{-i\theta} \end{bmatrix}\begin{bmatrix} e^{i\pi} & \\ & e^{-i\pi} \end{bmatrix} = \begin{bmatrix} e^{i\theta} & \\ & e^{-i\theta} \end{bmatrix}(-\mathbf{I})$$

Since *all* the representations of the center are of the form in (3.2.6), $(-\mathbf{I})^l = \pi_m(-\mathbf{I}) = (-\mathbf{I})^m$. So $m \equiv l(2)$. And representations of $U(2)$ can be parametrized by such m, l pairs.

Using the first general form of the Weyl Character Formula (3.2.5),

$$\pi_{m_1,m_2}\left(\begin{bmatrix} e^{i\theta_1} & \\ & e^{i\theta_2} \end{bmatrix}\right) = \frac{e^{i(m_1\theta_1+m_2\theta_2)/2} - e^{-i(m_1\theta_2+m_2\theta_1)/2}}{e^{i(\theta_1-\theta_2)/2} - e^{-i(\theta_1-\theta_2)/2}} \tag{0}$$

Multiplying numerator and denominator by $e^{-i(\theta_1-\theta_2)/2}$ and plugging $\theta_1 = \theta$ and $\theta_2 = -\theta$, we get:

$$\frac{e^{i(m_1-m_2)\theta/2} - e^{-i(m_1-m_2)\theta/2}}{e^{i\theta} - e^{-i\theta}}$$

In order for it to agree with (3.2.6), we must have $\frac{m_1-m_2}{2} = m+1$. Because $m > 0$, that immediately gives that $m_1 > m_2$ and m_1, m_2 must have the same parity. Plugging $\theta_1 + \pi$ and $\theta_2 + \pi$ into (3.2.6):

$$\frac{e^{i(m_1(\theta_1+\pi)+m_2(\theta_2+\pi))/2} - e^{-i(m_1(\theta_2+\pi)+m_2(\theta_1+\pi))/2}}{e^{i((\theta_1+\pi)-(\theta_2+\pi))/2} - e^{-i((\theta_1+\pi)-(\theta_2+\pi))/2}} =$$

$$= e^{i(\pi\frac{m_1+m_2}{2})} \frac{e^{i(m_1\theta_1+m_2\theta_2)/2} - e^{-i(m_1\theta_2+m_2\theta_1)/2}}{e^{i(\theta_1-\theta_2)/2} - e^{-i(\theta_1-\theta_2)/2}}$$

So $(-\mathbf{I})^l = \pi_{m_1,m_2}(-\mathbf{I}) = (-\mathbf{I})^{\frac{m_1+m_2}{2}}$. Calculating (mod 2), we get $\frac{m_1+m_2}{2} \equiv l \equiv m \equiv \frac{m_1-m_2}{2} - 1 \equiv \frac{m_1+m_2}{2} - (m_2+1)$. Thus m_2 and m_1 are odd.
So $m_1, m_2 \in \mathbb{Z}$ such that $m_1 > m_2$ parametrize irreducible unitary representations of $U(2)$ via (3.2.6).

3.3 Noncompact Forms

By the Theorem of the Highest Weight [Hal04, p 197], every irreducible finite dimensional representation of $\mathbb{GL}(n, \mathbb{C})$ occurs on the compact form $\mathbb{U}(n)$. So the irreducible discrete series representations of $\mathbb{U}(p, q)$ must be infinite dimensional.

3.3.1 Space of Infinite Dimensional Action

To have an infinite dimensional representation, we first must define the "vectors" on which the group would act. Let H be a Hilbert space. And let H^∞ denote all the C^∞ elements of H. H^∞ is dense in H. So H^∞ can be taken to be the linear space on which the representation acts.
[Wal98, p 32] does the usual construction of Lie algebra representation of $\mathfrak{g}$ on H^∞ by taking a derivative at 0 of $\exp(t\cdot)$ action on H^∞. He, further,

shows that $g \mapsto \pi(g)v$ (for $v \in H^\infty$) is a smooth Lie algebra representation. π extends to a representation of universal algebra of $\mathfrak{g}_\mathbb{C}$ on H^∞. Both the representation of $\mathfrak{g}$ and the representation of $U(\mathfrak{g})$ are denoted as (π, H).

3.3.2 Trace Character

A character on the linear operator π is defined as a functional. Let C_c^∞ denote smooth functions with compact support. Let $\mathscr{C}(G)$ be the Schwartz space defined by Harish-Chandra. Of note is the fact that functions of $\mathscr{C}(G)$ are K-bi-invariant and rapidly decreasing.

An operator $\pi(f) : V \to V$, where $f \in C_c^\infty(G)$ is defined by

$$\pi(f)v = \int_G f(x)\pi(x)V dx,$$

where the integration is with respect to a Haar measure. π is an operator of trace class. $f \mapsto \operatorname{tr} \pi(f)$ is a continuous linear functional on $C_c^\infty(G)$. π extends to a continuous linear functional on $\mathscr{C}(G)$. [Wal98, p 313]

Set $\Theta_\pi(f) = \operatorname{tr} \pi(f)$. Then

$$\Theta_\pi(f) = \int_G \Theta_\pi(x) f(x) dx$$

defines a real analytic function Θ_π on regular semisimple elements of G. Θ_π (on G) is called the **trace character** of π.

3.3.3 Infinitesmal Character

Following [Wal98], let $\mathscr{E}_2(G)$ denote the set of *equivalence classes* of *irreducible* square integrable representations of G. [Wal98, thm 7.2.1] (due to Harish-Chandra) states that

Theorem 3.1. *$\mathscr{E}_2(G)$ is non-empty if and only if G has a* <u>*compact*</u> *Cartan subgroup.*

Further, [Wal98, thm 7.7.2] states that for any $\sigma \in \mathscr{E}_2(G)$ there is an irreducible unitary representation of T, μ, such that the infinitesmal character of σ is χ_μ. So the same infinitesmal characters come up for the infinite dimensional case as for the finite dimensional case.
[Wal98, sec 3.2] uses Harish-Chandra's Isomorphism to show that π acts by a constant on the center of $U(\mathfrak{g})$, $Z(\mathfrak{g})$. [Wal98, thm 3.2.4] shows that this infinitesmal character χ, which is determined by $\pi(z)v = \chi(z)v$, is invariant under the action of Weyl group $W_{\mathbb{C}}$.

3.3.4 Weyl Character Formula

First, we define a $(\mathfrak{g}, K)$-module. Following [Wal98, p 80], let G be a *real* Lie group with Lie algebra $\mathfrak{g}$. Let K be a compact subgroup of G. Let V be a $\mathfrak{g}$-module that is also a K-module. Then V is a $(\mathfrak{g}, K)$-module if the following three conditions are satisfied:

(1) $k.(X.v) = Ad[k](X).v$ for all $v \in V, k \in K, X \in \mathfrak{g}$

(2) If $v \in V$ then Kv spans a finite dimensional vector subspace of V,W_v, such that the action of K on W_v is continuous.

(3) If $Y \in \mathfrak{t}$ and if $v \in V$ then $d/dt_{t=0} \exp(tY)v = Yv$

Now to use [Wal98, secs 8.1.2,8.1.4], set $K = \mathbb{U}(p) \times \mathbb{U}(q)$. Then the class function Θ_π must assume a distinct value on each coset of $W_{\mathbb{R}}$-orbit of K-module. And according to Harish-Chandra (as quoted in [She08]) there exists a tempered invariant eigendistribution with infinitesmal character $\chi(m_1, m_2, \ldots, m_n)$ on the matrices T for each such distinct Θ_π.

An element of T is **regular** if all eigenvalues are distinct. Regular element are dense in T and character is a continuous function. So it is sufficient to calculate characters on regular elements. The calculation in [She08]) shows that for a fixed coset $\omega\Omega_{\mathbb{R}}^j$ and fixed infinitesmal character $\chi(m_{\omega_1}, m_{\omega_2}, \ldots, m_{\omega_3})$ the character Θ is

$$\Theta_{\chi,\omega,j} = \frac{(-1)^{pq} \det \omega \sum_{\omega_0 \in \Omega_{\mathbb{R}}^j} \det \omega_0 \Lambda_{\omega\omega_0}}{\prod_{\alpha>0}(1-\alpha^{-1})}, \tag{0}$$

where Λ_ω is the character $\exp(\omega^{-1}\mu - \delta)$ and μ is linear form which is regular dominant for standard ordering of roots.

3.4 L-packet

Equation (3.3.4) shows that to each infinitesmal character $\chi(m_1, m_2, \ldots, m_3)$ determines a set of of irreducible discrete series representations parametrized by real Weyl groups $\Omega_{\mathbb{R}}^j$ and their cosets $\omega\Omega_{\mathbb{R}}^j$. This set of representations is the **L-packet** of the χ.

4 K-group

Fix an infinitesmal character $\chi(m_1, m_2, \ldots, m_n)$. A **$K$-group** will be an algebraic variety over $\mathbb{R}$ comprised of disjoint union of real forms which are inner forms of each other.

Following [She08], denote $G^j = \mathbb{U}(n-j, j)$, where $j \leq m$ and $m \leq \frac{n}{2}$. The reason that only $(n-1,1), (n-2,2), \ldots, (n-m,m)$ pairs are used is that

$\mathbb{U}(n-j) \times \mathbb{U}(j)$ is isomorphic to $\mathbb{U}(j) \times \mathbb{U}(n-j)$. And, in case $m = \frac{n}{2}$, $\mathbb{U}(n-m) \times \mathbb{U}(m)$ is actually equal to $\mathbb{U}(m) \times \mathbb{U}(n-m)$. The G^{n-m} case will be referred to as **quasi-split**.

Determining which K-group a particular representation belongs to will be part of the process of calculating an endoscopic code. If n is odd, then all G^j will be in one K-group called **G**.

$$\mathbf{G} = G^0 \amalg G^1 \amalg \ldots \amalg G^m$$

If n is even, then there will be two K-groups: **G**,**G**'. **G** will be comprised of the quasi-split group and two copies of each G^j with $m-j$ even:

$$\mathbf{G} = G^m \amalg G^{m-2} \amalg G^{m-2} \amalg \ldots$$

G' will be comprised of two copies of each G^j with $m-j$ odd and $j > 0$

$$\mathbf{G}' = G^{m-1} \amalg G^{m-1} \amalg G^{m-3} \amalg G^{m-3} \amalg \ldots$$

K-group, G, contaning the quasi-split group G^{m-n} will be referred to as **quasi-split K-group**.

5 Regular Elliptic Parameter φ For An L-Packet Of Discrete Series Representations

This section follows [She08] in defining Langland regular elliptic parameter but specializes it to unitary groups.

Let

$$\mathbf{J}_n = \begin{bmatrix} & & & & 1 \\ & & & -1 & \\ & & 1 & & \\ & \cdots & & & \\ (-1)^{n+1} & & & & \end{bmatrix}$$

Define conjugation of $\mathbb{GL}(n,\mathbb{C})$, $\boldsymbol{\sigma}$, by $\boldsymbol{\sigma}(g) = \mathbf{J}_n{}^t g^{-1} \mathbf{J}_n^{-1}$.

Conjugation $\boldsymbol{\sigma}$ determines Galois group Γ of $(\mathbb{GL}(n,\mathbb{C})/\mathbb{U}(n-m,m))$ extension to be $\{1,\boldsymbol{\sigma}\}$. Define

$$W = W_{\mathbb{R}} = \{z \times \tau : z \in \mathbb{C}^{\times}, \tau \in \Gamma\},$$

an extension of Γ by $\mathbb{C}^{\times}$ with multiplication defined by $(\epsilon,\boldsymbol{\sigma})(\epsilon,\boldsymbol{\sigma}) = -1$, where ϵ is $2n^{th}$ root of unity.

Then L**-group** is defined as semi-direct product

$${}^L G = \mathbb{GL}(n,\mathbb{C}) \rtimes W,$$

with typical element written as $g \times \tau$. A **regular elliptic parameter** is a $\mathbb{GL}(n,\mathbb{C})$-conjugacy class of the homomorphism $\varphi : W \to {}^L G$:

$$\varphi(\omega) = \varphi_0(\omega) \times \omega,$$

where φ_0 is a continuous map of W into $\mathbb{GL}(n,\mathbb{C})$ such that

$$\varphi_0(z \times 1) = \begin{bmatrix} (z/\overline{z})^{m_1/2} & & & \\ & (z/\overline{z})^{m_2/2} & & \\ & & \cdots & \\ & & & (z/\overline{z})^{m_n/2} \end{bmatrix}$$

and $\varphi_0(1 \times \boldsymbol{\sigma}) = \mathbf{J}_n$.

Proposition in [She08] states that

Proposition 5.1. *Regular elliptic parameters φ are in 1-1 correspodence with tuples $(m_1, m_2, \ldots, m_n)$ of integers, where $m_1 > m_2 > \ldots > m_n$ and each m_i is of parity opposite to n. Given such a tuple, the corresponding parameter*

$$\varphi = \varphi(m_1, m_2, \ldots, m_n)$$

has representative φ given by

$$\varphi(z \times 1) = \begin{bmatrix} (z/\overline{z})^{m_1/2} & & & \\ & (z/\overline{z})^{m_2/2} & & \\ & & \cdots & \\ & & & (z/\overline{z})^{m_n/2} \end{bmatrix} \times (z \times 1), \ z \in \mathbb{C}^\times$$

and

$$\varphi(1 \times \boldsymbol{\sigma}) = \mathbf{J}_n \times (1 \times \boldsymbol{\sigma}).$$

5.1 $1 \to Z_{sc} \to \mathbb{S}^{sc} \to \mathbb{S} \to 1$ extension

Let S be the centralizer in $\mathbb{GL}(n, \mathbb{C})$ of ϕ. Then S consists of elements of order 2 in T. Ie, it is the group of matrices of the form

$$\begin{bmatrix} \pm 1 & & & \\ & \pm 1 & & \\ & & \ddots & \\ & & & \pm 1 \end{bmatrix}$$

Define $\mathbb{S}$ to be the image of S under canonical $p : GL(n,\mathbb{C}) \to PGL(n,\mathbb{C})$ homomorphism. So, in particular, $\mathbb{S}$ contains all matrices of the form

$$\begin{bmatrix} \pm\epsilon & & & \\ & \pm\epsilon & & \\ & & \ddots & \\ & & & \pm\epsilon \end{bmatrix},$$

where ϵ is $2n^{th}$ primitive root of unity. That is, for any $\mathbf{D} \in S$, $\mathbf{D} = \epsilon\mathbf{D}$ in $\mathbb{S}$. $\mathbb{S}$ is isomorphic to $\overset{n-1}{\oplus} \mathbb{Z}_2$.

Define $\mathbb{S}^{sc}$ to be $SL(n,\mathbb{C}) \cap p^{-1}(\mathbb{S})$. Also denote the center of $SL(n,\mathbb{C})$ by Z_{sc}. Z_{sc} consists of matrices of the form

$$\begin{bmatrix} (\epsilon^2)^k & & & \\ & (\epsilon^2)^k & & \\ & & \ddots & \\ & & & (\epsilon^2)^k \end{bmatrix},$$

where k is an integer $0, \ldots, n-1$.

5.1.1 n Odd

If n is odd, then $p\,|_{\mathbb{S}^{sc}} : \mathbb{S}^{sc} \to \mathbb{S}$ is injective because $-\mathbf{I} \notin SL(n,\mathbb{C})$. And since the only elements of Z_{sc} that are in $\mathbb{S}^{sc}$ are $\pm\mathbf{I}$ (which are in the same equivalence class in $\mathbb{S}^{sc}$), the extension $1 \to Z_{sc} \to \mathbb{S}^{sc} \to \mathbb{S} \to 1$ splits.

5.1.2 n Even

If n is even, then $\mathbb{S}^{sc}$ is generated by matrices in S which have an even number of -1's on the diagonal an the matrix

$$\mathbf{E} = \begin{bmatrix} \epsilon & & & & \\ & \epsilon & & & \\ & & \ddots & & \\ & & & \epsilon & \\ & & & & -\epsilon \end{bmatrix},$$

where $\mathbf{E}$ is in $\mathbb{S}^{sc}$ because $\epsilon^n(-1) = 1$ and $p(\mathbf{E})$ is in the equivalence class of

$$\begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & & -1 \end{bmatrix}$$

in $\mathbb{S}$. $1 \to Z_{sc} \to \mathbb{S}^{sc} \to \mathbb{S} \to 1$ does not split because $p(\mathbf{E}^2) = \mathbf{I}$, $p(\mathbf{I}) = \mathbf{I}$ in $\mathbb{S}$ and $\mathbf{E}^2 \neq \mathbf{I}$ in $\mathbb{S}^{sc}$.

5.2 Characters on $\mathbb{S}$

Still following [She08] characters on $\mathbb{S}$ are binary words $\delta_1\delta_2\ldots\delta_n$ where each δ_i is a bit. These are characters are linearly independant on each position of diagonal elements of $\mathbb{S}$, the addition is without carry over. So the group of characters on $\mathbb{S}$ of $GL(n,\mathbb{C})$ can be identified with a subgroup of $\overset{n}{\oplus}\mathbb{Z}_2$ considered as an additive group.

5.2.1 n odd

Consider the group of "even" binary strings with even number of 1-bits. Let it operate on S element $diag(\epsilon_1, \epsilon_2, \ldots, \epsilon_n)$ by

$$\delta_1\delta_2\ldots\delta_n \mapsto \epsilon_1^{\delta_1}\epsilon_2^{\delta_2}\ldots\epsilon_n^{\delta_n}$$

In the case of n odd, $Z_{sc} \cap S$ is $\{\pm\mathbf{I}\}$. So this character is trivial on $Z_{sc} \cap S$. And since the extension splits, it is a characer on $\mathbb{S}^{sc}$. Since the order both $\mathbb{S}^{sc}$ and the group of such characters is 2^{n-1}, these are all the characters on $\mathbb{S}^{sc}$. Because the number of 1's is even in these characters, they will be called the **even characters**.

5.2.2 n even

All the characters of $\mathbb{S}^{sc}$ defined the same way as the ones for the odd n are well-defined for even n. But additional 2^{n-1} characters on $\mathbb{S}^{sc}$ can be defined by adding $011\ldots1$ to the even characters. These **odd characters** are distinct from the even characters on $\mathbf{E}$ because they happen to evaluate to an odd power of ϵ. This brings the total number of characters on $\mathbb{S}^{sc}$ to 2^n. Since this is the cardinality of $\mathbb{S}^{sc}$, these must be all the characters.

5.3 Characters on $\mathbb{S}^{sc}$ and L-packets

Since an L-packet of a fixed representation χ is determined by j and coset $\omega\Omega_{\mathbb{R}}^{j}$, if it were to be shown that there is a correspondence between characters of $\mathbb{S}^{sc}$ and the data consisting of j,$\omega\Omega_{\mathbb{R}}^{j}$ and an ω-orbit of a diagonal matrix (considered as a tuple) $diag(m_1, m_2, \ldots, m_n)$, then the characters of $\mathbb{S}^{sc}$ would determine a particular discrete series representations of $\mathbb{GL}(n, \mathbb{C})$.

6 Classification Of Representations Of $\mathbb{U}(n-j, j)$ By Endoscopic Codes

We now come to the actual purpose of this thesis: following [She08] to classify discrete series representations with a particular infinitesmal character $\chi(m_1, \ldots, m_n)$ by binary codes.

6.1 Notation

First, we note that for a fixed n the multiplicative group of diagonal real unitary $n \times n$ matrices (ie, S) is *group*-isomorphic to the additive group $\bigoplus_{n \text{ copies}} \mathbb{Z}_2$.

$$S = \left\{ \underbrace{\begin{bmatrix} \pm1 & & & & \\ & \pm1 & & & \\ & & \ddots & & \\ & & & \pm1 & \\ & & & & \pm1 \end{bmatrix}}_{n} \right\}$$

The isomorphism identifies 1's on the digonals with 0's in the binary strings and -1's on the diagonals with 1's in the binary strings. We'll call this isomorphism β. This is a slight abuse of notation because β is really a covariant functor between two isomorphic categories of groups. We'll use β though as if it were $\beta : S \to \bigoplus_{n \text{ copies}} \mathbb{Z}_2$ rather than $\beta : S \mapsto \bigoplus_{n \text{ copies}} \mathbb{Z}_2$. To further facilitate notation, we'll adopt the practice of labeling anything in the image of β by a letter "b" in the *upper left* corner of whatever object or notation morphism β operates on. For example, if

$$\mathbf{I_{3,2}} = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & -1 & \\ & & & & -1 \end{bmatrix}$$

and ω is the permutation matrix that corresponds to reflection (34), then in this notation

$$\beta(\mathbf{I_{3,2}}) = {}^bI_{3,2} = 00011$$

$$\beta(\omega) = {}^b\omega$$

$$\beta(\omega.\mathbf{I_{3,2}}) = \beta(\omega).\beta(\mathbf{I_{3,2}}) = {}^b\omega.\beta(\mathbf{I_{3,2}}) = \beta(\omega).{}^bI_{3,2} = {}^b\omega.{}^bI_{3,2} = 00101$$

The operation of addition of binary strings without carry over (ie, the addition in $\bigoplus_{n \text{ copies}} \mathbb{Z}_2$) has a well-accepted name in computer science. The name is "exclusive or". We'll take advantage of this and use the symbol " xor " for this abelian group addition operation. For example, $0110 \text{ xor } 1100 = 1010$. The following should be obvious:

Lemma 6.1. *Let τ act on binary strings by some permutation of bits. Then $\tau.(x \text{ xor } y) = \tau.x \text{ xor } \tau.y$.*

6.2 Cosets Of The Weyl Group $(\Omega/\,\Omega_{\mathbb{R}}^{j})$

An element of $\Omega/\Omega_{\mathbb{R}}^{j}$ has a well-defined action on the matrix $\mathbf{I_{n-j,j}}$. That is because permutting 1's or -1's among themselves does not change $\mathbf{I_{n-j,j}}$, the actual effect of such an element is to exchange r "1"'s with r "-1"'s, where $r \le j \le n-j$. In the binary-string view, this is the equivalent of interchanging r 0's with r 1's in the string $\underbrace{0\ldots0}_{n-j}\underbrace{1\ldots1}_{j}$.

Theorem 6.2. *Let $\omega\Omega^j_{\mathbb{R}} \in \Omega/\Omega^j_{\mathbb{R}}$. Let ${}^bI' = \beta(\omega.\mathbf{I_{n-j,j}})$. Let $\{u_1,\ldots,u_r\}$ be positions of 1's in the first $n-j$ fields of ${}^bI'$ and $\{v_1,\ldots,v_t\}$ be positions of 0's in the last j fields of ${}^bI'$. Let s be a binary string of length n which has 1's in positions $\{u_1,\ldots,u_r,v_1,\ldots,v_t\}$ and 0's in all other positions. Then*

(a) $r = t$,

(b) $\Omega^j_{\mathbb{R}}$ *action does not change* $\mathbf{I_{n-j,j}}$ *(so* $\omega\Omega^j_{\mathbb{R}}.\mathbf{I_{n-j,j}} = \omega.\mathbf{I_{n-j,j}}$*),*

(c) $\beta(\omega.\mathbf{I_{n-j,j}}) = {}^bI' = s \text{ xor } {}^bI_{n-j,j}$. *In other words,*

$$\begin{array}{ccc}
\mathbf{I_{n-j,j}} & \xrightarrow{\omega\,\Omega^j_{\mathbb{R}}} & \omega\,\Omega^j_{\mathbb{R}}\,.\mathbf{I_{n-j,j}} = {}^bI'_{n-j,j} \\
{\scriptstyle\beta}\big\downarrow & & \big\downarrow{\scriptstyle\beta} \\
{}^bI_{n-j,j} & \xrightarrow[s \text{ xor } \cdot]{} & s \text{ xor } {}^bI_{n-j,j}
\end{array}$$

commutes.

Proof. (a) Any permutation of bits leaves the total number of 0's and the total number of 1's fixed. ${}^bI_{n-j,j}$ has $n-j$ 0's. So ${}^bI'$ must have $n-j$ 0's as well. Since ${}^bI'$ has r 1's in the first $n-j$ positions, it must have $n-j-r$ 0's in the first $n-j$ positions. Which means it must have $n-j-(n-j-r) = r$ 0's in the remaining j positions. And t is precisely the number of 0's in the last j positions. So $r = t$.

(b)

$$\left[\begin{array}{c|c} S_{n-j} & \\ \hline & S_j \end{array}\right]\left[\begin{array}{c|c} I_{n-j} & \\ \hline & -I_j \end{array}\right]\left[\begin{array}{c|c} S_{n-j} & \\ \hline & S_j \end{array}\right]^{-1} = \left[\begin{array}{c|c} I_{n-j} & \\ \hline & -I_j \end{array}\right]$$

(c) By hypothesis, all u_i's are distinct. So are all v_i's. Further, $u_i < v_k$ for all i, k. Taking xor with any binary string, s', is equivalent to reversing all bits in the positions where s' has 1's. This is the "masking" operation in computer science. So taking xor with s will reverse all bits in positions $\{u_1, \ldots, u_r, v_1, \ldots, v_r\}$ and leave all other bits fixed. Since $1 \le u_i \le n - j < v_k \le n$ and ${}^bI_{n-j,j}$ has 0's in the first $n - j$ positions and 1's in the remaining j positions, $({}^bI_{n-j,j} \text{ xor } s)$ must have $1's$ in $\{u_1, \ldots, u_r\}$ positions and 0's in $\{v_1, \ldots, v_r\}$ positions. Which makes ${}^bI'_{n-j,j} = ({}^bI_{n-j,j} \text{ xor } s)$ by definition of u_i's and v_k's.

$\square$

Let u_i's and v_k'k be as in theorem 6.2. Now define

$$s_i \stackrel{\text{def}}{=} 000\ldots\underset{\substack{\uparrow \\ \text{position } u_i}}{0000100}\ldots\underset{\substack{\uparrow \\ \text{position } v_i}}{0000100}\ldots000$$

By induction, $s = s_1 \text{ xor } s_2 \text{ xor } \ldots \text{ xor } s_i \text{ xor } \ldots \text{ xor } s_r$.

Lemma 6.3.

$$s = s_1 \text{ xor } s_2 \text{ xor } \ldots \text{ xor } s_i \text{ xor } \ldots \text{ xor } s_r$$

is well-definied even though any particular s_i depends on a particular choice among all of the choices of permutations of $\{u_1, \ldots, u_r\}$ and all of the choices of permutations of $\{v_1, \ldots, v_r\}$.

Proof. This follows immediately from the fact that all u_i and v_j are distinct from each other. $\square$

Corollary 6.4. *Each coset $\omega\Omega^j_{\mathbb{R}}$ in the set $\Omega/\Omega^j_{\mathbb{R}}$ has a coset representative which can be written as a product of* commuting *reflections (ie, 2-element permutations)*

Proof. Fix j and $\omega\Omega^j_{\mathbb{R}}$. This fixes the sets $\{u_1, \ldots, u_r\}$ and $\{v_1, \ldots, v_r\}$ (as they are defined in theorem 6.2). Define τ_i as the reflection that permutes positions u_i and v_i. Then $\beta(\tau_i.\mathbf{I_{n-j,j}}) = s_i \text{ xor } {}^bI_{n-j,j}$. Because u_i's and v_k's are all distinct, τ_i's all commute with each other. So

$$\beta(\tau_r\tau_{r-1}\ldots\tau_1.\mathbf{I_{n-j,j}}) = \beta(\tau_r\tau_{r-1}\ldots\tau_2).\beta(\tau_1.\mathbf{I_{n-j,j}}) = \tag{0}$$

$$\beta(\tau_r\tau_{r-1}\ldots\tau_2).(s_1 \text{ xor } {}^bI_{n-j,j}) = \beta(\tau_r\ldots\tau_2).s_1 \text{ xor } \beta(\tau_r\ldots\tau_2).{}^bI_{n-j,j} \tag{0}$$

The equality in (6.2) follows from lemma 6.1. But ${}^b\tau_i$ has no effect on s_1 if $i \neq 1$. So

$$\beta(\tau_r\ldots\tau_2).s_1 = s_1 \tag{0}$$

Plugging (6.2) into (6.2) gives $s_1 \text{ xor } \beta(\tau_r\tau_{r-1}\ldots\tau_2).{}^bI_{n-j,j}$. Proceeding by induction, this is equal to $(s_1 \text{ xor } s_2\ldots \text{ xor } s_r) \text{ xor } {}^bI_{n-j,j} = s \text{ xor } {}^bI_{n-j,j}$. Which by theorem 6.2(c) is equal to ${}^bI'$. Applying β^{-1} to both sides, we get that $\tau_r\tau_{r-1}\ldots\tau_1$ acts the same way on $\mathbf{I_{n-j,j}}$ as $\omega\Omega^j_{\mathbb{R}}$. So $\tau_r\tau_{r-1}\ldots\tau_1 \in \omega\Omega^j_{\mathbb{R}}$ is the desired product of commuting reflections. $\square$

Corollary 6.5. *Coset represantative whose existence is guaranteed by corollary 6.4 can be picked to be an involution.*

Proof. Immediate from corollary 6.4. $\square$

Since all characters on $\mathbb{S}^{sc}$ are quadratic, this establishes that each endoscopic code does correspond to a character of $\mathbb{S}^{sc}$ and thus there is a correspondence between $j, \omega\Omega^j_{\mathbb{R}}$ data and an L-packet of a particular representation χ.

7 Calculation Of The Codes

7.1 Shuffle Product

As it turns out a number of calculations can be expressed much more succinctly using the "shuffle product". It will be denoted by ϕ from now on. We define it as a permutation of a finite sequence best defned by its inverse:

$$\phi^{-1}(i) = \begin{cases} 2(i-1)+1 & 1 \leq i \leq n-m, \\ 2(n-i+1) & n-m+1 \leq i \leq n \end{cases}$$

ϕ^{-1} has the following effect on the **sequence** $\{1, n, 2, n-1, \ldots, n-m+1\}$:

$$\{1, n, 2, n-1, 3, \ldots, n-m+1\} \xrightarrow{\phi^{-1}} \{1, 2, 3, \ldots, n\}$$

Giving ϕ the following effect on the **sequence** $\{1, 2, 3, \ldots, n\}$:

$$\{1, 2, 3, \ldots, n\} \xrightarrow{\phi} \{1, n, 2, n-1, 3, \ldots, n-m+1\}$$

We will also abuse notation and allow ϕ operate on binary strings in the same manner as it operates on sequences. That is if $b' = \phi(b)$, then bit $\phi(i)$ of b' will be equal to bit i of b. For example,

$$\phi(\underbrace{00\ldots00}_{m+1}\underbrace{11\ldots11}_{m}) = 01010101\ldots1010,$$
$$\phi(\underbrace{00\ldots00}_{m}\underbrace{11\ldots11}_{m}) = 01010101\ldots101$$

If we treat each binary string as a function $b : \{1, \ldots, n\} \to \{0, 1\}$, that amounts to saying that $b(\phi(i)) = (\phi(b))(i)$.

7.2 Chamber

For the χ that results from φ-parameter acted on by $\omega \in \Omega$, the representation is determined by the ineqaulity $m_{\omega_1} > m_{\omega_2} > \ldots > m_{\omega_n}$. And the representation does not change if we pick another in $\omega' \in \omega\Omega_{\mathbb{R}}^{j}$. So this representation does not change if it is defined by $m_{\omega'_1} > m_{\omega'_2} > \ldots > m_{\omega'_n}$. For example (in terms of standard positive root system)

$$m_1 > m_2 > m_3 > m_4 > m_5$$

and

$$m_3 > m_1 > m_2 > m_5 > m_4$$

define the same representation of $\mathbb{U}(3,2)$. That is, rather than the inequality itself, the represenatation is determined by the Weyl chamber of the roots under the real Weyl group, $\Omega_{\mathbb{R}}^{j}$.

Instead of using the standard simple positive root system, we'll use a positive root system that is non-compact for quasi-split G^m. More specifically, we pick the system of non-compact simple positive roots

$$t_{\phi(1)} - t_{\phi(2)}, t_{\phi(2)} - t_{\phi(3)}, \ldots, t_{\phi(n-2)} - t_{\phi(n-1)}, t_{\phi(n-1)} - t_{\phi(n)}$$

For example, $t_1 - t_3, t_3 - t_2$ for $n = 3$ and $t_1 - t_4, t_4 - t_2, t_2 - t_3$ for $n = 4$. In terms of these roots, the chamber of the $\Omega_{\mathbb{R}}^{m}$ (ie, the orbit of identity element in $\Omega/\Omega_{\mathbb{R}}^{m}$) is

$$m_1 > m_n > m_2 > m_{n-1} > \ldots > m_m > m_{n-m+1} = m_{m+1} \qquad \text{if } n \textbf{ is even},$$

$$m_1 > m_n > m_2 > m_{n-1} > \ldots > m_{m+1} > m_{n-m+1} = m_{m+2} \qquad \text{if } n \textbf{ is odd}.$$

This special chamber, to be denoted $\mathcal{C}_*$, can be more clearly rewritten in terms of the shuffle product as

$$m_{\phi(1)} > m_{\phi(2)} > m_{\phi(3)} > \ldots > m_{\phi(n)}$$

An element ω of Ω acting on the roots can be considered as element ω^{-1} acting on the chamber. Together with corollary 6.4 this implies that the action of $\omega^{-1}\Omega^j_{\mathbb{R}}$ on a chamber $\mathcal{C}$ is equivalent to the action of $\omega\Omega^j_{\mathbb{R}}$ on $\phi(\mathbf{I}_{n-j,j})$ (and consequently to the action of ${}^b\omega\Omega^j_{\mathbb{R}}$ on $\phi({}^bI_{n-j,j})$), where $\mathcal{C}$ is a chamber of G^j. Denoting the 1's by the black balls and 0's by the white balls, it's the ω^{-1} permutation of the balls in figure 7.2. Using the $\omega_j = \omega^{-1}(j)$ notation

Figure 2: Standard chamber of non-compact roots $\mathcal{C}_*$

from section 3.2.5, this is the chamber

$$m_{\omega_{\phi(1)}} > m_{\omega_{\phi(2)}} > m_{\omega_{\phi(3)}} > \ldots > m_{\omega_{\phi(n)}}$$

For example, for $n = 5$, the special chamber $\mathcal{C}_*$ is (the elements corresponding to -1's in $\mathbf{I}_{n-m,m}$ are boxed for clarity)

$$m_1 > \boxed{m_5} > m_2 > \boxed{m_4} > m_3$$

the chamber of identity element of $\Omega/\Omega^1_{\mathbb{R}}$ is

$$m_1 > \boxed{m_5} > m_2 > m_4 > m_3$$

while the chamber of the ω that corresponds to the reflection that exchanges elements 3 and 5 is

$$m_1 > m_3 > m_2 > m_4 > \boxed{m_5}$$

Now extend the ${}^b\boxed{\cdot}$ notation to chambers by denoting the 1's in ${}^bI_{n-j,j}$ of G^j by 0 in the binary string and -1's in ${}^bI_{n-j,j}$ of G^j by 1's in the binary string. So that for a chamber $\mathcal{C}$ of G^j, ${}^b\mathcal{C}_* = \phi({}^bI_{n-j,j})$ and

$$\omega(\mathcal{C}_*) = \phi(\omega(\mathbf{I}_{n-j,j})) \tag{0}$$

and consequently

$$^{b}(\omega(\mathcal{C}_*)) = \phi(^{b}\omega(^{b}I_{n-j,j})). \tag{0}$$

The order of applying shuffle and permutation have to be $\phi \circ \omega$ for indices of m_i's of a chamber and $\omega \circ \phi$ for binary strings because they are applied to the indices when operating on m_i's and to the position in the string when operating on binary strings.

7.2.1 $^{b}\delta$

We define $^{b}\delta$ as the binary string associated to a given chamber $\mathcal{C}$ such that if $\mathcal{C} = \omega\Omega^{j}_{\mathbb{R}}(\mathcal{C}_*)$, then

$$^{b}\delta \ \text{ xor } \ ^{b}I_{n-j,j} = {}^{b}\omega(^{b}I_{n-j,j}).$$

Lemma 7.1. $^{b}\delta = \phi^{-1}(^{b}\mathcal{C}) \ \text{ xor } \ ^{b}I_{n-j,j}$

Proof. Because xor addition with a particular string is an involution,

$$^{b}\delta = {}^{b}I_{n-j,j} \ \text{ xor } \ ^{b}\omega(^{b}I_{n-j,j}). \tag{0}$$

$\mathcal{C} = \omega\mathcal{C}_*$ by definition of $^{b}\delta$. So $^{b}\mathcal{C} = {}^{b}(\omega\mathcal{C}_*)$. $^{b}(\omega\mathcal{C}_*) = \phi(^{b}\omega(^{b}I_{n-j,j}))$ by (7.2). Applying ϕ^{-1} to both sides and plugging into (7.2.1), we get

$$^{b}\delta = {}^{b}I_{n-j,j} \ \text{ xor } \ ^{b}\omega(^{b}I_{n-j,j}) = {}^{b}I_{n-j,j} \ \text{ xor } \ \phi^{-1}(^{b}(\omega\mathcal{C}_*)) = {}^{b}I_{n-j,j} \ \text{ xor } \ \phi^{-1}(^{b}\mathcal{C})$$

$\square$

In the following examples the boxed m_i's correspond to -1's in $\mathbf{I}_{n-j,j}$. So it should be clear which G^j each chamber belongs to (j will be equal to the total number of boxed elements).

$\mathcal{C}$	${}^b\mathcal{C}$	$\phi^{-1}({}^b\mathcal{C})$	${}^bI_{n-j,j}$	${}^b\delta$	ω
$m_1 > \boxed{m_3} > m_2$	010	001	001	000	id
$\boxed{m_3} > m_1 > m_2$	100	100	001	101	(13)
$m_1 > m_3 > m_2 > m_4 > \boxed{m_5}$	00001	00100	00001	00101	(35)
$m_1 > \boxed{m_4} > \boxed{m_3} > m_2$	0110	0101	0011	0110	(23)
$\boxed{m_5} > m_2 > \boxed{m_6} > m_1 > m_3 > m_4$	101000	110000	000011	110011	(26)(15)

7.3 Codes Attached to K-group G

For the quasi-split K-group the codes are produced by twisting ${}^b\delta$ through addition of a binary string ${}^bt$ which is uniquely determined by j. $\mathbf{t}$ is defined as the unique element of $T \cap SL(n,\mathbb{C})$ whose conjugation action on $T(\mathbb{R})$ is equal the action of $\sigma^m\sigma^j$. For a fixed m (and, therefore, fixed n), this t depends only on the choice of j.

To calculate t, we first calculate $\mathbf{t}' = \mathbf{I}_{n-m,m}\mathbf{I}_{n-j,j}$. And then calculate $\mathbf{t}$ as the element of $\{\mathbf{t}', -\mathbf{t}'\}$ which is in $SL(n,\mathbb{C})$. This is justified because conjugaton by $-\mathbf{I}$ does not change anything. ${}^bt$ is calculated as ${}^bt = \beta(\mathbf{t})$.

An equivalent method of computation of ${}^bt$ is to, first, set ${}^bt' = {}^bI_{n-m,m}$ xor ${}^bI_{n-j,j}$; and then set ${}^bt$ to either ${}^bt'$ or $(111\ldots1$ xor ${}^bt')$ depending on which one of them has an even number of bits.

7.3.1 n Odd

For n odd the endoscopic code attached to a particular chamber is $\phi({}^bt \text{ xor } {}^b\delta)$. This addition of ${}^bt$ is the twist that makes the codes of a particular inequality

$$m_{i_1} > m_{i_2} > \ldots > m_{i_n}$$

(which is used to define a chamber) unique across all G^j.

7.3.2 n Even

In case of n even, the quasi-split case is calculated the same. But the other G^j have 2 codes attached to each chamber. The first one is calculated the same way as for the n odd case. And the second one is calculated by taking xor with $111\ldots1$ (or equivalently by reversing all bits) of the first one.

7.4 Codes Attached to K-group G'

As described in section 4, G' will contain 2 copies of each of $G^{m-1}, G^{m-3}, \ldots$ and it is only defined for even n.

To calculate the code attached to the first copy, we need a ${}^b\delta$ and a twist ${}^bt$. Because, after all, the codes are used to enumerate the ${}^b\delta$'s by ditinguishing them from each other by a twist.

${}^b\delta$ is calculated in the same way as it is for the groups in the K-group G: via the result of lemma 7.1. ${}^bt$ is calculated by adding an arbitrary odd character (we pick $011\ldots11$) to the twist

$$
{}^bt' = \beta(\mathbf{t}'),
$$

where $\mathbf{t}'$ is the element of $T \cap SL(n, \mathbb{C})$ whose conjugation action is equal to the action of $\sigma^{m-1}\sigma^j$. $\mathbf{t}' = \mathbf{I}_{n-m+1,m-1}\mathbf{I}_{n-j,j}$.

Because $m - j$ is odd for all G^j in K-group G', $\mathbf{t}'$ will have an even number of -1's. So ${}^bt'$ is an even code and ${}^bt = ({}^bt' \text{ xor } 011\ldots11)$ is an odd code. Thus the actual code is $\phi({}^b\delta \text{ xor } {}^bt)$.

The code attached to the second copy of a particular chamber is $111\ldots1$ "xor"'ed with the first copy. It is still an odd character (because n is even). Equivalently, it can be calculated as $\phi(100\ldots0 \text{ xor } {}^b\delta \text{ xor } {}^bt')$ because

$$011\ldots1 \text{ xor } 111\ldots1 = 100\ldots0.$$

8 Application: Characters Of $\mathbb{U}(n-1,1)$ for odd n

Lemma 8.1. *To each code c associated to $G^1(\mathbb{R})$ there corresponds a unique binary string s with exactly one 1-bit (and consequenctly diagonal matrix $\beta^{-1}(s)$ with exactly one -1) such that for all other codes $\tilde{c}$ associated to $G^1(\mathbb{R})$ it is true that $\pi_{\tilde{c}}(\beta^{-1}(s)) \neq \pi_c(\beta^{-1}(s))$.*

Proof. ${}^bt$ will denote the twist for $j = 1$. Given a coroot, the representation associated to it is π_c, where $c = \phi^{-1}({}^b\delta \text{ xor } {}^bt)$. ϕ^{-1} is an isomorphism with respect to the addition xor (because ϕ is just a permutation of bits). So $\phi^{-1}({}^b\delta \text{ xor } {}^bt) = \phi^{-1}({}^b\delta) \text{ xor } \phi^{-1}({}^bt)$. Because $\pi_{c \text{ xor } r} = \pi_c\pi_r$ and because $\phi^{-1}({}^bt)$ is a constant for all elements of $\mathbb{U}(n-1,1)$, we only need to consider $\pi_{\phi^{-1}({}^b\delta)}$. Also (again because ϕ is just a permutation of bits), it is true that $\pi_c(s') = \pi_{\phi(c)}(\phi(s'))$. So we can consider just $\pi_{{}^b\delta}(\phi(s'))$. Further, because the statement is that *some* string c exists, we can ignore the isomorphism ϕ

and only consider $\pi_{^b\delta}(s')$. The following is the list of all ${}^b\delta$'s for $\mathbb{U}(n-1,1)$:

$$00\ldots0000$$

$$00\ldots0011$$

$$00\ldots0101$$

$$00\ldots1001$$

$$\ldots$$

$$01\ldots0001$$

$$10\ldots0001$$

We can now prove the existence part of the lemma by stating which string corresponds to which ${}^b\delta$:

${}^b\delta$	s'
$00\ldots0000$	$00\ldots0001$
$00\ldots0011$	$00\ldots0010$
$00\ldots0101$	$00\ldots0100$
$\ldots$	
$01\ldots0001$	$01\ldots0000$
$10\ldots0001$	$10\ldots0000$

The uniqueness comes from the fact that all the strings with exactly one 1-bit have been used. □

9 Recovering Chambers From Codes

The point that has not been addressed yet is why the endoscopic twists make the codes unique (up to "1"'s complement) across all G^j for a given n. This uniqueness will follow from the results of this section. In this section, we will calculate the Harish-Chandra data associated to a particular endoscopic code. That is the sequence $\{i_1, i_2, \ldots, i_n\}$ and j such that the chamber

$$m_{i_1} > m_{i_2} > \ldots > m_{i_n}$$

will correspond to the given code in G^j.

In a case of n odd, there will be $\sum_{j=0}^{\frac{n-1}{2}} \begin{pmatrix} n \\ j \end{pmatrix} = 2^{n-1}$ such codes. In a case of n even, there will be $\sum_{j=0}^{n} \begin{pmatrix} n \\ j \end{pmatrix} = 2^n$ such codes. Thus the number of codes will correspond to the number of orbits of $\Omega/\Omega^j_{\mathbb{R}}$'s (with appropriate orbits' elements counted twice). Establishing that the number of codes corresponds to the number of chambers will show that the endoscopic twists do, indeed, produce distinct codes for all G^j: if two different twists produced the same code, then the total number of codes would end up being less than the total number of chambers.

In order to produce the inverses, it will be helpful to consider codes in terms of standard basis rather than non-compact basis (ie, the de-shuffled version of each code). So that the code for ${}^bI_{3,2}$ would be 00011 rather than 01010. So the first step in calculating the inverse is to take ϕ^{-1}.

Then each code will be considered as composed of 3 parts in such a way that when ${}^bt$ is broken into 3 parts in the same way as the code, each part of ${}^bt$ will have only 1's or only 0's in it. For example, for

$${}^bt = \boxed{00000}\boxed{11}\boxed{00},$$

the code will be broken into parts composed of the first 5 digits, the following

2 and the last 2. While for

$$ {}^{b}t = \boxed{00000}\boxed{}\boxed{000}, $$

the code will be broken into parts composed of the first 5 digits, the following 0 and the last 3 (this is a code attached to $G^{3,1}$ in the nonquasisplit K-group of $\mathbb{GL}(8, \mathbb{C})$.

The three parts of the code will be refered to as **box 1**, **box 2** and **box 3** from now on. They codes will be considered as subdivided into 3 boxes even if box 2 or box 3 is empty. The inverses will be calculated by first examining the invariants of the codes.

9.1 n Odd

For a fixed j, boxes 1,2,3 will have $m+1$,$m-j$,j elements respectively. Let r be as in theorem 6.2. So r is the number of 0's in box 3 before the twist is applied. Before the twist, the total number of 1's in boxes 1 and 2 has to be r as well. Let k denote the number of these 1's in box 2. So there will be $r-k$ 1's in box 1.

Assume, for now, that $m-j$ is such that the twisting element ${}^{b}t$ will have 1's in box 2 and 0's in boxes 1 and 3. This means that $m-j$ is even.

After the twist is applied, the number of 0's and 1's will be as following:

	box 1	box 2	box 3
$1's$	$r-k$	$m-j-k$	r
$0's$	$m+1-r+k$	k	$j-r$

While trying to recover the Harish-Chandra data from the code, the value of j is not yet known. But m is known and so is the total number of 0's and

1's in boxes 2 and 3. The total number of 0's in boxes 2 and 3 is $j+k-r$ while the number of 0's in box 1 is $m+1-(r-k)$. And

$$j+k-r < m+1-(r-k)$$

because $j \leq m$.

Now let $m-j$ be odd. The twisting element can be considered as an xor of $11\ldots11$ with an element which has 1's in box 2 and 0's in boxes 1 and 3. So in effect the only change that needs to be made to the table above is that the rows of 0's and 1's need to be interchanged:

	box 1	box 2	box 3
$1's$	$m+1-r+k$	k	$j-r$
$0's$	$r-k$	$m-j-k$	r

In this case, the total number of 0's in boxes 2 and 3 is $m-j-k+r$ while the total number of 0's in box 1 is $r-k$. And

$$r-k < m-j-k+r$$

because $m-j$ is odd (in particular $m-j>0$).

Thus comparing the number of 0's in box 1 with the combined number of 0's in boxes 2 and 3 will reveal whether we are dealing with $m-j$ odd or even case without knowing a priori what the value of j is.

Depending on which of the two cases is applicable, we can retrieve $r-k$ as the count of 0's or 1's in box 1. Also we can retrieve $j-r+k$ as the combined count (of 0's or 1's) of boxes 2 and 3. This allows to retrieve j as $j=j-r+k+(r-k)$. Once j is known, ${}^{b}t$ can be calculated and xor 'ed with the code to retrieve the cocycle of real Weyl group. And that's enough to calculate the chamber.

CodeToChamberOdd Mathematica^{TM} function in the appendix A performs this calculation. The general entry point for calculating of Harish-Chandra data CodeToHCData can also be used because it distiguishes between n even and n odd.

9.2 n Even

First step to determine the Harish-Chandra data in this case is to identify which K-group the code is associated to. If the binary word is even, it's the quasi-split one and if the binary word is odd, it's the non-quasi-split one. Just as in the case of n odd, we'll use an invariant to check whether $11\ldots11$ string was applied to the code. Unlike the case of n odd, this will result in two codes producing the same Harish-Chandra data, but that is as expected.

9.2.1 Quasi-split Case ($m-j$ Even)

For a fixed j, boxes 1,2,3 will have m,$m-j$,j elements respectively. Again, let r be as in theorem 6.2. Just as in the case of n odd, let k be the number of 1's in box 2 before any twisting binary codes are applied. So the string corresponding to the cocycle will have $r-k$ 1's in box 1, k 1's in box 2 and r 1's in box 3. Assuming the $11\ldots11$ string was not applied, the string corresponding to the code will have 1's and 0's as following:

	box 1	box 2	box 3
$1's$	$r-k$	$m-j-k$	r
$0's$	$m-r+k$	k	$j-r$

The second code for the same data will have the 1's and 0's lines exchanged:

	box 1	box 2	box 3
$1's$	$m-r+k$	k	$j-r$
$0's$	$r-k$	$m-j-k$	r

We don't know a priori which of the two codes is given. Relabeling $r-k$ as x, we do, however, know that in the case of the *second code*:

	box 1	total of box 2 and box 3
$1's$	$m-x$	$j-x$
$0's$	x	$m-j+x$

and $m > j$ (because there is only 1 copy of each code for G^m). Which allows for the invariant

$$(m-x)+(m-j+x) = 2m-j > j = x+(j-x).$$

Whereas in the case of the *first code*:

	box 1	total of box 2 and box 3
$1's$	x	$m-j+x$
$0's$	$m-x$	$j-x$

And

$$x+(j-x) = j \leq 2m-j = (m-x)+(m-j+x).$$

Which allows to distiguish between first and second code by comparing the sum of the number of 1's in box 1 with the combined sum of the number of 0's in boxes 2 and 3 to the sum of the number of 0's in box 1 with the combined sum of the number of 1's in boxes 2 and 3.

Once it is clear which of the codes it is, the $111\ldots11$ can be unmasked from the code. Thus guaranteeing that we are dealing with the first code. Then j can be retrieved as $m-((m-j+x)-x)$. Ie, as m minus the difference in the number of 1's in the combined boxes 2,3 with the number of ones in box 1.

Once j is known the twisting binary string ${}^{b}t$ can be calculated and unmasked from the code. The remaining string is the cocycle binary string and it can be used to calculate the chamber.

CodeToChamberEvenQuasiSplit Mathematica™ function in appendix A calculates Harish-Chandra data for the codes for which n is even and $m-j$ is even. As with all codes, the generic entry point function CodeToHCData can be used as well.

9.2.2 Non-quasi-split Case ($m-j$ Odd)

In this case, for a fixed j, the boxes 1,2,3 will have $m+1$,$m-1-j$,j elements respectively.

The first step is to unmask the $011\ldots111$ string that was xor 'ed to the code to make it an odd character.

Let r, k and x be as in the previous section. Then the counts of 1's and 0's are as following for the *first code*:

	box 1	total of box 2 and box 3
$1's$	x	$m-1-j+x$
$0's$	$m+1-x$	$j-x$

,

and as following for the *second code*:

	box 1	total of box 2 and box 3
$1's$	$m+1-x$	$j-x$
$0's$	x	$m-1-j+x$

.

The invariant

$$(m+1-x)+(m-1-j+x)=2m-j>j=x+(j-x)$$

is more straight forward in this case because $j<m$ implies that $2m-j$ is strictly greater than j. So the count of 1's on the left plus the 0's on the right is eiher strictly greater or strictly lesser than the count 0's on the left plus the count of 1's on the right. Which allows to easily identify whether this is the first or the second code and unmask $111\ldots111$ if necessery.
Then j can be calculated as $x+(j-x)$. Which allows to calcualte the binary string for ${}^{b}t$ and unmask it from the remaining code string to retrieve the binary string corresponding to the cocycle of the real Weyl group's coset. This allows to calculate the chamber.
CodeToChamberEvenNonQuasiSplit Mathematica$^{\mathrm{TM}}$ function in appendix A calculates Harish-Chandra data for the codes for which n is even and $m-j$ is even. As with all codes, the generic entry point function CodeToHCData can be used as well.

A Appendix: Mathematica$^{\text{TM}}$ Module For Computing Endoscopic Codes Over Reals

Below is the Mathematica$^{\text{TM}}$ module tested on version 5.1 of Mathematica. Function PrintCodesForDimension of the module will be used in Appendix B to calculate a few low-dimensional codes. Function EndoscopicCode can be used to calculate codes for a given module (specified as a list of indecis of m_i's) and a given j.

```
(*----cut----CodesFunctions Package Begin----cut----*)
(*
 *  Author: Dmitry Rubanovich
 *  Copyright 2009 by Dmitry Rubanovich
 *
 *  The copyright and the rest of this notice pertains to the
 *  content between the "cut" lines.
 *
 *  This module is part of a doctorate thesis to be submitted
 *  at Rutgers University, Newark.  This notice acts as permission
 *  to use and distribute, but not modify, without limitation.
 *
 *  Permission to modify is granted with the only limitation
 *  that any modified version must contain an attribution to the original
 *  and the original's author.
 *)
```

```
BeginPackage["CodesFunctions`"];

Shuffle[L_List] := Module[{len,halfLen},
    len = Length[L];
    halfLen = Floor[ (len + 1)/2 ];
    Take[
        Flatten[
            Transpose[{
                Take[L, {1, halfLen}],
                PadRight[
                    Reverse[Take[L, {halfLen + 1, -1} ]],
                    halfLen
                ]
            }]
        ],
        {1, len}
        ]
];
DeShuffle[L_List] := Module[{len},
    len = Length[L];
    Join[
        L[[ Range[1, len, 2] ]],
        Reverse[L[[ Range[2, len, 2] ]]]
    ]
];
```

```
IPQBits[p_Integer, q_Integer] := Join[Table[0, {p}], Table[1, {q}]];
ListXor[L_List, n_Integer] := IntegerDigits[
    Fold[ BitXor, 0, Map[(FromDigits[#1, 2]) &, L] ],
    2, n
];
QuasiSplitTwist[n_Integer,j_Integer] := Module[{m},
    m = Floor[n/2];
    ListXor[{
        IPQBits[n - m, m],
        IPQBits[n - j, j],
        Table[1, {n}]*Mod[m - j, 2]
        },
        n
    ]
];
NonQuasiSplitMask[n_Integer] := Prepend[Table[0,{n-1}],1];
NonQuasiSplitTwist[n_Integer,j_Integer] := Module[{m},
    m = Floor[n/2];
    ListXor[{
        NonQuasiSplitMask[n],
        IPQBits[m+1, m-1],
        IPQBits[n - j, j]
    }, n]
];
EndoscopicTwist[n_Integer,j_Integer] := Module[{m},
```

```
    m = Floor[n/2];
    If[ Mod[n, 2] == 1,
        QuasiSplitTwist[n, j],
        If[ Mod[m - j, 2] == 0,
            QuasiSplitTwist[n, j],
            NonQuasiSplitTwist[n, j]
        ]
    ]
];
ChamberCocycle[M_List,j_Integer] := Module[{halfLen,lambda},
    halfLen = Length[M] - j;
    lambda=(Block[{i},i=#1;
            Piecewise[{ {1,(i > halfLen)} , {0, (i <= halfLen)} }]
    ])&;
    Map[lambda,M]
];
EndoscopicCode[Ms_List,j_Integer] := Module[{n},
    n = Length[Ms];
    Shuffle[ListXor[{
        ChamberCocycle[Range[n], j],
        ChamberCocycle[DeShuffle[Ms], j],
        EndoscopicTwist[n, j]},
        n
    ]]
];
```

```
SecondEndoscopicCode[Ms_List] := BitXor[Ms,1];
StandardBasisChamber[Ms_List] := Module[{i = 1, j = Length[Ms]-Total[Ms]+1},
    Map[(If[(#1 == 0), i++, j++]) &, Ms]
];
NonCompactBasisChamber[Ms_List] := Shuffle[StandardBasisChamber[Ms]];
CodeToString[cl_List] := Fold[
    (StringJoin[ToString[#1],ToString[#2]])&,
    "",
    cl
];
ChamberToString[chl_List] := Apply[Greater,Map[(\!\(m\_#1\))&,chl]];
PrintEndoscopicCodes[p_Integer,q_Integer] :=
  Module[{useSecondCode,headings},
    useSecondCode = (Mod[p+q,2]==0 && (p != q));
    headings = {
            "Chamber",
            "Code"};
    If[useSecondCode,headings=Append[headings,"Second Code"]];
    TableForm[Map[(
        Block[{l=#1,code,code2,chamber,row},
            chamber = NonCompactBasisChamber[l];
            code = EndoscopicCode[chamber, q];
            code2 = SecondEndoscopicCode[code];
            row={
                ChamberToString[chamber],
```

```
                CodeToString[code]
            };
            If[ useSecondCode,
                Append[
                    row,
                    CodeToString[code2]
                ],
                row
            ]
        ])&,
        Permutations[IPQBits[p,q]]
        ],
        TableDirections -> {Column, Row, Row},
        TableHeadings -> {None,headings}
    ]
];
PrintCodesForDimension[n_Integer]:=Module[{j},
    For[j = Floor[n/2],
        j >= 0,
        j--,
        Print[
            "U(" <>
            ToString[n - j] <>
            "," <>
            ToString[j] <>
```

```
            "):"
        ];
        Print[PrintEndoscopicCodes[n - j, j]]
    ]
];
StringToCode[s_String] := ToExpression[StringSplit[s,""]];
CodeToChamberOdd[Code_List]:=
  Module[{LeftZeros,RightZeros,mjEven,j,t,cocycle,n,m,LeftOnes,RightOnes},
    n = Length[Code];
    m = Floor[n/2];
    LeftZeros = Count[Take[Code,{1,m+1}],0];
    RightZeros = Count[Take[Code,{m+2,n}],0];
    LeftOnes = m+1-LeftZeros;
    RightOnes = m-RightZeros;
    If[LeftZeros > RightZeros, mjEven=True, mjEven=False];
    If[mjEven, j=LeftOnes+RightZeros,j=RightOnes+LeftZeros];
    t = QuasiSplitTwist[n,j];
    cocycle = ListXor[{Code,t,IPQBits[n-j,j]},n];
    {NonCompactBasisChamber[cocycle],j}
];
CodeToChamberEvenQuasiSplit[Code_List]:=
  Module[{LeftZeros,RightZeros,LeftOnes,RightOnes,j,t,cocycle,n,m},
    n = Length[Code];
    m = Floor[n/2];
    LeftZeros = Count[Take[Code,{1,m}],0];
```

```
    RightZeros = Count[Take[Code,{m+1,n}],0];
    LeftOnes = m - LeftZeros;
    RightOnes = m - RightZeros;
    t = QuasiSplitTwist[n,j];
    If[LeftOnes+RightZeros > RightOnes+LeftZeros
        ,
        j = RightOnes+LeftZeros;
        t = BitXor[t,1];
        ,
        j = LeftOnes+RightZeros
    ];
    cocycle = ListXor[{Code,t,IPQBits[n-j,j]},n];
    {NonCompactBasisChamber[cocycle],j}
];
CodeToChamberEvenNonQuasiSplit[Code_List]:=
  Module[{LeftZeros,RightZeros,LeftOnes,RightOnes,
          j,t,invMask,cocycle,unmaskedCode,n,m},
    n = Length[Code];
    m = Floor[n/2];
    unmaskedCode = ListXor[{NonQuasiSplitMask[n],Code},n];
    LeftZeros = Count[Take[unmaskedCode,{1,m+1}],0];
    LeftOnes = m+1-LeftZeros;
    RightZeros = Count[Take[unmaskedCode,{m+2,n}],0];
    RightOnes = m-1 - RightZeros;
    If[LeftOnes+RightZeros > RightOnes+LeftZeros
```

```
        ,
        j = RightOnes+LeftZeros;
        invMask = Table[1,{n}]
        ,
        j = LeftOnes+RightZeros;
        invMask = Table[0,{n}]
    ];
    t = NonQuasiSplitTwist[n,j];
    cocycle = ListXor[{Code,t,IPQBits[n-j,j],invMask},n];
    {NonCompactBasisChamber[cocycle],j}
];
CodeToChamberEven[Code_List]:=
  Module[{mjEven,n,m},
    n = Length[Code];
    m = Floor[n/2];
    If[Mod[Count[Code,1],2]==0, mjEven=True, mjEven=False];
    If[mjEven,
       CodeToChamberEvenQuasiSplit[Code],
       CodeToChamberEvenNonQuasiSplit[Code]
    ]
];
CodeToHCData[C_List]:=Module[{n,m,Code},
    n=Length[C];
    m=Floor[n/2];
    Code=DeShuffle[C];
```

```
    If[ Mod[n,2]==1,CodeToChamberOdd[Code],CodeToChamberEven[Code] ]
];
HCDataToString[HCData_List] := Module[{j,chamber},
    {chamber,j} = HCData;
    TableForm[{"j="<>ToString[j]<>", ", ChamberToString[chamber]},
              TableDirections->{Row}]
];

VerifyCodeInverse[chamber_List,j_Integer] :=
  Module[{useSecondCode,code,code2,n,check},
    n = Length[chamber];
    useSecondCode = (Mod[n,2]==0 && (n-j != j));
    code = EndoscopicCode[chamber,j];
    code2 = SecondEndoscopicCode[code];
    check = (code == Apply[EndoscopicCode,CodeToHCData[code]]);
    If[useSecondCode,
       check = check &&
               (code == Apply[EndoscopicCode,CodeToHCData[code2]])
    ];
    {{ChamberToString[chamber],j},check}
];
VerifyCodeInverses[p_Integer,q_Integer] := TableForm[Map[
    VerifyCodeInverse[NonCompactBasisChamber[#1],q]&,
    Permutations[IPQBits[p,q]]
  ],TableDirections->{Column,Row,Row}
```

```
];
VerifyCodeInverses[n_Integer] := Module[{j},
    For[j=Floor[n/2],j>=0,j--,Print[VerifyCodeInverses[n-j,j]]]
];


EndPackage[];


(*----cut----CodesFunctions Package End----cut----*)
```

B Appendix: Codes For Dimensions 2-7

These codes were produced by running the first line below.

For[$i = 2, i \leq 7, i$++, PrintCodesForDimension[i]]

U(1,1):

Chamber	Code
$m_1 > m_2$	00
$m_2 > m_1$	11

U(2,0):

Chamber	Code	Second Code
$m_1 > m_2$	10	01

U(2,1):

Chamber	Code
$m_1 > m_3 > m_2$	000
$m_1 > m_2 > m_3$	011
$m_3 > m_2 > m_1$	110

U(3,0):

Chamber	Code
$m_1 > m_3 > m_2$	101

U(2,2):

Chamber	Code
$m_1 > m_4 > m_2 > m_3$	0000
$m_1 > m_4 > m_3 > m_2$	0011
$m_1 > m_2 > m_3 > m_4$	0110
$m_3 > m_4 > m_1 > m_2$	1001
$m_3 > m_2 > m_1 > m_4$	1100
$m_3 > m_2 > m_4 > m_1$	1111

U(3,1):

Chamber	Code	Second Code
$m_1 > m_4 > m_2 > m_3$	1000	0111
$m_1 > m_3 > m_2 > m_4$	1101	0010
$m_1 > m_3 > m_4 > m_2$	1110	0001
$m_4 > m_3 > m_1 > m_2$	0100	1011

U(4,0):

Chamber	Code	Second Code
$m_1 > m_4 > m_2 > m_3$	0101	1010

U(3,2):

Chamber	Code
$m_1 > m_5 > m_2 > m_4 > m_3$	00000
$m_1 > m_5 > m_2 > m_3 > m_4$	00011
$m_1 > m_3 > m_2 > m_5 > m_4$	01001
$m_1 > m_5 > m_4 > m_3 > m_2$	00110
$m_1 > m_3 > m_4 > m_5 > m_2$	01100
$m_1 > m_3 > m_4 > m_2 > m_5$	01111
$m_4 > m_5 > m_1 > m_3 > m_2$	10010
$m_4 > m_3 > m_1 > m_5 > m_2$	11000
$m_4 > m_3 > m_1 > m_2 > m_5$	11011
$m_4 > m_3 > m_5 > m_2 > m_1$	11110

U(4,1):

Chamber	Code
$m_1 > m_5 > m_2 > m_4 > m_3$	11101
$m_1 > m_4 > m_2 > m_5 > m_3$	10111
$m_1 > m_4 > m_2 > m_3 > m_5$	10100
$m_1 > m_4 > m_5 > m_3 > m_2$	10001
$m_5 > m_4 > m_1 > m_3 > m_2$	00101

U(5,0):

Chamber	Code
$m_1 > m_5 > m_2 > m_4 > m_3$	01010

U(3,3):

Chamber	Code
$m_1 > m_6 > m_2 > m_5 > m_3 > m_4$	000000
$m_1 > m_6 > m_2 > m_5 > m_4 > m_3$	000011
$m_1 > m_6 > m_2 > m_3 > m_4 > m_5$	000110
$m_1 > m_3 > m_2 > m_6 > m_4 > m_5$	010010
$m_1 > m_6 > m_4 > m_5 > m_2 > m_3$	001001
$m_1 > m_6 > m_4 > m_3 > m_2 > m_5$	001100
$m_1 > m_3 > m_4 > m_6 > m_2 > m_5$	011000
$m_1 > m_6 > m_4 > m_3 > m_5 > m_2$	001111
$m_1 > m_3 > m_4 > m_6 > m_5 > m_2$	011011
$m_1 > m_3 > m_4 > m_2 > m_5 > m_6$	011110
$m_4 > m_6 > m_1 > m_5 > m_2 > m_3$	100001
$m_4 > m_6 > m_1 > m_3 > m_2 > m_5$	100100
$m_4 > m_3 > m_1 > m_6 > m_2 > m_5$	110000
$m_4 > m_6 > m_1 > m_3 > m_5 > m_2$	100111
$m_4 > m_3 > m_1 > m_6 > m_5 > m_2$	110011
$m_4 > m_3 > m_1 > m_2 > m_5 > m_6$	110110
$m_4 > m_6 > m_5 > m_3 > m_1 > m_2$	101101
$m_4 > m_3 > m_5 > m_6 > m_1 > m_2$	111001
$m_4 > m_3 > m_5 > m_2 > m_1 > m_6$	111100
$m_4 > m_3 > m_5 > m_2 > m_6 > m_1$	111111

U(4,2):

Chamber	Code	Second Code
$m_1 > m_6 > m_2 > m_5 > m_3 > m_4$	100000	011111
$m_1 > m_6 > m_2 > m_4 > m_3 > m_5$	100101	011010
$m_1 > m_4 > m_2 > m_6 > m_3 > m_5$	110001	001110
$m_1 > m_6 > m_2 > m_4 > m_5 > m_3$	100110	011001
$m_1 > m_4 > m_2 > m_6 > m_5 > m_3$	110010	001101
$m_1 > m_4 > m_2 > m_3 > m_5 > m_6$	110111	001000
$m_1 > m_6 > m_5 > m_4 > m_2 > m_3$	101100	010011
$m_1 > m_4 > m_5 > m_6 > m_2 > m_3$	111000	000111
$m_1 > m_4 > m_5 > m_3 > m_2 > m_6$	111101	000010
$m_1 > m_4 > m_5 > m_3 > m_6 > m_2$	111110	000001
$m_5 > m_6 > m_1 > m_4 > m_2 > m_3$	000100	111011
$m_5 > m_4 > m_1 > m_6 > m_2 > m_3$	010000	101111
$m_5 > m_4 > m_1 > m_3 > m_2 > m_6$	010101	101010
$m_5 > m_4 > m_1 > m_3 > m_6 > m_2$	010110	101001
$m_5 > m_4 > m_6 > m_3 > m_1 > m_2$	011100	100011

U(5,1):

Chamber	Code	Second Code
$m_1 > m_6 > m_2 > m_5 > m_3 > m_4$	000101	111010
$m_1 > m_5 > m_2 > m_6 > m_3 > m_4$	010001	101110
$m_1 > m_5 > m_2 > m_4 > m_3 > m_6$	010100	101011
$m_1 > m_5 > m_2 > m_4 > m_6 > m_3$	010111	101000
$m_1 > m_5 > m_6 > m_4 > m_2 > m_3$	011101	100010
$m_6 > m_5 > m_1 > m_4 > m_2 > m_3$	110101	001010

U(6,0):

Chamber	Code	Second Code
$m_1 > m_6 > m_2 > m_5 > m_3 > m_4$	110100	001011

U(4,3):

Chamber	Code
$m_1 > m_7 > m_2 > m_6 > m_3 > m_5 > m_4$	0000000
$m_1 > m_7 > m_2 > m_6 > m_3 > m_4 > m_5$	0000011
$m_1 > m_7 > m_2 > m_4 > m_3 > m_6 > m_5$	0001001
$m_1 > m_4 > m_2 > m_7 > m_3 > m_6 > m_5$	0100001
$m_1 > m_7 > m_2 > m_6 > m_5 > m_4 > m_3$	0000110
$m_1 > m_7 > m_2 > m_4 > m_5 > m_6 > m_3$	0001100
$m_1 > m_4 > m_2 > m_7 > m_5 > m_6 > m_3$	0100100
$m_1 > m_7 > m_2 > m_4 > m_5 > m_3 > m_6$	0001111
$m_1 > m_4 > m_2 > m_7 > m_5 > m_3 > m_6$	0100111
$m_1 > m_4 > m_2 > m_3 > m_5 > m_7 > m_6$	0101101
$m_1 > m_7 > m_5 > m_6 > m_2 > m_4 > m_3$	0010010
$m_1 > m_7 > m_5 > m_4 > m_2 > m_6 > m_3$	0011000
$m_1 > m_4 > m_5 > m_7 > m_2 > m_6 > m_3$	0110000
$m_1 > m_7 > m_5 > m_4 > m_2 > m_3 > m_6$	0011011
$m_1 > m_4 > m_5 > m_7 > m_2 > m_3 > m_6$	0110011
$m_1 > m_4 > m_5 > m_3 > m_2 > m_7 > m_6$	0111001
$m_1 > m_7 > m_5 > m_4 > m_6 > m_3 > m_2$	0011110
$m_1 > m_4 > m_5 > m_7 > m_6 > m_3 > m_2$	0110110
$m_1 > m_4 > m_5 > m_3 > m_6 > m_7 > m_2$	0111100
$m_1 > m_4 > m_5 > m_3 > m_6 > m_2 > m_7$	0111111
$m_5 > m_7 > m_1 > m_6 > m_2 > m_4 > m_3$	1000010
$m_5 > m_7 > m_1 > m_4 > m_2 > m_6 > m_3$	1001000
$m_5 > m_4 > m_1 > m_7 > m_2 > m_6 > m_3$	1100000

U(4,3) (...Continued):

Chamber	Code
$m_5 > m_7 > m_1 > m_4 > m_2 > m_3 > m_6$	1001011
$m_5 > m_4 > m_1 > m_7 > m_2 > m_3 > m_6$	1100011
$m_5 > m_4 > m_1 > m_3 > m_2 > m_7 > m_6$	1101001
$m_5 > m_7 > m_1 > m_4 > m_6 > m_3 > m_2$	1001110
$m_5 > m_4 > m_1 > m_7 > m_6 > m_3 > m_2$	1100110
$m_5 > m_4 > m_1 > m_3 > m_6 > m_7 > m_2$	1101100
$m_5 > m_4 > m_1 > m_3 > m_6 > m_2 > m_7$	1101111
$m_5 > m_7 > m_6 > m_4 > m_1 > m_3 > m_2$	1011010
$m_5 > m_4 > m_6 > m_7 > m_1 > m_3 > m_2$	1110010
$m_5 > m_4 > m_6 > m_3 > m_1 > m_7 > m_2$	1111000
$m_5 > m_4 > m_6 > m_3 > m_1 > m_2 > m_7$	1111011
$m_5 > m_4 > m_6 > m_3 > m_7 > m_2 > m_1$	1111110

U(5,2):

Chamber	Code
$m_1 > m_7 > m_2 > m_6 > m_3 > m_5 > m_4$	1111101
$m_1 > m_7 > m_2 > m_5 > m_3 > m_6 > m_4$	1110111
$m_1 > m_5 > m_2 > m_7 > m_3 > m_6 > m_4$	1011111
$m_1 > m_7 > m_2 > m_5 > m_3 > m_4 > m_6$	1110100
$m_1 > m_5 > m_2 > m_7 > m_3 > m_4 > m_6$	1011100
$m_1 > m_5 > m_2 > m_4 > m_3 > m_7 > m_6$	1010110
$m_1 > m_7 > m_2 > m_5 > m_6 > m_4 > m_3$	1110001
$m_1 > m_5 > m_2 > m_7 > m_6 > m_4 > m_3$	1011001
$m_1 > m_5 > m_2 > m_4 > m_6 > m_7 > m_3$	1010011
$m_1 > m_5 > m_2 > m_4 > m_6 > m_3 > m_7$	1010000
$m_1 > m_7 > m_6 > m_5 > m_2 > m_4 > m_3$	1100101
$m_1 > m_5 > m_6 > m_7 > m_2 > m_4 > m_3$	1001101
$m_1 > m_5 > m_6 > m_4 > m_2 > m_7 > m_3$	1000111
$m_1 > m_5 > m_6 > m_4 > m_2 > m_3 > m_7$	1000100
$m_1 > m_5 > m_6 > m_4 > m_7 > m_3 > m_2$	1000001
$m_6 > m_7 > m_1 > m_5 > m_2 > m_4 > m_3$	0110101
$m_6 > m_5 > m_1 > m_7 > m_2 > m_4 > m_3$	0011101
$m_6 > m_5 > m_1 > m_4 > m_2 > m_7 > m_3$	0010111
$m_6 > m_5 > m_1 > m_4 > m_2 > m_3 > m_7$	0010100
$m_6 > m_5 > m_1 > m_4 > m_7 > m_3 > m_2$	0010001
$m_6 > m_5 > m_7 > m_4 > m_1 > m_3 > m_2$	0000101

U(6,1):

Chamber	Code
$m_1 > m_7 > m_2 > m_6 > m_3 > m_5 > m_4$	0001010
$m_1 > m_6 > m_2 > m_7 > m_3 > m_5 > m_4$	0100010
$m_1 > m_6 > m_2 > m_5 > m_3 > m_7 > m_4$	0101000
$m_1 > m_6 > m_2 > m_5 > m_3 > m_4 > m_7$	0101011
$m_1 > m_6 > m_2 > m_5 > m_7 > m_4 > m_3$	0101110
$m_1 > m_6 > m_7 > m_5 > m_2 > m_4 > m_3$	0111010
$m_7 > m_6 > m_1 > m_5 > m_2 > m_4 > m_3$	1101010

U(7,0):

Chamber	Code
$m_1 > m_7 > m_2 > m_6 > m_3 > m_5 > m_4$	1010101

C Appendix: Calculating Harish-Chandra Data From A Code And Other Auxiliary Functions

The general entry point to calculate the chamber and the j (of G^j) is function CodeToHCData. For example,

CodeToHCData[{1, 1, 0, 1, 1}]

produces

$\{\{4, 3, 1, 2, 5\}, 2\}$

So the code 11011 corresponds to a representation of G^2 with chamber $m_4 > m_3 > m_1 > m_2 > m_5$. This can be printed more clearly with HCDataToString:

HCDataToString[CodeToHCData[{1, 1, 0, 1, 1}]]

which outputs

$j = 2, \quad m_4 > m_3 > m_1 > m_2 > m_5$

The code entry can be simplified with StringToCode function:

StringToCode["11011"]

$\{1, 1, 0, 1, 1\}$

And putting it all together:

HCDataToString[CodeToHCData[StringToCode["11011"]]]

$j = 2, \quad m_4 > m_3 > m_1 > m_2 > m_5$

And to verify that this Harish-Chandra data will produce the same code:

Apply[EndoscopicCode,
CodeToHCData[StringToCode["11011"]]]

$\{1, 1, 0, 1, 1\}$

The module also has two functions (both with the same name) VerifyCodeInverses which will calculate all codes and their inverses. They, then, verify that the codes are what they should be. The first of these functions takes an p, q pair and checks all the codes for G^q for $n = p + q$ while the second of these functions takes just n and verifies all codes for dimension n.

References

[Art99] J. Arthur. On the transfer of distributions: weighted orbital integrals. *Duke Math*, 99:209–283, 1999.

[GW98] Roe Goodman and Nolan R. Wallach. *Representations and Invariants of the Classical Groups*. Cambridge University Press, 1998.

[Hal04] Brian C. Hall. *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*. Springer, first edition, 2004. 2nd printing (corrected).

[HC66] Harish-Chandra. Discrete series for semisimple lie groups II. Explicit determination of characters. *Acta Mathematica*, 116, 1966.

[Kna05] Anthony W. Knapp. *Lie Groups Beyond an Introduction*. Birkhäuser, second edition, 2005.

[LS87] R.P. Langlands and D. Shelstad. On the definition of transfer factors. *Mathematische Annalen*, 278:219–271, 1987. To Friedrich Hirzebruch on his sixtieth birthday.

[She08] Diana Shelstad. Examples in endoscopy for real groups. Part A of the notes on talk given at Banff, 2008.

[She09] Diana Shelstad. Tempered endoscopy for real groups II: spectral transfer factors. In *Automorphic Forms and the Langlands Program*, pages 236–276. International Press, 2009.

[Wal98] Nolan R. Wallach. *Real Reductive Groups I*. Academic Press, 1998.