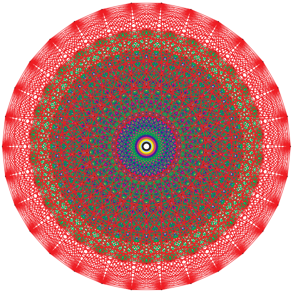
The Atlas of Lie Groups and Representations

www.liegroups.org



Atlas Project Members

- Jeffrey Adams
- Dan Barbasch
- Birne Binegar
- Bill Casselman
- Dan Ciubotaru
- Scott Crofts
- Fokko du Cloux
- Alfred Noel
- Tatiana Howard
- Alessandra Pantano
- Annegret Paul
- Patrick Polo
- Siddhartha Sahi
- Susana Salamanca
- John Stembridge
- Peter Trapa
- Marc van Leeuwen
- David Vogan
- Wai-Ling Yee
- Jiu-Kang Yu
- Gregg Zuckerman

red: directly worked on the E_8 calculation

E_8 was a worldwide media event in March, 2007:

E_8 was a worldwide media event in March, 2007:

- New York Times Science Section (March 20)
- Science
- Nature (online)
- Le Monde
- London Times
- Los Angeles Times
- Scientific American (online)
- Al Arabiya TV (satellite, Dubai)
- Economist
- Yahoo news (top 5 news, top emailed news story for several days)
- Good Morning America
- Fox News
- NPR
- Front page of the NSF site
- AP and other wire services

Why did E_8 take off in the press?

Why did E_8 take off in the press?

- We don't know

Why did E_8 take off in the press?

- We don't know
- Great graphic (Peter McMullen/John Stembridge)

Why did E_8 take off in the press?

- We don't know
- Great graphic (Peter McMullen/John Stembridge)
- Catchy title: A Caclulation the Size of Manhattan (on Eurekalert)

Why did E_8 take off in the press?

- We don't know
- Great graphic (Peter McMullen/John Stembridge)
- Catchy title: A Caclulation the Size of Manhattan (on Eurekalert)
- Computational aspect,huge amount of data, analogy with the genome project

Why did E_8 take off in the press?

- We don't know
- Great graphic (Peter McMullen/John Stembridge)
- Catchy title: A Caclulation the Size of Manhattan (on Eurekalert)
- Computational aspect,huge amount of data, analogy with the genome project
- Collaborative nature of the project

Why did E_8 take off in the press?

- We don't know
- Great graphic (Peter McMullen/John Stembridge)
- Catchy title: A Caclulation the Size of Manhattan (on Eurekalert)
- Computational aspect,huge amount of data, analogy with the genome project
- Collaborative nature of the project
- Symmetry and the mysterious 248 dimensional object

Why did E_8 take off in the press?

- We don't know
- Great graphic (Peter McMullen/John Stembridge)
- Catchy title: A Caclulation the Size of Manhattan (on Eurekalert)
- Computational aspect,huge amount of data, analogy with the genome project
- Collaborative nature of the project
- Symmetry and the mysterious 248 dimensional object
- "100 year old problem"

Why did E_8 take off in the press?

- Great external reviewers (Peter Sarnak, Hermann Nicolai, Gregg Zuckerman)

Why did E_8 take off in the press?

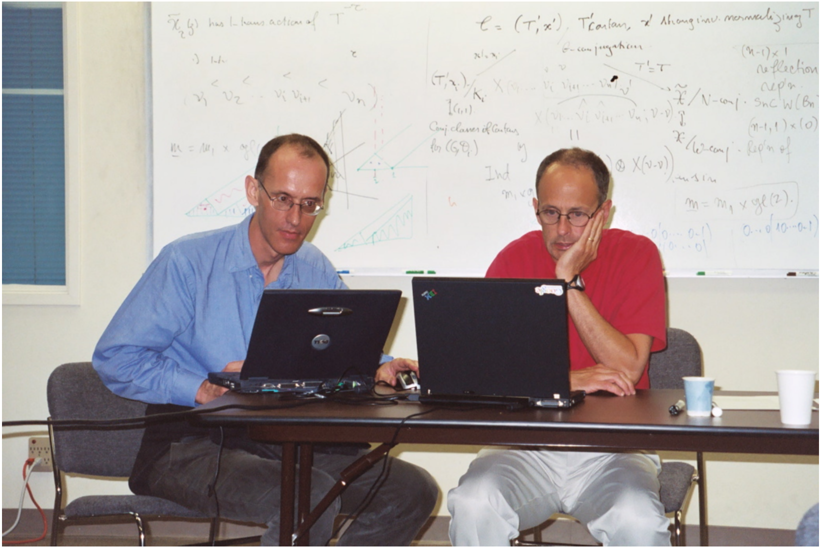
- Great external reviewers (Peter Sarnak, Hermann Nicolai, Gregg Zuckerman)
- Groundwork (Brian Conrey and David Farmer of AIM)

Why did E_8 take off in the press?

- Great external reviewers (Peter Sarnak, Hermann Nicolai, Gregg Zuckerman)
- Groundwork (Brian Conrey and David Farmer of AIM)
- Connection with string theory

Why did E_8 take off in the press?

- Great external reviewers (Peter Sarnak, Hermann Nicolai, Gregg Zuckerman)
- Groundwork (Brian Conrey and David Farmer of AIM)
- Connection with string theory
- It was not necessary to overly simplify the material or invent ties to other branches of mathematics or science


reflection
Conj classes of Cartans
Ind

What Fokko did

Abstract Mathematics		
Harish-Chandra		
Langlands		**Algorithm**
Knapp/Zuckerman/Vogan	$\longrightarrow$	Combinatorial set
Vogan		
Adams/Barbasch/Vogan		

What Fokko did

Abstract Mathematics		**Algorithm**		**Software**
Harish-Chandra				
Langlands				
Knapp/Zuckerman/Vogan	$\longrightarrow$	Combinatorial set	$\longrightarrow$	C++ code
Vogan				
Adams/Barbasch/Vogan				

Fokko du Cloux

December 20, 1954–November 10, 2006

Overview of the Atlas Project

G is a real (reductive) Lie group, such as:

Overview of the Atlas Project

G is a real (reductive) Lie group, such as:

$GL(n, \mathbb{R})$ ($n \times n$ invertible matrices)

Overview of the Atlas Project

G is a real (reductive) Lie group, such as:

$GL(n,\mathbb{R})$ ($n\times n$ invertible matrices)

$SO(p,q)$ (matrices preserving a quadratic form of signature (p,q))

Overview of the Atlas Project

G is a real (reductive) Lie group, such as:

$GL(n, \mathbb{R})$ ($n \times n$ invertible matrices)

$SO(p, q)$ (matrices preserving a quadratic form of signature (p, q))

$Sp(2n, \mathbb{R})$ (matrices preserving a skew-symmetric forms)

Overview of the Atlas Project

G is a real (reductive) Lie group, such as:

$GL(n, \mathbb{R})$ ($n \times n$ invertible matrices)

$SO(p, q)$ (matrices preserving a quadratic form of signature (p, q))

$Sp(2n, \mathbb{R})$ (matrices preserving a skew-symmetric forms)

A representation π of G is a homomorphism $\pi : G \to GL(\mathcal{H})$ (invertible operators on a Hilbert space $\mathcal{H}$). It is unitary if it is length preserving: $|\pi(g)v| = |v|$ for all $v \in \mathcal{H}$. It is irreducible if there are no closed invariant subspaces.

Example: $\mathcal{H} = L^2(G)$, $\pi(g)(f)(x) = f(g^{-1}x)$ This is the regular representation. It is highly reducible:

$$L^2(G) \simeq \int_{\hat{G}} \pi d\mu(\pi)$$

where $d\mu(\pi)$ is a measure on the space $G\hat{}$ of irreducible unitary representations of G.
More generally if G acts on X, preserving a measure μ, study action of G on X by linearizing, i.e. study representation of G on $L^2(X)$.

Problem: Compute the set of irreducible unitary representations of G.

Problem: Compute the set of irreducible unitary representations of G. Known:

- $SL(2,\mathbb{R})$ (Bargmann, 1947)

Problem: Compute the set of irreducible unitary representations of G. Known:

- $SL(2,\mathbb{R})$ (Bargmann, 1947)
- $GL(n,\mathbb{R})$ (Vogan, 1986)

Problem: Compute the set of irreducible unitary representations of G. Known:

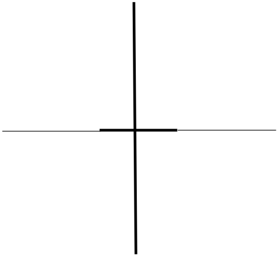
- $SL(2,\mathbb{R})$ (Bargmann, 1947)
- $GL(n,\mathbb{R})$ (Vogan, 1986)
- real rank 1: $SU(n,1), SO(n,1), Sp(n,1)$

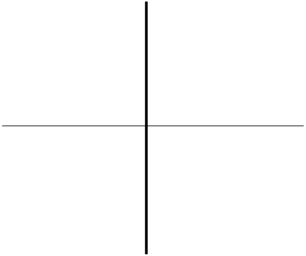
Problem: Compute the set of irreducible unitary representations of G. Known:

- $SL(2,\mathbb{R})$ (Bargmann, 1947)
- $GL(n,\mathbb{R})$ (Vogan, 1986)
- real rank 1: $SU(n,1), SO(n,1), Sp(n,1)$
- Complex classical groups: $SL(n,\mathbb{C}), SO(n,\mathbb{C}), Sp(2n,\mathbb{C})$ (Barbasch, 1989)

A few other small cases, no other infinite families

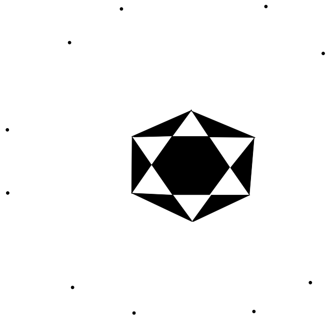
Unitary dual of $SL(2,\mathbb{R})$





$\mathbb{Z} - 0$

Spherical unitary dual of G_2



Theorem [... Vogan, 1980s]

Fix G. There is a finite algorithm to compute $G\widehat{\ }$.

Theorem [... Vogan, 1980s]

Fix G. There is a finite algorithm to compute $G\widehat{\ }$.

Note: $GL(7,\mathbb{R})$, not $GL(n,\mathbb{R})$

Theorem [... Vogan, 1980s]

Fix G. There is a finite algorithm to compute $G\hat{}$.

Note: $GL(7,\mathbb{R})$, not $GL(n,\mathbb{R})$

Not at all clear this algorithm can be made explicit, not to mention implemented on a computer.

Theorem [... Vogan, 1980s]

Fix G. There is a finite algorithm to compute $\widehat{G}$.

Note: $GL(7,\mathbb{R})$, not $GL(n,\mathbb{R})$

Not at all clear this algorithm can be made explicit, not to mention implemented on a computer.

Atlas of Lie Groups and Representations:

Theorem [...Vogan, 1980s]

Fix G. There is a finite algorithm to compute $G\hat{}$.

Note: $GL(7,\mathbb{R})$, not $GL(n,\mathbb{R})$

Not at all clear this algorithm can be made explicit, not to mention implemented on a computer.

Atlas of Lie Groups and Representations:

Take this idea seriously!

Goals:

1. Theoretical: Compute the unitary dual

Goals:

1. Theoretical: Compute the unitary dual
2. Educational:

Goals:

1. Theoretical: Compute the unitary dual
2. Educational:
    1. Provide software to compute with Lie groups and their representations.

Goals:

1. Theoretical: Compute the unitary dual
2. Educational:
    1. Provide software to compute with Lie groups and their representations.
    2. Provide information and interactive tools on a web site for non-experts.

The Groups

The following are in bijection:

1. Irreducible root systems

The Groups

The following are in bijection:

1. Irreducible root systems
2. Irreducible Dynkin diagrams

The Groups

The following are in bijection:

1. Irreducible root systems
2. Irreducible Dynkin diagrams
3. Simple complex Lie algebras

The Groups

The following are in bijection:

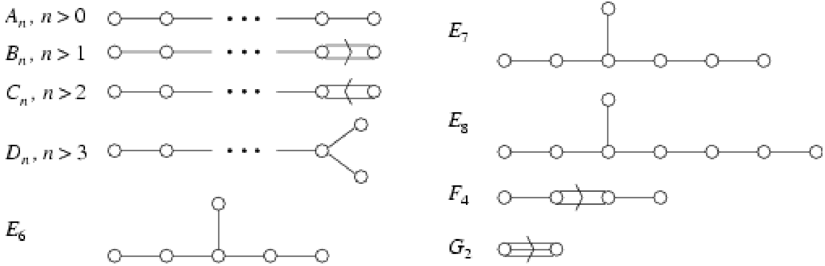
1. Irreducible root systems
2. Irreducible Dynkin diagrams
3. Simple complex Lie algebras
4. Simple complex Lie groups

The Groups

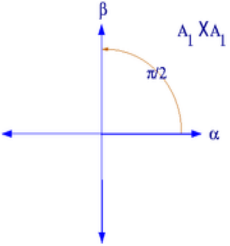
The following are in bijection:

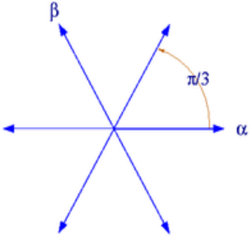
1. Irreducible root systems
2. Irreducible Dynkin diagrams
3. Simple complex Lie algebras
4. Simple complex Lie groups
5. A_n, B_n, C_n, D_n, $n = 1, 2, 3, \dots$ (classical) G_2, F_4, E_6, E_7, E_8 (exceptional)

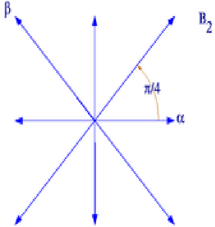
Dynkin Diagrams

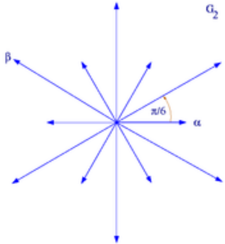


Rank Two Root Systems









Build all groups out of simple ones (similar to finite groups)

Build all groups out of simple ones (similar to finite groups)

$PSL(2,\mathbb{C}) = SL(2,\mathbb{C})/\pm I$

Build all groups out of simple ones (similar to finite groups)

$PSL(2,\mathbb{C}) = SL(2,\mathbb{C})/\pm I$

$SL(2,\mathbb{C}) \times SL(2,\mathbb{C})/(-I,-I)$

Build all groups out of simple ones (similar to finite groups)

$PSL(2,\mathbb{C}) = SL(2,\mathbb{C})/\pm I$

$SL(2,\mathbb{C}) \times SL(2,\mathbb{C})/(-I,-I)$

$GL(n,\mathbb{C}) = SL(n,\mathbb{C}) \times \mathbb{C}^\times/(\zeta,\zeta I)$

Build all groups out of simple ones (similar to finite groups)

$PSL(2,\mathbb{C}) = SL(2,\mathbb{C})/\pm I$

$SL(2,\mathbb{C}) \times SL(2,\mathbb{C})/(-I,-I)$

$GL(n,\mathbb{C}) = SL(n,\mathbb{C}) \times \mathbb{C}^\times/(\zeta,\zeta I)$

$\{(g,h) \in GL(n,\mathbb{C}) \times GL(m,\mathbb{C}) \mid \det(g)\det(h) = 1\}$

Grothendieck classified complex reductive (algebraic) groups in terms of root data:

$$(X, \Phi, X^\vee, \Phi^\vee)$$

where $X, X^\vee \simeq \mathbb{Z}^n$, Φ and $\Phi^\vee$ are finite subsets of $X, X^\vee$, in bijection ($\alpha \to \alpha^\vee$), satisfying properties:
$\langle \alpha, \alpha^\vee \rangle \in \mathbb{Z}$
$s_\alpha(\Phi^\vee) = \Phi^\vee, s_{\alpha^\vee}(\Phi) = \Phi$

Grothendieck classified complex reductive (algebraic) groups in terms of root data:

$$(X, \Phi, X^\vee, \Phi^\vee)$$

where $X, X^\vee \simeq \mathbb{Z}^n$, Φ and $\Phi^\vee$ are finite subsets of $X, X^\vee$, in bijection ($\alpha \to \alpha^\vee$), satisfying properties:
$\langle \alpha, \alpha^\vee \rangle \in \mathbb{Z}$
$s_\alpha(\Phi^\vee) = \Phi^\vee, s_{\alpha^\vee}(\Phi) = \Phi$

Data: two $m \times n$ matrices of integers.

Beautifully suited to a computer!

Each complex group has various real forms:

Each complex group has various real forms:

$SL(n, \mathbb{C}) \to SL(n, \mathbb{R}), SU(p, q), SL(n/2, \mathbb{H})$

Each complex group has various real forms:

$SL(n, \mathbb{C}) \rightarrow SL(n, \mathbb{R}), SU(p, q), SL(n/2, \mathbb{H})$

$SO(n, \mathbb{C}) \rightarrow SO(p, q), SO^*(n)$

Each complex group has various real forms:

$SL(n,\mathbb{C}) \rightarrow SL(n,\mathbb{R}), SU(p,q), SL(n/2,\mathbb{H})$

$SO(n,\mathbb{C}) \rightarrow SO(p,q), SO^*(n)$

There is always a unique compact real form $(SU(n), SO(n))$

Each complex group has various real forms:

$SL(n, \mathbb{C}) \rightarrow SL(n, \mathbb{R}), SU(p, q), SL(n/2, \mathbb{H})$

$SO(n, \mathbb{C}) \rightarrow SO(p, q), SO^*(n)$

There is always a unique compact real form $(SU(n), SO(n))$

There is always a unique split real form $(SL(n, \mathbb{R}), SO(n, n))$

First goal: write software to input an arbitrary real reductive group, and compute its structure theory.

The unitary representations occuring in $L^2(G)$ are known (Harish-Chandra, 1970s). These are called tempered: $\widehat{G}_t \subset \widehat{G}_u$.

Unitary representations are contained in a larger class, called admissible: $\widehat{G}_u \subset \widehat{G}_a$. These are also known (Langlands, Knapp, Zuckerman, Vogan)

$$\widehat{G}_t \subset \widehat{G}_u \subset \widehat{G}_a$$

To compute $\widehat{G}_u$: take each representation $\pi \in \widehat{G}_a$, and test if it is unitary. Not obvious this is a finite calculation even for a single π (not to mention uncountably many π).

Finite Calculation

How do we reduce to a finite calculation?

Finite Calculation

How do we reduce to a finite calculation?

Basic reduction: The number of irreducible representations with fixed "central character" for the Lie algebra is finite. Our calculations all take place in one of these fixed sets.

Finite Calculation

How do we reduce to a finite calculation?

Basic reduction: The number of irreducible representations with fixed “central character” for the Lie algebra is finite. Our calculations all take place in one of these fixed sets.

We will always work in the set of representations with the same “central character” as the trivial representation. This is the hardest case, others reduce to this.

Second Goal: find an algorithm to compute $\widehat{G}_a$, and write software to implement it.

Second Goal: find an algorithm to compute $\widehat{G}_a$, and write software to implement it.

More precisely: compute the finite set of irreducible admissible representations $\widehat{G}_{a,1}$ with trivial "central character".

Second Goal: find an algorithm to compute $\widehat{G}_a$, and write software to implement it.

More precisely: compute the finite set of irreducible admissible representations $\widehat{G}_{a,1}$ with trivial “central character”.

Although the mathematics is “known”, we greatly deepened our understanding of the mathematics in doing this.
For example: figuring out the data structures to adequately capture the mathematics required us to rethink the mathematics carefully.

Old days: representation of G on $L^2(X)$ (for example)

Old days: representation of G on $L^2(X)$ (for example)

Example: $G = SL(2,\mathbb{R})$ on $L^2(\mathbb{R})$:

$$\pi_\nu(g)f(x) = |-bx+d|^\nu f((ax-c)/(-bx+d))$$

where $g = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$

Old days: representation of G on $L^2(X)$ (for example)

Example: $G = SL(2,\mathbb{R})$ on $L^2(\mathbb{R})$:

$$\pi_\nu(g)f(x) = |-bx+d|^\nu f((ax-c)/(-bx+d))$$

where $g = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$

Today: $\pi = \cdot$

Old days: representation of G on $L^2(X)$ (for example)

Example: $G = SL(2, \mathbb{R})$ on $L^2(\mathbb{R})$:

$$\pi_\nu(g)f(x) = |-bx + d|^\nu f((ax - c)/(-bx + d))$$

where $g = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$

Today: $\pi = \cdot$

We parametrize $\widehat{G}_{a,1}$ by a finite set $\mathcal{X}$. Throw away π, and keep only the parameter space $\mathcal{X}$.

Algorithm to compute $\widehat{G}_{a,1}$

The heart of the algorithm is illustrated by this example.

Algorithm to compute $\widehat{G}_{a,1}$

The heart of the algorithm is illustrated by this example.

$G = GL(n, \mathbb{C})$

$B =$ upper triangular matrices

$X = G/B$ is a projective variety, a generalized Grassmannian

$H_m = GL(m, \mathbb{C}) \times GL(n - m, \mathbb{C})$

Algorithm to compute $\widehat{G}_{a,1}$

The heart of the algorithm is illustrated by this example.

$G = GL(n, \mathbb{C})$

$B =$ upper triangular matrices

$X = G/B$ is a projective variety, a generalized Grassmannian

$H_m = GL(m, \mathbb{C}) \times GL(n-m, \mathbb{C})$

Problem: Compute the orbits of H_m on X. This is a finite set. Compute the closure relations.

Combinatorial Solution:

$\widetilde{W}$ = generalized permutation matrices (one non-zero entry in each row and column)
$\simeq S^n \rtimes \mathbb{C}^{\times n}$
D = diagonal matrices
$\mathcal{X} = \{x \in \tilde{W} \mid x^2 = 1\}/D$

Combinatorial Solution:

$\widetilde{W}$ = generalized permutation matrices (one non-zero entry in each row and column)
$\simeq S^n \rtimes \mathbb{C}^{\times n}$
D = diagonal matrices
$\mathcal{X} = \{x \in \tilde{W} \mid x^2 = 1\}/D$

Fact: $\mathcal{X}$ is in natural bijection with $\cup_m X/H_m$
Computing $\mathcal{X}$ is an explicit combinatorial problem in finite group theory, a little harder than computing the elements of order 2 in S^n.

The software now calculates $\widehat{G}_{a,1}$ for any G.

Example: $SL(2,\mathbb{R})$:

```
This is the Atlas of Reductive Lie Groups Software Package version 0.2.5.
Build date: Nov 24 2006 at 09:16:16.
Enter "help" if you need assistance.

empty: block
Lie type: A1 sc s
(weak) real forms are:
0: su(2)
1: sl(2,R)
enter your choice: 1
possible (weak) dual real forms are:
0: su(2)
1: sl(2,R)
enter your choice: 1
Name an output file (hit return for stdout):
0(0,1):  1    (2,*)    [i1]  0
1(1,1):  0    (2,*)    [i1]  0
2(2,0):  2    (*,*)    [r1]  1  1
```

$Sp(4,\mathbb{R})$:

```
 0( 0,6):   1   2     ( 6, *)  ( 4, *)     [i1,i1]  0
 1( 1,6):   0   3     ( 6, *)  ( 5, *)     [i1,i1]  0
 2( 2,6):   2   0     ( *, *)  ( 4, *)     [ic,i1]  0
 3( 3,6):   3   1     ( *, *)  ( 5, *)     [ic,i1]  0
 4( 4,4):   8   4     ( *, *)  ( *, *)     [C+,r1]  1  2
 5( 5,4):   9   5     ( *, *)  ( *, *)     [C+,r1]  1  2
 6( 6,5):   6   7     ( *, *)  ( *, *)     [r1,C+]  1  1
 7( 7,2):   7   6     (10,11)  ( *, *)     [i2,C-]  2  2,1,2
 8( 8,3):   4   9     ( *, *)  (10, *)     [C-,i1]  2  1,2,1
 9( 9,3):   5   8     ( *, *)  (10, *)     [C-,i1]  2  1,2,1
10(10,0):  11  10     ( *, *)  ( *, *)     [r2,r1]  3  1,2,1,2
11(10,1):  10  11     ( *, *)  ( *, *)     [r2,rn]  3  1,2,1,2
```

So far we've said the atlas software should (and does) do:

1. Calculate with structure theory of reductive groups
2. Calculate the admissible dual $\widehat{G}_{a,\mathbf{1}}$.

So far we've said the atlas software should (and does) do:

1. Calculate with structure theory of reductive groups
2. Calculate the admissible dual $\widehat{G}_{a,1}$.

One more ingredient is needed.

Character theory

Let G be a finite group. Then a representation $\pi : G \to GL(n, \mathbb{C})$ is determined by its character $\theta_\pi(g) = \mathrm{Trace}(\pi(g))$.
The functions θ_π are a basis of $L^2(G)^G$.
So are $\chi_{\mathcal{O}}$ where $\mathcal{O}$ is a conjugacy class.

The character table of G contains all information about its representations:

Character Table of Weyl Group of type D4

Class		1	2	3	4	5	6	7	8	9	10	11	12	13
Size		1	1	6	6	6	12	12	32	12	24	24	24	32
Order		1	2	2	2	2	2	2	3	4	4	4	4	6
p =	2	1	1	1	1	1	1	1	8	2	5	4	3	8
p =	3	1	2	3	4	5	6	7	1	9	10	11	12	2
X.1	+	1	1	1	1	1	1	1	1	1	1	1	1	1
X.2	+	1	1	1	1	1	-1	-1	1	1	-1	-1	-1	1
X.3	+	2	2	2	2	2	0	0	-1	2	0	0	0	-1
X.4	+	3	3	-1	-1	3	-1	-1	0	-1	-1	1	1	0
X.5	+	3	3	3	-1	-1	-1	-1	0	-1	1	1	-1	0
X.6	+	3	3	-1	3	-1	-1	-1	0	-1	1	-1	1	0
X.7	+	3	3	-1	-1	3	1	1	0	-1	1	-1	-1	0
X.8	+	3	3	3	-1	-1	1	1	0	-1	-1	-1	1	0
X.9	+	3	3	-1	3	-1	1	1	0	-1	-1	1	-1	0
X.10	+	4	-4	0	0	0	-2	2	1	0	0	0	0	-1
X.11	+	4	-4	0	0	0	2	-2	1	0	0	0	0	-1
X.12	+	6	6	-2	-2	-2	0	0	0	2	0	0	0	0
X.13	+	8	-8	0	0	0	0	0	-1	0	0	0	0	1

We need the character table of G.

We need the character table of G.

$\mathcal{X}$ is the parameter space for $\widehat{G}_{a,1}$

We need the character table of G.

$\mathcal{X}$ is the parameter space for $\widehat{G}_{a,1}$

$x \rightarrow \pi(x) \in \widehat{G}_{a,1}$.

We need the character table of G.

$\mathcal{X}$ is the parameter space for $\widehat{G}_{a,1}$

$x \to \pi(x) \in \widehat{G}_{a,1}$.

$x \to I(x)$ a standard module. This is typically reducible, but is simpler than $\pi(x)$, and has a known character formula.

We need the character table of G.

$\mathcal{X}$ is the parameter space for $\widehat{G}_{a,1}$

$x \to \pi(x) \in \widehat{G}_{a,1}$.

$x \to I(x)$ a standard module. This is typically reducible, but is simpler than $\pi(x)$, and has a known character formula.

$$I(x) = \sum_{y \in \mathcal{X}} m(x,y)\pi(y) \quad m(x,y) \in \mathbb{Z}$$

Langlands, Zuckerman: this identity is invertible:

$$\pi(x) = \sum M(x,y) I(y)$$

This gives a character formula for $\pi(x)$.

Kazdhan-Lusztig, Vogan:
The integers $m(x, y), M(x, y)$ are computed in terms of the geometry of a complex group $K(\mathbb{C})$ acting on a complex projective algebraic ariety with finitely many orbits (intersection cohomology).

Kazdhan-Lusztig, Vogan:
The integers $m(x,y), M(x,y)$ are computed in terms of the geometry of a complex group $K(\mathbb{C})$ acting on a complex projective algebraic ariety with finitely many orbits (intersection cohomology).

For $x, y \in \mathcal{X}$ there is a polynomial $P_{x,y} \in \mathbb{Z}[q]$, such that

$$M(x,y) = \pm P_{x,y}(1)$$

Kazdhan-Lusztig, Vogan:
The integers $m(x,y), M(x,y)$ are computed in terms of the geometry of a complex group $K(\mathbb{C})$ acting on a complex projective algebraic ariety with finitely many orbits (intersection cohomology).

For $x, y \in \mathcal{X}$ there is a polynomial $P_{x,y} \in \mathbb{Z}[q]$, such that

$$M(x,y) = \pm P_{x,y}(1)$$

These are the famous Kazhdan-Lusztig-Vogan polynomials.

Kazdhan-Lusztig, Vogan:
The integers $m(x,y), M(x,y)$ are computed in terms of the geometry of a complex group $K(\mathbb{C})$ acting on a complex projective algebraic ariety with finitely many orbits (intersection cohomology).

For $x, y \in \mathcal{X}$ there is a polynomial $P_{x,y} \in \mathbb{Z}[q]$, such that

$$M(x,y) = \pm P_{x,y}(1)$$

These are the famous Kazhdan-Lusztig-Vogan polynomials.

Problem: compute $P_{x,y}$.

Summary of the atlas software

The atlas software now does the following:

1. Input arbitrary reductive complex algebraic group $G(\mathbb{C})$
2. Input real form G of $G(\mathbb{C})$
3. Compute structure theory of G
4. Compute the space $\mathcal{X}$ parametrizing $\widehat{G}_{a,1}$
5. Compute the Kazhdan-Lusztig-Vogan polynomials

Summary of the atlas software

The atlas software now does the following:

1. Input arbitrary reductive complex algebraic group $G(\mathbb{C})$
2. Input real form G of $G(\mathbb{C})$
3. Compute structure theory of G
4. Compute the space $\mathcal{X}$ parametrizing $\widehat{G}_{a,1}$
5. Compute the Kazhdan-Lusztig-Vogan polynomials

We hope this will be enough information to compute the unitary dual of G. It *is* enough information to list the most interesting, conjecturally unitary representations: the unipotent representations of Jim Arthur.

The hardest part of the calculation is the KLV polyonmials.

The hardest part of the calculation is the KLV polyonmials.

Split Group	time in seconds
$SL(2,\mathbb{R})$	.003
G_2	.008
F_4	.13
A_8	.17
A_9	.8
E_6	1.3
A_{10}	15
E_7	107
E_8	∞

Overview of the E_8 calculation

Recall E_8 is the largest exceptional group. The split real form is a real manifold of dimension 248, and it has $453,060$ irreducible representation in $\widehat{G}_{a,1}$.

Overview of the E_8 calculation

Recall E_8 is the largest exceptional group. The split real form is a real manifold of dimension 248, and it has 453,060 irreducible representation in $\widehat{G}_{a,1}$.

Problem: compute Kazhdan-Lusztig-Vogan polynomials for the split real form of E_8

Overview of the E_8 calculation

Recall E_8 is the largest exceptional group. The split real form is a real manifold of dimension 248, and it has $453,060$ irreducible representation in $\widehat{G}_{a,1}$.

Problem: compute Kazhdan-Lusztig-Vogan polynomials for the split real form of E_8

This is an upper triangular matrix, of size $453,060$, with 1^s on the diagonal, and polynomial entries. Each polynomial has degree ≤ 31.

Why?

Why?

1. Because it was there.

Why?

1. Because it was there.
2. Because David Vogan couldn't be stopped

Why?

1. Because it was there.
2. Because David Vogan couldn't be stopped
3. To test the mathematics.

Why?

1. Because it was there.
2. Because David Vogan couldn't be stopped
3. To test the mathematics.
4. To test the technology.

Why?

1. Because it was there.
2. Because David Vogan couldn't be stopped
3. To test the mathematics.
4. To test the technology.
5. To force us to improve the technology. We have much harder calculations to do to compute $\widehat{G}_u$. We have no hope of computing the unitary dual of F_4 if we can't compute KLV polynomials for E_8.

Why?

1. Because it was there.
2. Because David Vogan couldn't be stopped
3. To test the mathematics.
4. To test the technology.
5. To force us to improve the technology. We have much harder calculations to do to compute $\widehat{G}_u$. We have no hope of computing the unitary dual of F_4 if we can't compute KLV polynomials for E_8. It would not be enough to find a big enough computer.
6. Because E_8 is a particularly interesting group, and arises in string theory.

Recursion Relations

$\mathcal{X}$ is the set of parameters.
There is a partial order $<$ on $\mathcal{X}$, and a length function. For E_8 $\ell(x) \leq 62$.
The matrix is upper triangular:
$P_{x,x} = 1$
$P_{x,y} = 0$ unless $x \leq y$

Recursion Relations

$\mathcal{X}$ is the set of parameters.
There is a partial order $<$ on $\mathcal{X}$, and a length function. For E_8 $\ell(x) \leq 62$.
The matrix is upper triangular:
$P_{x,x} = 1$
$P_{x,y} = 0$ unless $x \leq y$

Recursion relations: compute $P_{x,y}$ by upward induction on $\ell(y)$ and downward induction on $\ell(y)$.
(0,0); (1,1), (0,1); (2,2), (1,2), (0,2)...

Recursion Relations

$\mathcal{X}$ is the set of parameters.
There is a partial order $<$ on $\mathcal{X}$, and a length function. For E_8 $\ell(x) \leq 62$.
The matrix is upper triangular:
$P_{x,x} = 1$
$P_{x,y} = 0$ unless $x \leq y$

Recursion relations: compute $P_{x,y}$ by upward induction on $\ell(y)$ and downward induction on $\ell(y)$.
(0,0); (1,1), (0,1); (2,2), (1,2), (0,2)...

Long list of complicated recursion formulas.

Recursion Relations

Type I: There exists y' with $\ell(y') < \ell(y)$ such that

$$P_{x,y} = \sum_{x'} c(x') P_{x',y'} \quad (\leq 3 \text{ terms})$$

Recursion Relations

Type I: There exists y' with $\ell(y') < \ell(y)$ such that

$$P_{x,y} = \sum_{x'} c(x') P_{x',y'} \quad (\leq 3 \text{ terms})$$

Type II: There is y', $\ell(y') = \ell(y)$, y'', $\ell(y'') = \ell(y) - 1$,

$$P_{x,y} = \sum_{\ell(x')=\ell(x)+1} P_{x',y'} + \sum_{x''} P_{x'',y''} \quad (\leq 4 \text{ terms})$$

)

Recursion Relations

Type III: There is x', y' with $\ell(x') = \ell(x) - 1, \ell(y') = \ell(y) - 1$,

$$P_{x,y} = P_{x',y'} + qP_{x,y'} - \sum_{x' \leq z < y'} \mu(z, y') q^{(l(y') - l(z) - 1)/2} P_{x',z}.$$

Average number of terms for E_8 is 150.

Recursion Relations

Type III: There is x', y' with $\ell(x') = \ell(x) - 1, \ell(y') = \ell(y) - 1$,

$$P_{x,y} = P_{x',y'} + qP_{x,y'} - \sum_{x' \leq z < y'} \mu(z, y') q^{(l(y') - l(z) - 1)/2} P_{x',z}.$$

Average number of terms for E_8 is 150.

Conclusion: In order to compute $P_{x,y}$ you need to use many all $P_{x',y'}$ with $\ell(y') < \ell(y)$.

Recursion Relations

Type III: There is x', y' with $\ell(x') = \ell(x) - 1, \ell(y') = \ell(y) - 1$,

$$P_{x,y} = P_{x',y'} + qP_{x,y'} - \sum_{x' \leq z < y'} \mu(z, y') q^{(l(y')-l(z)-1)/2} P_{x',z}.$$

Average number of terms for E_8 is 150.

Conclusion: In order to compute $P_{x,y}$ you need to use many all $P_{x',y'}$ with $\ell(y') < \ell(y)$.

We need to keep all $P_{x,y}$ in RAM!

Rough estimate

Problem: we did not have a good idea of the size of the answer beforehand.

Rough estimate

Problem: we did not have a good idea of the size of the answer beforehand.

Recall 1 byte= 8 bits can store $2^8 = 256$ numbers.

Rough estimate

Problem: we did not have a good idea of the size of the answer beforehand.

Recall 1 byte= 8 bits can store $2^8 = 256$ numbers.

We don't know the sizes of the coefficients. Proabably some are $> 65,535 = 2^{16} = 2$ bytes. We hope each coefficient is less than 4 bytes, i.e. 4.3 billion.

Rough estimate

Problem: we did not have a good idea of the size of the answer beforehand.

Recall 1 byte= 8 bits can store $2^8 = 256$ numbers.

We don't know the sizes of the coefficients. Proabably some are $> 65,535 = 2^{16} = 2$ bytes. We hope each coefficient is less than 4 bytes, i.e. 4.3 billion.

Each polynomial has ≤ 32 coefficients.

$450,060^2 \times 32 = 6.5$ trillion coefficients =26 trillion bytes

Many of the polynomials are equal for obvious reasons. Number of distinct polynomials ≤ 6 billion.
Store only the distinct polynomials.

Many of the polynomials are equal for obvious reasons. Number of distinct polynomials ≤ 6 billion.
Store only the distinct polynomials.

$6 \times 10^9 \times 32 = 200$ billion coefficents, or 800 billion bytes
Plus about 100 billion bytes for the pointers = 900 billion bytes

Many of the polynomials are 0, and many are equal for non-obvious reasons.

Many of the polynomials are 0, and many are equal for non-obvious reasons.

Hope: number of distinct polynomials is about 200 million
$300 \times 10^6 \times 4 \times 32 = 25$ billion bytes
Plus 100 billions bytes for index = 125 billion bytes

Many of the polynomials are 0, and many are equal for non-obvious reasons.

Hope: number of distinct polynomials is about 200 million
$300 \times 10^6 \times 4 \times 32 = 25$ billion bytes
Plus 100 billions bytes for index = 125 billion bytes

Marc van Leeuwen: much smarter indexing: 35 billion bytes $\rightarrow$
35+25=60 billion bytes

Many of the polynomials are 0, and many are equal for non-obvious reasons.

Hope: number of distinct polynomials is about 200 million
$300 \times 10^6 \times 4 \times 32 = 25$ billion bytes
Plus 100 billions bytes for index = 125 billion bytes

Marc van Leeuwen: much smarter indexing: 35 billion bytes $\rightarrow$ 35+25=60 billion bytes

Hope: average degree = 20 $\rightarrow$ 35+8=43 billion bytes

Bad news: experiments indicate the number of distinct polynomials is more like 800 billion $\rightarrow$ 65 billion bytes

Bad news: experiments indicate the number of distinct polynomials is more like 800 billion → 65 billion bytes

William Stein at Washington lent us sage, with 64 gigabytes of ram (all accessible from one processor)

Bad news: experiments indicate the number of distinct polynomials is more like 800 billion $\rightarrow$ 65 billion bytes

William Stein at Washington lent us sage, with 64 gigabytes of ram (all accessible from one processor)

Marc van Leeuwen and David Vogan spent a lot of time trying to squeeze down the calculation.
Marc reduced the size of the indices to about 15 billion bytes (by using a lot of information about the nature of the data)

Bad news: experiments indicate the number of distinct polynomials is more like 800 billion $\rightarrow$ 65 billion bytes

William Stein at Washington lent us sage, with 64 gigabytes of ram (all accessible from one processor)

Marc van Leeuwen and David Vogan spent a lot of time trying to squeeze down the calculation.
Marc reduced the size of the indices to about 15 billion bytes (by using a lot of information about the nature of the data)

David threaded the code to run many calculations simultaneously (on some platforms this slowed the calculation down

Calculating Modulo n

Noam Elkies: have to think harder

Idea:

Calculating Modulo n

Noam Elkies: have to think harder

Idea:

$\mathsf{2}^{16} = \mathsf{65,536} <$ Maximum coefficient $< \mathsf{2}^{32} = \mathsf{4.3}$ billion (?)

Calculating Modulo n

Noam Elkies: have to think harder

Idea:

$2^{16} = 65,536 <$ Maximum coefficient $< 2^{32} = 4.3$ billion (?)

$31 < 2^5$, so to do the calculation $\pmod p$ for $p < 32$ requires 5 bits for each coefficient instead of 32, reducing storage by a factor of $5/32$.

Calculating Modulo n

Noam Elkies: have to think harder
Idea:

$2^{16} = 65,536 <$ Maximum coefficient $< 2^{32} = 4.3$ billion (?)

$31 < 2^5$, so to do the calculation $\pmod p$ for $p < 32$ requires 5 bits for each coefficient instead of 32, reducing storage by a factor of $5/32$.

$2^{32} < 3 \times 5 \times 7 \times 11 \times 13 \times 17 \times 19 \times 23 \times 29 \times 31 = 100$ billion
You then get the answer mod 100,280,245,065 using the Chinese Remainder theorem (cost: running the calculation 9 times)

Calculating Modulo n

Noam Elkies: have to think harder

Idea:

$2^{16} = 65,536 <$ Maximum coefficient $< 2^{32} = 4.3$ billion (?)

$31 < 2^5$, so to do the calculation $\pmod p$ for $p < 32$ requires 5 bits for each coefficient instead of 32, reducing storage by a factor of $5/32$.

$2^{32} < 3 \times 5 \times 7 \times 11 \times 13 \times 17 \times 19 \times 23 \times 29 \times 31 = 100$ billion
You then get the answer mod 100,280,245,065 using the Chinese Remainder theorem (cost: running the calculation 9 times)

This gets us down to about $15 + 4 = 19$ billion bytes

But can we really reduce the calculation $\pmod{p}$?

But can we really reduce the calculation $\pmod{p}$?

The recursion relations use $+, -\times$ and extraction of coefficients in specific degrees. This last step looks bad but it is OK (coefficient=0 $\pmod{p}$, affects the recursion step, but you would have gotten $0 \pmod{p}$ anyway).

But can we really reduce the calculation $\pmod{p}$?

The recursion relations use $+, -\times$ and extraction of coefficients in specific degrees. This last step looks bad but it is OK (coefficient=0 $\pmod{p}$, affects the recursion step, but you would have gotten $0 \pmod{p}$ anyway).

In fact we can work $\pmod{n}$ for any n.

The final result

In the end:

Run the program 4 times modulo $n = 251, 253, 255, 256$
Least common multiple: 4,145,475,840
Combine the answers using the Chinese Remainder Theorem.
Answer is correct if the biggest coefficient is less then 4,145,475,840
Total time (on sage): 77 hours

Some Statistics

Some Statistics

Number of distinct polynomials: 1,181,642,979

Some Statistics

Number of distinct polynomials: 1,181,642,979

Maximal coefficient: 11,808,808

Some Statistics

Number of distinct polynomials: 1,181,642,979

Maximal coefficient: 11,808,808

Polynomial with the maximal coefficient:

$$152q^{22} + 3,472q^{21} + 38,791q^{20} + 293,021q^{19} + 1,370,892q^{18} + 4,067,059q^{17} + 7,964,012q^{16} + 11,159,003q^{15} + 11,808,808q^{14} + 9,859,915q^{13} + 6,778,956q^{12} + 3,964,369q^{11} + 2,015,441q^{10} + 906,567q^{9} + 363,611q^{8} + 129,820q^{7} + 41,239q^{6} + 11,426q^{5} + 2,677q^{4} + 492q^{3} + 61q^{2} + 3q$$

Some Statistics

Number of distinct polynomials: 1,181,642,979

Maximal coefficient: 11,808,808

Polynomial with the maximal coefficient:
$152q^{22} + 3,472q^{21} + 38,791q^{20} + 293,021q^{19} + 1,370,892q^{18} + 4,067,059q^{17} + 7,964,012q^{16} + 11,159,003q^{15} + 11,808,808q^{14} + 9,859,915q^{13} + 6,778,956q^{12} + 3,964,369q^{11} + 2,015,441q^{10} + 906,567q^{9} + 363,611q^{8} + 129,820q^{7} + 41,239q^{6} + 11,426q^{5} + 2,677q^{4} + 492q^{3} + 61q^{2} + 3q$

Value of this polynomial at q=1: 60,779,787

Some Statistics

Number of distinct polynomials: 1,181,642,979

Maximal coefficient: 11,808,808

Polynomial with the maximal coefficient:
$152q^{22} + 3,472q^{21} + 38,791q^{20} + 293,021q^{19} + 1,370,892q^{18} + 4,067,059q^{17} + 7,964,012q^{16} + 11,159,003q^{15} + 11,808,808q^{14} + 9,859,915q^{13} + 6,778,956q^{12} + 3,964,369q^{11} + 2,015,441q^{10} + 906,567q^{9} + 363,611q^{8} + 129,820q^{7} + 41,239q^{6} + 11,426q^{5} + 2,677q^{4} + 492q^{3} + 61q^{2} + 3q$

Value of this polynomial at q=1: 60,779,787

Number of coefficients in distinct polynomials: 13,721,641,221 (13.9 billion)

What comes next?

Using the results of the KLV calculation, we have a list of unipotent representations for E_8. These are conjecturally the building blocks of all unitary representations.

What comes next?

Using the results of the KLV calculation, we have a list of unipotent representations for E_8. These are conjecturally the building blocks of all unitary representations.

Serious mathematics to do:

What comes next?

Using the results of the KLV calculation, we have a list of unipotent representations for E_8. These are conjecturally the building blocks of all unitary representations.

Serious mathematics to do:

bringing K-types into the picture

What comes next?

Using the results of the KLV calculation, we have a list of unipotent representations for E_8. These are conjecturally the building blocks of all unitary representations.

Serious mathematics to do:

bringing K-types into the picture

Computing signatures of Hermitian forms

What comes next?

Using the results of the KLV calculation, we have a list of unipotent representations for E_8. These are conjecturally the building blocks of all unitary representations.

Serious mathematics to do:

bringing K-types into the picture

Computing signatures of Hermitian forms

Serious programming (Alfred Noel and Marc van Leeuwen)

What comes next?

Using the results of the KLV calculation, we have a list of unipotent representations for E_8. These are conjecturally the building blocks of all unitary representations.

Serious mathematics to do:

bringing K-types into the picture

Computing signatures of Hermitian forms

Serious programming (Alfred Noel and Marc van Leeuwen)

Big goal: the Unitary Dual

What comes next?

Using the results of the KLV calculation, we have a list of unipotent representations for E_8. These are conjecturally the building blocks of all unitary representations.

Serious mathematics to do:

bringing K-types into the picture

Computing signatures of Hermitian forms

Serious programming (Alfred Noel and Marc van Leeuwen)

Big goal: the Unitary Dual

Check back in a few years…