The Character Table for E_8

David Vogan

On January 8, 2007, just before 9 in the morning, a computer finished writing to disk about sixty gigabytes of files containing the Kazhdan-Lusztig polynomials for the split real group G of type E_8. Values at 1 of these polynomials are coefficients in characters of irreducible representations of G; so all irreducible characters were written down. The biggest coefficient appearing was 11,808,808, in the polynomial

$$\begin{gathered}152q^{22} + 3472q^{21} + 38791q^{20} + 293021q^{19} \\ +1370892q^{18} + 4067059q^{17} + 7964012q^{16} \\ +11159003q^{15} + 11808808q^{14} + 9859915q^{13} \\ +6778956q^{12} + 3964369q^{11} + 2015441q^{10} \\ +906567q^{9} + 363611q^{8} + 129820q^{7} \\ +41239q^{6} + 11426q^{5} + 2677q^{4} \\ +492q^{3} + 61q^{2} + 3q.\end{gathered}$$

Its value at 1 is 60,779,787.

This calculation is part of a larger project called the `atlas of Lie groups and representations`. In this article I'll try to explain the `atlas` project, what Kazhdan-Lusztig polynomials are, why one might care about them, and something about the nature of this calculation and the calculators.

David Vogan is professor of mathematics at the Massachusetts Institute of Technology. His email address is `dav@math.mit.edu`. *Supported in part by NSF FRG grant 0554278.*

What's E_8?

A *Lie group* is a group endowed with the structure of a smooth manifold, in such a way that group multiplication and inversion are smooth maps. Every finite group is a Lie group: the manifold structure is just the zero-dimensional discrete structure. For that reason the study of Lie groups is necessarily more complicated than the study of finite groups. But it's not unreasonable to concentrate on connected Lie groups. If you do that, a miracle happens: connected Lie groups are *less* complicated than finite groups. The reason is that a connected Lie group is almost completely determined by its *Lie algebra*. The Lie algebra is the tangent space to the group manifold at the identity element, endowed with a nonassociative product called the *Lie bracket*. Since the Lie algebra is a finite-dimensional vector space, it can be studied using linear algebra ideas. A typical method is to look at one element X of the Lie algebra, and to regard "Lie bracket with X" as a linear transformation from the Lie algebra to itself. This linear transformation has eigenvalues and eigenvectors, and those invariants can be used to describe the structure of the Lie algebra.

Just as finite groups are successive extensions of nonabelian simple groups (and $\mathbb{Z}/p\mathbb{Z}$), connected Lie groups are successive extensions of connected simple Lie groups (and the additive group $\mathbb{R}$). Many questions about general Lie groups can be reduced to the case of connected simple Lie groups, and so to questions about simple Lie algebras over the real numbers.

Many algebra problems are easier over algebraically closed fields, so it's natural to relate Lie algebras over $\mathbb{R}$ to Lie algebras over $\mathbb{C}$. If $k \subset K$ is any field extension, an n-dimensional Lie algebra

$\mathfrak{g}_k$ over the small field k gives rise naturally to an n-dimensional Lie algebra $\mathfrak{g}_K = \mathfrak{g}_k \otimes_k K$ over the large field K. The algebra $\mathfrak{g}_k$ is called a *k-form* of $\mathfrak{g}_K$. We can study real Lie algebras by first studying complex Lie algebras, and then studying their real forms.

Wilhelm Killing in 1887 was able to classify simple Lie algebras over the complex numbers. He found four infinite families of *classical Lie algebras* A_n, B_n, C_n, and D_n; and five *exceptional Lie algebras* G_2, F_4, E_6, E_7, and E_8. Each of these complex Lie algebras has a finite number of real forms; the real forms were described completely by Élie Cartan in the 1890s.

In each case there are two distinguished real forms: the *compact real form*, for which the corresponding Lie group is compact, and the *split real form*. The term "split" refers to factorization of certain characteristic polynomials. The split form has the property that there is an open set of Lie algebra elements X for which the linear transformation of Lie bracket with X has only real eigenvalues. If one works with simple Lie algebras over other fields, there is always an analogue of the split form. The compact form is special to the real field.

The Lie groups attached to classical Lie algebras are all related to classical linear algebra and geometry. A real Lie group of type B_n, for instance, is the group of linear transformations of $\mathbb{R}^{2n+1}$ preserving a nondegenerate quadratic form. These groups were already known in Lie's work, and in some sense they go back even to Euclid.

The great surprise in Killing's work was his discovery of the exceptional Lie algebras: five simple Lie algebras (of dimensions 14, 52, 78, 133, and 248) having no straightforward connection to classical geometry. Work in the twentieth century on the classification of finite simple groups shows that we should be delighted with such a short and tractable list of exceptions.

The `atlas` project is aimed at understanding the structure and representation theory of Lie groups. Each member of the project has a slightly different idea about what "understanding" means. Certainly it should include a thorough understanding of the exceptional groups.

There is an aesthetic in this subject according to which the best proof is one not referring to the Cartan-Killing classification. In practice, such a proof may be difficult to find. One of the most fundamental results is Cartan and Weyl's description of the finite-dimensional irreducible representations of a connected Lie group. This theorem was first proved in the 1930s using explicit constructions of representations of simple Lie groups, given separately in each case of the classification. Only twenty years later did Harish-Chandra give a construction independent of the classification. Even today, when Harish-Chandra's approach is well established as the "right" way to present the theory, the older explicit constructions continue to be a powerful tool.

What we are seeking is an understanding of Lie groups and (infinite-dimensional) representations of this aesthetically inferior sort: one that for the exceptional groups in particular may rely on explicit calculations. Always our goal is to find *formulations* of results that are as clean and simple and general as possible; but we allow for the possibility of verifying some results by long computations. The calculation we have done for E_8 is certainly long. I will say a few words in the last section about the kind of clean and simple results we are extracting from it.

A general warning about mathematical precision. I have tried to make the mathematical statements convey accurately our level of understanding; but I have deliberately omitted or obscured many important details. (Here is an example. In equation (D) below, I say that Harish-Chandra found a basis for the solutions of a system of differential equations. When certain eigenvalues for the system are zero, Harish-Chandra did *not* find all the solutions. The ones that he found suffice for the expression of irreducible characters: equation (E) remains true.) Undoubtedly the number of unintentional obscurities and omissions is equally large. For both categories, I apologize in advance.

The level of historical precision is perhaps even lower. I have omitted reference to many mathematicians whose work played a crucial role in the developments reported here. For these omissions I will not attempt to give an illustrative example but I will again apologize.

Unitary Representations and Their Disreputable Cousins

A *unitary representation* of a topological group G is a continuous action of G by automorphisms of a Hilbert space (preserving the inner product). Another way to say this is that a unitary representation is a realization of G as symmetries of a (possibly infinite-dimensional) Euclidean geometry. Because Hilbert spaces are the basic objects of quantum mechanics, one can also say that a unitary representation is a realization of G as symmetries of a quantum-mechanical system. A unitary representation is called *irreducible* if the Hilbert space has exactly two closed G-stable subspaces.

Because so many function spaces are closely related to Hilbert spaces, unitary representations are a fundamental tool for understanding actions of topological groups. To prepare this tool for use, we seek to understand arbitrary unitary representations of arbitrary topological groups.

An arbitrary unitary representation can often be written as a *direct integral* of irreducible unitary representations. The notion of direct integral extends that of direct sum. If $\mathbb{T}$ is the unit circle, the Hilbert space $L^2(\mathbb{T})$ has a direct sum decomposition $L^2(\mathbb{T}) = \sum_{n\in\mathbb{Z}} \mathbb{C} \cdot e^{in\theta}$ given by Fourier series. The Hilbert space $L^2(\mathbb{R})$ has a direct integral decomposition $L^2(\mathbb{R}) = \int_{\xi\in\mathbb{R}} \mathbb{C} \cdot e^{ix\xi} d\xi$ given by the Fourier transform.

There is an extremely general theorem guaranteeing *existence* of a direct integral decomposition into irreducible representations: it suffices that the topological group have a countable dense subset. There are moderately general theorems guaranteeing *uniqueness* of direct integral decompositions. This uniqueness holds (for example) for all algebraic Lie groups—subgroups of $n \times n$ matrices defined by polynomial equations in the matrix entries. We therefore seek to understand irreducible unitary representations for algebraic Lie groups.

Work begun by George Mackey and completed by Michel Duflo describes irreducible unitary representations of algebraic Lie groups by a very concrete and explicit reduction to the case of *reductive algebraic groups*. (These are essentially direct products of simple and abelian Lie groups.) One of the goals of the `atlas` project is this:

to describe the set $\Pi_u(G)$ of irreducible unitary representations of each reductive algebraic Lie group G.

Since we haven't yet reached this goal, we want to identify steps that represent progress towards it.

Harish-Chandra in the 1950s, following a suggestion of Chevalley, began to study the larger class of irreducible representations that are not necessarily unitary. A *representation* of a topological group G means a continuous action of G on a complete locally convex topological vector space. We may write it as a group homomorphism

$$\pi\colon G \to \mathrm{Aut}(V_\pi),$$

with V_π the vector space. We say that π is *irreducible* if V_π has exactly two closed invariant subspaces. The study of irreducible representations in general is complicated by the existence of invertible linear operators on infinite-dimensional Banach spaces having no nontrivial closed invariant subspaces. Such operators define irreducible representations of the group $\mathbb{Z}$, but they have little to do with most problems of harmonic analysis. Harish-Chandra found a natural technical condition called *quasisimplicity* on irreducible representations of reductive Lie groups that excludes such pathological behavior. (The condition is that all operators commuting with the representation are assumed to be scalar. In the case of unitary irreducible representations, quasisimplicity is the theorem called Schur's Lemma.) If G is any reductive algebraic Lie group, we define

$$\Pi_q(G) = \text{equivalence classes of irreducible quasisimple representations of } G.$$

(The correct notion of equivalence—Harish-Chandra's *infinitesimal equivalence*—is a bit subtle, and involves unbounded operators. That causes difficulties with making precise statements in the rest of this section, but nothing insurmountable.) Quasisimple representations turn out to be easier to describe than unitary representations.

What is the relation to the original problem of describing unitary representations? A unitary representation is automatically quasisimple, so we want to understand when a quasisimple irreducible representation is actually unitary. That is, we want to know whether V_π admits a G-invariant Hilbert space structure. This question can be broken into two parts: whether there is a G-invariant Hermitian form, and whether this form is definite. We can therefore define

$$\Pi_h(G) = \text{equivalence classes of irreducible quasisimple Hermitian representations of } G,$$

and get inclusions

$$\Pi_u(G) \subset \Pi_h(G) \subset \Pi_q(G).$$

The `atlas` goal of understanding the irreducible unitary representations of a reductive algebraic Lie group G can now be divided into three steps:

- **describe $\Pi_q(G)$ (all representations);**
- **describe $\Pi_h(G)$ (hermitian representations) as a subset of $\Pi_q(G)$; and**
- **describe $\Pi_u(G)$ (unitary representations) as a subset of $\Pi_h(G)$.**

The first two steps of this program have been addressed by Langlands and by Knapp-Zuckerman; there is an excellent account in Knapp's book *Representation Theory of Semisimple Groups*. Here is an approximate statement.

"Theorem" 1 (Langlands, Knapp-Zuckerman). *Suppose G is a reductive algebraic Lie group. Then the set $\Pi_q(G)$ of equivalence classes of irreducible quasisimple representations of G is in natural bijection with a countable discrete collection of complex algebraic varieties $X_i(\mathbb{C})$. Each of these algebraic varieties is defined over $\mathbb{R}$, and the subset $\Pi_h(G)$ corresponds to the real points $X_i(\mathbb{R})$.*

Stated in this way, the "Theorem" is equally true for reductive algebraic groups over arbitrary local fields. The quotation marks correspond to some small and well-understood technical difficulties; for the experts, the magic words are "R-groups". Each subset $\Pi_u(G) \cap X_i(\mathbb{R})$ is defined by real algebraic inequalities; the difficulty is that we do not know a simple description of those inequalities.

Representations of Compact Lie Groups

What is the nature of the information provided by "Theorem" 1 above? The example of compact Lie groups (which is a special case!) is helpful. I will go into some detail about that classical theory, seeking to formulate results in a way that carries over to general reductive algebraic Lie groups. An irreducible representation of a compact Lie group is automatically quasisimple, Hermitian, and unitary; so those distinctions will not appear at all.

Suppose K is a compact connected Lie group, and T is a maximal torus in K (a maximal connected abelian subgroup). Necessarily T is isomorphic to a product of ℓ circles, where ℓ is a nonnegative integer called the *rank* of K.

Any irreducible unitary representation of T must be one-dimensional. A one-dimensional unitary representation (of any topological group G) is a continuous homomorphism from G to the group $U(1)$ of 1×1 unitary matrices, which is again just the circle group. Any homomorphism from the circle to itself is given by raising to the mth power for some integer m. That is, $\Pi_u(\text{circle}) \simeq \mathbb{Z}$. Because T is a product of ℓ circles, it follows that $\Pi_u(T) \simeq \mathbb{Z}^\ell$. In a coordinate-free way, we say that $\Pi_u(T)$ is a lattice of rank ℓ.

The structure theory of compact Lie groups provides a finite collection $R^\vee(K,T)$ of nonzero $\mathbb{Z}$-linear maps $\alpha^\vee\colon \Pi_u(T) \to \mathbb{Z}$, called the *coroots of T in K*. The structure theory points also to certain natural subsets of the coroots, called *simple coroots*; we fix such a subset

$$S^\vee = \{\alpha_1^\vee, \dots \alpha_m^\vee\}. \qquad (A)$$

The simple coroots are linearly independent, so they define a nonempty cone

$$P = \{\mu \in \Pi_u(T) \mid \alpha^\vee(\mu) \geq 0 \quad (\alpha^\vee \in S^\vee)\}, \qquad (B)$$

called the *cone of dominant weights*.

It is a fundamental fact that everything about the structure of the compact Lie group K—and indeed of arbitrary reductive algebraic groups—is encoded by the lattice $\Pi_u(T)$ and the finite set of $\mathbb{Z}$-linear maps $S^\vee$. (To be precise, one needs also the simple roots $S \subset \Pi_u(T)$, introduced in (C) below.) In the hands of Chevalley and Grothendieck and others, this fact led to the theory of reductive groups over arbitrary fields (or even commutative rings). For the `atlas` project, it means that the structure of reductive groups can be described in terms of strings of integers and so is perfectly suited to exact computer calculation.

Theorem 2 (Cartan and Weyl). *Suppose K is a compact connected Lie group, and T is a maximal torus in K. Pick a set S of simple coroots for T in K as in (A) above, and define dominant weights P as in (B). Then there is a natural bijection between dominant weights and irreducible representations of K*

$$P \leftrightarrow \Pi_u(K).$$

In the language of the Langlands and Knapp-Zuckerman Theorem 1, P parametrizes the countable set of algebraic varieties X_i. Each X_i consists of a single point, so $X_i(\mathbb{C}) = X_i(\mathbb{R})$.

This theorem is very satisfactory as a parametrization of the irreducible representations of K, because the set P is very easy to compute and to manipulate. It is very unsatisfactory as a description of the irreducible representations, because it does not even explicitly describe the "natural bijection". In order to do that, we need a bit more structure theory. Attached to each coroot $\alpha^\vee$ there is a root $\alpha \in \Pi_u(T)$; the set of all roots is written

$$R(K,T) \subset \Pi_u(T).$$

The roots are by definition the nontrivial representations of T appearing in the "adjoint action" of T on the complexified Lie algebra of K. Corresponding to the simple coroots $S^\vee$ are simple roots

$$S = \{\alpha_1, \dots \alpha_m\} \subset R(K,T) \subset \Pi_u(T). \qquad (C)$$

Here is a description of the Cartan-Weyl bijection.

Theorem 3 (Cartan and Weyl). *In the setting of Theorem 2, an irreducible representation π of K corresponds to a dominant weight $\mu \in P$ if and only if the following conditions are satisfied: a) the weight μ appears in the restriction of π to T; and b) for every simple root $\alpha \in S$, the weight $\mu + \alpha$ does* not *appear in the restriction of π to T.*

Theorem 3 is in some sense a complete description of the irreducible representations of K, but it is still not completely satisfactory. It does not say how to calculate the dimension of a representation or its restriction to a compact subgroup of K. We will address those questions (and generalizations for noncompact groups) in the next section.

Character Tables for Lie Groups

Much of the content of the *Atlas of finite groups and representations* consists of character tables. In this section I'll recall what that means, how to extend the notion to Lie groups, and how it's possible to write character tables for reductive Lie groups in a finite form.

Suppose $\pi\colon G \to \mathrm{Aut}(V_\pi)$ is a representation of a topological group on a *finite-dimensional* complex vector space V_π. The *character of π* is a complex-valued function on G, defined by

$$\Theta_\pi(g) = \operatorname{tr} \pi(g).$$

It's very easy to see that Θ_π is a class function on G (that is, Θ_π is constant on conjugacy classes in G). What is not quite so obvious, but still elementary, is that *irreducible (finite-dimensional) representations having the same character are equivalent.*

In the case of a finite group G of order N, the eigenvalues of G are Nth roots of unity, so the values of characters of G are integer combinations of the N complex numbers $\exp(2\pi m i/N)$ (for m an integer between 0 and $N-1$). It is therefore possible to write a character of G precisely: for each conjugacy class in G, one can write the N integers that are the coefficients of these roots of unity. (Fortunately it is possible in practice to find far more compact representations of the character values.) A *character table* for G is a list of all the character values of all the irreducible representations.

If G is a compact Lie group, the notion of character table still makes sense (since the irreducible representations are finite-dimensional). What is not so clear is whether it can be written down in a finite way. The values of each character Θ_π must be specified at each of the infinitely many conjugacy classes in G; and then this task must be repeated for each of the infinitely many π.

Hermann Weyl solved these problems. I'll write his solution completely in the simplest case and then say a few words about the general case.

Theorem 4 (Weyl). *Suppose*

$$G = \{a + bi + cj + dk \mid a^2 + b^2 + c^2 + d^2 = 1\}$$

is the group of unit quaternions, and

$$T = \{a + bi \mid a^2 + b^2 = 1\} = \{\exp(i\theta) \mid \theta \in \mathbb{R}\} \subset G$$

is a maximal torus. a) There is exactly one irreducible representation π_n of G for each strictly positive integer n. b) Every conjugacy class in G meets T, so a class function on G is determined by its restriction to T. c) The value of the character of π_n on T is $\Theta_{\pi_n}(\exp(i\theta)) = \sin(n\theta)/\sin(\theta)$.

We have in (c) an infinite character table presented in finite form. The infinitely many rows are indexed by n, and the infinitely many columns by θ (more precisely, by θ up to sign and addition of multiples of 2π).

In part (a) of the Theorem, one can think of the integer $n-1$ as corresponding to a one-dimensional representation of T, that is, to an element of $\Pi_u(T)$. A version of (a) for general compact Lie groups is provided by the Cartan-Weyl Theorem 2 in the last section. Part (b) makes sense as stated for a general compact connected Lie group and is true. The Weyl character formula for general G looks something like (c). There is a denominator (generalizing the function $\sin(\theta)$) that is a trigonometric polynomial on T, independent of the representation. The numerator (generalizing $\sin(n\theta)$) is a trigonometric polynomial built from the weight μ that parametrizes the representation.

For infinite-dimensional representations, the difficulties with character theory are more fundamental. The operators $\pi(g)$ are essentially never of trace class. Harish-Chandra understood that the character makes sense only after "regularization" in the sense of distribution theory. Each individual operator $\pi(g)$ does not have a trace: one first has to smooth the operator by averaging over a nice compact collection of values of g.

Here is how to do that. Recall that a *test density* on a smooth manifold M is a compactly supported complex-valued measure ξ on M, which in local coordinates is a smooth multiple of Lebesgue measure. A *generalized function* on M is a continuous linear functional on the space of test densities. Any continuous function f on M defines a generalized function by the formula

$$f(\xi) = \int_M f(m)d\xi(m).$$

The trace of a finite-dimensional representation of a Lie group G is a continuous function on G and therefore may be regarded as a generalized function. The following theorem of Harish-Chandra shows that the character of an irreducible quasisimple representation of a reductive algebraic Lie group G is a generalized function on G.

Theorem 5 (Harish-Chandra). *Suppose G is a reductive algebraic Lie group, π is an irreducible quasisimple representation of G on a Hilbert space, and ξ is a test density on G. The operator $\pi(\xi) = \int_G \pi(g)d\xi(g)$ is trace class, and defining $\Theta_\pi(\xi) = \operatorname{tr}\pi(\xi)$ makes Θ_π a generalized function on G.*

There is a conjugation-invariant open subset $G' \subset G$, whose complement has measure zero, so that the restriction of Θ_π to G' is a conjugation-invariant analytic function Θ'_π, locally integrable on G. The generalized function Θ_π is equal to integration against Θ'_π.

Writing a character table for the reductive algebraic Lie group G means writing down each of the functions Θ'_π, as π runs over the (infinite) family of irreducible quasisimple representations of G. The reason that each such function can be written down is that it turns out (just as in the case of compact groups) to be a quotient of finite integer combinations of exponential functions.

The possibility of handling infinitely many π is a consequence of the Jantzen-Zuckerman "translation principle". They partition all irreducible representations into finitely many *translation families.* In the formulas for the characters of representations in one translation family, only the exponential functions change: the coefficients in the formulas remain the same. In the case of a compact group, there is single translation family, with characters given by the Weyl character formula; all that varies with the representation are the exponents. For example, for the quaternion group described in Theorem 4, the parameter for the translation family is the integer n.

Here is a little more detail. The differential equations for the character Θ_π come from the center $\mathfrak{Z}$ of the universal enveloping algebra of the Lie algebra of G. They are eigenvalue equations; the eigenvalues are the complex scalars by which $\mathfrak{Z}$ acts in the representation π. (That $\mathfrak{Z}$ *does* act by scalars is exactly Harish-Chandra's definition of quasisimplicity.) The eigenvalues are encoded by an algebra homomorphism $\lambda\colon \mathfrak{Z} \to \mathbb{C}$ called the *infinitesimal character* of π.

After appropriate (very subtle!) changes of variables, the differential equations become (in local coordinates) systems of constant-coefficient eigenvalue equations on $\mathbb{R}^n$. For each choice λ of eigenvalues, the solutions are finite linear combinations of exponential functions. Harish-Chandra was able to describe these solutions very explicitly and completely. (Much of the difficulty is understanding how solutions on different coordinate patches fit together.) For each choice of eigenvalues, he found an explicit basis

$$\Theta_1, \Theta_2, \ldots, \Theta_N \qquad (D)$$

for the global solutions of the differential equations. (It would be mathematically more precise but notationally more burdensome to write $\Theta_1^\lambda, \Theta_2^\lambda, \ldots, \Theta_{N(\lambda)}^\lambda$, indicating explicitly the fact that the equations being solved depend on the system λ of eigenvalues. I will choose the path of unburdened imprecision.) As a consequence, each irreducible character Θ_π has a unique expression

$$\Theta_\pi = \sum_{j=1}^{N} \overline{P}_{\pi,j}\Theta_j, \qquad (E)$$

for some complex numbers $\overline{P}_{\pi,j}$. (The reason for calling the coefficients $\overline{P}$ will emerge in the next section: the overline means the image in a quotient map, defined from polynomials to coefficients by evaluation at 1.) Writing down an irreducible character Θ_π is therefore equivalent to writing down the N complex numbers $\overline{P}_{\pi,j}$.

The last important fact is that the coefficients $\overline{P}_{\pi,j}$ are all *integers*. It is not quite clear to whom this observation should be attributed. At least with forty years of hindsight, it is easy to deduce from the work of Langlands and Knapp-Zuckerman on the classification of irreducible representations. Zuckerman may have been the first to recognize the existence and importance of character formulas (E) with integer coefficients. He wrote an explicit formula for the character of a finite-dimensional representation (ϕ, F) in his thesis; in that case the coefficients $\overline{P}_{\phi,j}$ are all ± 1 or zero.

How to Compute the Characters

Equation (E) above says that the character table for a real reductive Lie group G may be expressed as a matrix of integers $\overline{P}_{\pi,j}^\lambda$; here I have temporarily reinserted the dependence on the infinitesimal character λ. The index j runs over Harish-Chandra's solutions (D) to the differential equations (with fixed eigenvalues λ). The index π runs over irreducible representations of G (with fixed infinitesimal character λ).

The Jantzen-Zuckerman translation principle says (in a very explicit and computable way) that, as λ varies, there are only finitely many possibilities for $\overline{P}_{\pi,j}^\lambda$. Henceforth I will therefore drop the λ and speak only of computing one matrix $\overline{P}_{\pi,j}$. Thinking about this matrix is simplified by

Theorem 6 (Langlands and Knapp-Zuckerman). *There is a natural bijection between the set $\{\pi\}$ of irreducible representations of G and Harish-Chandra's solutions (D) to the differential equations for characters. Write π_i for the irreducible representation corresponding to the solution Θ_i. Then the (square) matrix $\overline{P}_{\pi_i,j}$ in (E) above is a lower triangular integer matrix with 1s on the diagonal.*

To say that the matrix is lower triangular requires an appropriate ordering of the solutions Θ_j. Harish-Chandra's construction of the solutions analyzes exponential terms of greatest possible growth at infinity on G. If we assume that the Θ_j are ordered so that the later ones have faster growth, then we get the lower triangularity. Another way to achieve it is explained after equation (F) below.

In this section I will say a bit about the mathematics underlying the computation of the matrix $\overline{P}_{\pi_i,j}$. The main tool is a geometric reinterpretation of the matrix introduced by Beilinson and Bernstein. It takes place in a smooth complex projective algebraic variety X (depending on the reductive group G) called the *complete flag variety* $X = X(G)$. One way to define X is as the variety of maximal solvable Lie subalgebras inside the complexified Lie algebra $\mathfrak{g}_\mathbb{C}$ of G.

The variety X is degenerate in some very interesting ways. Most algebraic varieties are not $\mathbb{P}^1$ bundles in any way. The variety X has a finite collection

$$\pi_s \colon X \to X_s \qquad (s \in S) \qquad (F)$$

of $\mathbb{P}^1$ fibrations. (That is, each π_s is a smooth submersion with fiber the Riemann sphere $\mathbb{CP}^1$.) The parametrizing set S is the set of simple roots introduced in (C) above.

In case G is $GL(n, \mathbb{R})$, the variety X may be identified with complete flags in $\mathbb{C}^n$: increasing chains of n linear subspaces F_j, with $\dim F_j = j$. There are $n-1$ $\mathbb{P}^1$ fibrations; for $1 \le j < n$, the jth fibration arises by throwing away the j-dimensional subspace F_j in a complete flag.

The ideas of Beilinson and Bernstein concern the equivariant geometry of X. One might expect that what ought to matter is the action of G on X. For technical reasons, however, what enters their work is equivariance with respect to $K(\mathbb{C})$, the complexification of a maximal compact subgroup $K \subset G$.

Theorem 7 (Beilinson and Bernstein) *Suppose G is a real reductive group with complete flag variety X, K is a maximal compact subgroup of G, and $K(\mathbb{C})$ is its complexification (an algebraic group acting on X).*

a) Harish-Chandra's solutions (D) to the differential equations are naturally in one-to-one correspondence with pairs $(Z_o, \mathcal{L})$ consisting of a $K(\mathbb{C})$ orbit Z_o on X and a $K(\mathbb{C})$-equivariant local system $\mathcal{L}$ on Z_o.

According to Theorem 6, exactly the same parameters index the irreducible representations of G.

b) Suppose π_i is an irreducible representation, corresponding to the pair $(Z_{i,o}, \mathcal{L}_i)$. Write Z_i for the closure of $Z_{i,o}$, a (possibly singular) algebraic subvariety of X. Suppose Θ_j is one of Harish-Chandra's solutions (D), corresponding to the pair $(Z_{j,o}, \mathcal{L}_j)$. Then the character formula coefficient $\overline{P}_{\pi_i,j}$ of (E) is equal to the Euler characteristic of the local intersection homology of Z_i with coefficients in the local system $\mathcal{L}_i$, evaluated at $(Z_{j,o}, \mathcal{L}_j)$.

Because Harish-Chandra solved systems of differential equations to get the Θ_j, and Beilinson and Bernstein work with derived categories of constructible sheaves, you may imagine that there is some work required to pass from what Beilinson and Bernstein proved to the statement of Theorem 7. This translation was one of my own contributions to the subject. The most difficult part was convincing some of the experts that the result was really not quite obvious.

Despite my promise at the beginning to be historically incomplete, I have been asked to interrupt the mathematics here to say a few words about the sources of the ideas in Theorem 7. Kazhdan and Lusztig in 1979 formulated a precise conjecture describing the characters of irreducible highest weight modules in terms of the combinatorics of the Weyl group. They observed that the first appearance in these characters of coefficients other than ± 1 can be empirically related to the first failures of local Poincaré duality for singular Schubert varieties. MacPherson suggested that their observations might be formalized using intersection homology. Kazhdan and Lusztig did this, proving that their combinatorial construction in the Weyl group actually calculated the intersection homology of Schubert varieties. This calculation is short but very deep, using the tools developed by Deligne to prove the Weil conjectures. It seems to have been an inspiration for the development by Beilinson, Bernstein, Deligne, and Gabber of the general theory of perverse sheaves, which has since become a basic tool in representation theory and algebraic geometry.

The Kazhdan-Lusztig conjectural character formula (still for highest weight modules) was now a statement involving intersection homology of Schubert varieties. In this form it was proved independently by Brylinski-Kashiwara and by Beilinson-Bernstein, using the algebraic theory of differential equations created by Sato, Kashiwara-Kawai, and Beilinson-Bernstein.

I will not try to describe "intersection homology" here. (The book *Introduction to Intersection Homology Theory* by Kirwan and Woolf is highly recommended by my colleagues who should know.) In order to understand the nature of the statement, what matters is that intersection homology is a topological invariant of the singular algebraic variety Z_i. It measures at the same time the nature of the singularity of Z_i (in the theorem, the singularity at points in $Z_{j,o}$) and the possibility of extending the local system $\mathcal{L}_i$ from the open subset $Z_{i,o}$ to all of Z_i.

As the term "Euler characteristic" suggests, intersection homology provides (for each i and j) not just a single integer but rather a finite collection of nonnegative integers p_{ij}^m, the ranks of individual local intersection homology groups. I will modify the indexing of the homology in order to arrange that the index m can run from 0 to the complex codimension of Z_j in Z_i. (One of the key properties of intersection homology is that the top degree can appear only if $Z_j = Z_i$.) A consequence of this reindexing is that the Euler characteristic of Theorem 7(b) is

$$(-1)^{\dim Z_i - \dim Z_j} \sum_m (-1)^m p_{ij}^m.$$

If Z_i is smooth and $\mathcal{L}_i$ is the trivial local system, then p_{ij}^m is equal to zero unless $m = 0$, Z_j is contained in Z_i, and the local system $\mathcal{L}_j$ is also trivial; in that case p_{ij}^0 is equal to 1.

The *Kazhdan-Lusztig polynomial* for the pair (i, j) is by definition

$$P_{i,j}(q) = \sum_m p_{ij}^m q^{m/2}.$$

The parameters i and j represent local systems on orbits of $K(\mathbb{C})$ on the complete flag variety X. The polynomial can be nonzero only if the orbit $Z_{j,o}$ is contained in the closure of the orbit $Z_{i,o}$; this explains the "lower triangular" result in Theorem 6. It turns out that the local groups vanish in odd degrees, so that $P_{i,j}$ is actually a polynomial in q (with nonnegative integer coefficients). (This is a special fact about the varieties Z_i, not a general fact about intersection homology.) The degree of $P_{i,j}$ is bounded by half the complex codimension of Z_j in Z_i; the bound is strict if $i \neq j$. (This is a general fact about intersection homology.) Because of the vanishing in odd degrees, the Euler characteristic is just the (nonnegative) value at $q = 1$, times the sign $(-1)^{\dim Z_i - \dim Z_j}$.

The point of Theorem 7 is that characters can be computed from Kazhdan-Lusztig polynomials and that these polynomials depend on the geometry of $K(\mathbb{C})$ orbit closures on the complete flag variety X. The next theorem describes the geometric tools needed to compute intersection homology for these orbit closures.

Theorem 8 (Wolf). *Suppose G is a real reductive group with complete flag variety X, K is a maximal compact subgroup of G, and $K(\mathbb{C})$ is its complexification (an algebraic group acting on X).*

a) The action of $K(\mathbb{C})$ on X has finitely many orbits.

Suppose Z_o is an orbit of $K(\mathbb{C})$ on X, and $s \in S$. Write Z for the closure of Z_o (a projective algebraic subvariety of X). Define $Z_s = \pi_s(Z)$ (cf. (F) above), a projective algebraic subvariety of X_s, which we call the s-flattening of Z. Define $Z^s = \pi_s^{-1}(Z_s)$, which we call the s-thickening of Z. The map π_s exhibits Z^s as a $\mathbb{P}^1$ bundle over Z_s.

b) The s-thickening Z^s is the closure of a unique $K(\mathbb{C})$ orbit Z_o^s.

c) There are two mutually exclusive possibilities. 1) The s-thickening Z^s is equal to Z, so that Z is a $\mathbb{P}^1$ bundle over Z_s. In this case $Z_o^s = Z_o$. 2) The map π_s from Z to Z_s is generically finite (and in particular is finite over the orbit Z_o). In this case $\dim Z_o^s = \dim Z_o + 1$*; and $Z_s = (Z^s)_s$. In particular, the thickened orbit Z_o^s falls in case (1).*

d) Every orbit of $K(\mathbb{C})$ on X arises by a finite succession of thickening operations applied to some closed orbit.

Part (a) of this theorem is due to Wolf. The remaining assertions are quite easy, although it is more difficult (for me at least) to attribute them precisely. The idea of constructing and describing the orbits in this way has its roots in the theory of Schubert varieties and so is very old.

Theorem 8 describes the geometry of orbit closures and so leads to Kazhdan and Lusztig's algorithm for computing intersection homology by induction on the dimension. Here is a sketch. An orbit of minimal dimension is closed and is itself a complete flag variety for the algebraic group $K(\mathbb{C})$. Such varieties are smooth, so the local intersection cohomology is simple. According to part (d), any orbit W_o of greater than minimal dimension must arise by thickening: $W_o = Z_o^s$, with Z_o an orbit of dimension one less. Now the orbit closure Z is a ramified cover of the flattening Z_s (according to (c)(2)); so the intersection homology of Z_s is very close to that of Z, which is known by induction. Finally the orbit closure W is a $\mathbb{P}^1$ bundle over $W_s = Z_s$; so the intersection homology of W is made in a simple way from that of the flattened variety Z_s.

Making this sketch precise uses versions of the Weil conjectures for intersection homology, proved by Beilinson, Bernstein, and Deligne. The most subtle point is descent from Z to Z_s by the ramified covering map π_s. What happens there is that the intersection homology down on the flattening Z_s arises from that on the covering Z by removing something. Exactly what should be removed is determined by certain highest degree intersection homology of Z; that is, by top degree coefficients in Kazhdan-Lusztig polynomials.

In the setting of Kazhdan and Lusztig's original work, the orbits Z_o are simply connected, so the local systems involved are all trivial. For general real groups the orbits have fundamental groups of size up to $(\mathbb{Z}/2\mathbb{Z})^{\text{rank}}$ and therefore a wealth of local systems. Keeping track of these local systems under the maps π_s is subtle and was another of my contributions to this mathematics. (One ends up in some cases with inductive formulas not for individual Kazhdan-Lusztig polynomials, but for sums of two polynomials, corresponding to two local systems. Part of the difficulty is to find a way to solve the resulting collection of equations.)

A critical point is that the algorithm needs to know the highest degree coefficients and not just the values of the polynomials at $q = 1$. Even though we are interested (for character theory) only in values of the polynomials at 1, the algorithm does not allow us to *compute* only values at 1.

Once the algorithm has forced us to look at coefficients of the Kazhdan-Lusztig polynomials, it is very natural to ask for representation-theoretic interpretations of those coefficients. There is a great deal to say on this subject. I will mention only that the top degree coefficients mentioned above turn out to be dimensions of Ext^1 groups between irreducible representations.

The Atlas of Lie Groups and Representations

That brings the mathematical story up to about 1985. I'll now turn away from abstract mathematics, toward the story of the `atlas` project.

In 2002, Jeff Adams had the idea of getting computers to make interesting calculations about infinite-dimensional representations of reductive Lie groups: ultimately, he hoped, to calculate unitary representations. Of course as mathematicians we want completely general theorems, and it's by no means clear that there is a finite calculation to find the unitary duals of all reductive groups at once. But the work of Dan Barbasch (for example, his classification of the unitary duals of the complex classical groups) makes it possible to hope that one *can* find a finite description of the unitary duals of all classical Lie groups. The exceptional groups are finite in number, so treating them *is* a finite calculation. That became Jeff's standard for measuring the effectiveness of any piece of representation-theoretic software: could it treat the largest exceptional group E_8?

It was clear that the first problem was to write a program that could work with the Cartan subgroups, maximal compact subgroups, and Weyl groups of any real reductive group. Jeff recruited Fokko du Cloux to do this, and Fokko began to work in 2003. By 2004 his software could handle this structure theory (better than most mathematicians, at least).

The next step was less obvious, but Fokko and Jeff settled on computing Kazhdan-Lusztig polynomials for real groups. Fokko had written the best software in the world to do this for Coxeter groups; the algorithms for real groups are similar in structure (although much more complicated in detail, because of the complications attached to local systems). Theorem 7 above says that knowing these polynomials provides formulas for irreducible characters; it is also a critical step in several computational approaches to classifying unitary representations.

So late in 2004, Fokko began to add to his software an implementation of the Kazhdan-Lusztig algorithm for real groups. The papers in which this algorithm is formulated are extremely dense, and written with no consideration for computational practice. An expert could easily spend many months just to understand the mathematical statements. Fortunately, Jeff Adams had been working on a new formulation of Theorem 6 (parametrizing irreducible representations), growing out of earlier work that he did with Dan Barbasch and me. Jeff's formulation seemed suited to computer implementation; he had been working with Fokko to make it more so.

Over the course of the next year, Fokko understood the Kazhdan-Lusztig algorithm for real groups, recasting it in the language that he and Jeff had developed. He wrote clear and efficient code to implement it. In November of 2005—incredibly soon!—he finished. Very quickly he and Jeff used the software to compute Kazhdan-Lusztig polynomials (and so character tables) for all of the real forms of F_4, E_6, and E_7, and for the non-split form of E_8.

The most complicated of these calculations is for the non-split form of E_8. There are 73,410 distinct (translation families of) irreducible representations, so the character table is a $73,410 \times 73,410$ matrix of integers. The integers are values at $q = 1$ of Kazhdan-Lusztig polynomials. These polynomials have degrees from 0 to 27. Their coefficients are nonnegative integers, of which the largest is 2545. The total number of distinct polynomials appearing (among the three billion or so entries below the diagonal in the matrix) is 10,147,581. Here is the polynomial with largest coefficient:

$$q^{13} + 30q^{12} + 190q^{11}$$
$$+682q^{10} + 1547q^9 + 2364q^8$$
$$+2545q^7 + 2031q^6 + 1237q^5$$
$$+585q^4 + 216q^3 + 60q^2 + 11q + 1$$

It's hard to say what constitutes a "typical" polynomial, but here is the one at the midpoint of the lexicographically ordered list:

$$q^9 + 7q^8 + 13q^7 + 6q^6 + 6q^5$$
$$+14q^4 + 18q^3 + 16q^2 + 7q + 1.$$

Fokko's software will calculate this character table on my laptop in about half an hour, using 1500 megabytes of RAM. (I bought a big memory chip at about the same time as I sold my soul to the silicon devil.)

Among the exceptional groups, that left the split form of E_8.

Warming Up for E_8

How big a computation is the character table for split E_8? Fokko's software told us that there were exactly 453,060 (translation families of) irreducible representations. According to Theorem 6 above, the character table can be described by a square matrix of integers, of size 453,060. The number of entries is therefore about 2×10^{11}, or 200 billion.

Fortunately the matrix is lower triangular, so we only need 100 billion entries.

Unfortunately we need to calculate not the entries directly, but rather the Kazhdan-Lusztig polynomials whose values at 1 are the entries. The degrees of the polynomials are bounded by 31; we expected an average degree of about 20, and therefore a total number of coefficients around 2 trillion.

Fortunately many of the matrix entries are easily seen to be equal. An example is Zuckerman's formula for the character of a finite-dimensional representation, where I said that all the coefficients are ± 1 or 0; this is the last row of the character matrix. In the case of E_8, there are 320,206 nonzero terms in this row. Fokko's software recognizes that all 320,206 of those Kazhdan-Lusztig polynomials are going to be equal and stores only the diagonal entry 1. In general one needs to store only one representative polynomial for each family of "obviously equal" entries. Fokko's software calculated how many such families there were: a bit more than 6 billion. So we were down to storing about 6 billion polynomials with about 120 billion coefficients.

Unfortunately we had no clear idea how big the (nonnegative integer) coefficients of these polynomials could be. In the case of split D_5, the largest is 5. For split E_6, the largest is 27, and for split E_7, it's 3583. This trend was not encouraging; it seemed clear that the coefficients would exceed $65,535 = 2^{16} - 1$, so that they could not be stored in two bytes (sixteen bits) of computer memory. The next practical size is four bytes.

Fortunately Fokko wrote the software to compute with four-byte integers and to test carefully for numeric overflow throughout the computation. If overflow happened, the plan was to switch to eight-byte integers and try again.

Unfortunately, 120 billion 4-byte integers require 480 billion bytes of RAM, or 480G. That's a lot of RAM. (The nature of the Kazhdan-Lusztig algorithm, which constantly looks at widely distributed results from earlier in the computation, makes storing results on disk impractically slow. We tried!)

Fortunately, some of the six billion polynomials are zero, and some of them are equal to others "by chance" (that is, for reasons that we have yet to understand). So Fokko wrote the software to store only one copy of each distinct polynomial. He hoped that the number of distinct polynomials might be a few hundred million: so perhaps 6 billion coefficients, requiring 25G of RAM. The indexes keeping track of all these polynomials would also occupy a lot of memory: by the most optimistic estimates perhaps 45G, but quite possibly a lot more.

Unfortunately, we didn't have a computer with even 50G of RAM.

Fortunately computer science is often computer art, and even Fokko's work could be improved. Fokko worked on that constantly during 2006, and Marc van Leeuwen began to make serious contributions as well. The two of them rearranged the indexes and the code in some extraordinarily clever ways.

Unfortunately, tests running partway through the E_8 calculation (done mostly by Birne Binegar) revealed that Fokko's first hopes about the number of distinct polynomials were too optimistic. Even an optimistic reading of Birne's tests suggested more like 800 million distinct polynomials, meaning perhaps 60G or more to hold the coefficients.

Fortunately Dan Barbasch is now chair of the math department at Cornell, and in September of 2006 he managed to gain access to a machine with 128G of RAM and 128G of swap space. He used it to run the E_8 computation to the end. The fact that Fokko's overflow tests were not set off showed that all the coefficients really fit in four bytes.

Unfortunately he had no reasonable way to write the results to disk, so they disappeared. (Fokko's software was written to produce output in human-readable form. In the case of E_8, his output for the character table would have consisted of about fifty billion lines (one for each nonzero entry in the character table) averaging about 80 characters. As a disk file this would have been several terabytes.) Also unfortunately, Dan didn't have the improvements that Fokko and Marc had made to the code: Dan's computation used 224G of memory (half of it swap space). Because of the use of swap space, it took twelve days to finish.

Fortunately, by November of 2006, Fokko and Marc had trimmed memory use in the code a great deal. Through the persistence of Birne Binegar, and the generosity of number theorist William Stein, the `atlas` group got access to William Stein's computer `sage` at the University of Washington (with 64G of RAM and 75G of swap). On this machine we could finally do some large fraction of the E_8 character table computation. By late November, we believed that we could finish E_8 with about 150G of RAM.

Unfortunately, 150G is just a little more than `sage` has, even with swap.

(Not) Buying a Really Big Computer

Birne Binegar and Jeff Adams suggested that we start looking seriously at applying for an NSF grant to buy a machine with perhaps 256G of RAM: something that might cost US$150,000 or more. I asked a number of mathematicians whether they might be able to make use of such a computer.

Noam Elkies had a fascinating reply. First he explained some theoretical limits on the computations that could use a lot of memory.

> A computation that actually uses N bytes of storage must take time *at least* N. But once N gets as large as 256GB it might not be feasible to spend much *more* than N time: $N \cdot \log(N)$ or $N \cdot \log^2(N)$ is certainly OK (e.g., fast integer or polynomial arithmetic, and other applications of the Fast Fourier Transform; also solving $f(x) = f(x')$ by sorting N values of $f(x)$ and finding a consecutive match); maybe also $N^{3/2}$ (e.g., linear algebra with dense matrices of size $N^{1/2}$, or computing the first N coefficients of modular forms such as Δ without fast arithmetic); probably not N^2. So there might not be all that much room for making use of such a huge machine…

He went on to ask

> Is it clear that the E_8 computation cannot fit into "only" 128 or 64GB?

I explained the demands of polynomial storage:

> We know that the polynomial coefficients can exceed 2^{16} (by computation), and we hope that they don't exceed 2^{32}. Each polynomial is stored as a vector of 32-bit integers, of size exactly equal to its degree plus one. Assuming an average degree of 19, that's 80 bytes per polynomial.

On November 30, Noam replied

> Well 2^{32} is less than the product of the ten odd primes less than 2^5, so unless the computation requires divisions by numbers other than 2 you could reduce this from 80 bytes to something like $(5/32) \cdot 80 = 12.5$ bytes, at the cost of running the computation 9 times (counting once for mod $3 \cdot 5$).

In other words, we needed to stop thinking about intersection cohomology for a while and use the Chinese Remainder Theorem. Noam's suggestion was to compute the Kazhdan-Lusztig polynomials modulo m for several small values of m and to store the results to disk. A second program could (for each polynomial) read in these several mod m reductions; apply the Chinese Remainder Theorem to compute the polynomial modulo the least common multiple of all the moduli; and write the result to disk. This second program would need to have only a handful of polynomials in RAM at the same time, so it could run on a much smaller computer.

I started to compose a reply explaining why modular reduction of the Kazhdan-Lusztig algorithm doesn't work, but I had to throw it away: the algorithm works perfectly over $\mathbb{Z}/m\mathbb{Z}$. Fokko's code was beautifully compartmentalized, and Marc van Leeuwen is amazing, so by December 4 we were getting character table entries mod m for any m up to 256. In these calculations, we needed just one byte of memory for each polynomial coefficient. A billion polynomials of degree 20 could live in 20G of RAM.

Computing Characters Mod m

On December 6 Marc's modifications of Fokko's code on `sage` computed about three-fourths of the entries in the E_8 character table mod 251, finding almost 700 million distinct polynomials and using 108G of memory. Since 90% of that was indexes, working on their structure became worthwhile. Marc redesigned the indexes rather completely (replacing 8-byte pointers by 4-byte counters several billion times, for example). In the end he reduced the size of the indexes to about 35G; they would have required more than 100G for the original code. He also added code to output the answers to (small machine-readable) disk files.

Meanwhile Birne Binegar ran various versions of the code on `sage`. Among other things he established for certain that there were more than one billion distinct polynomials.

Early on December 19, Marc's modified code began a computation for E_8 mod 251, with the possibility of actually writing the result usefully at the end. Essentially it worked, finishing the computation in about 17 hours. From diagnostic output of the software, we learned that there were exactly 1,181,642,979 distinct Kazhdan-Lusztig polynomials mod 251. (That turned out to be the number over $\mathbb{Z}$ as well.) The calculation used only 65G of memory; the improvement over 108G on December 6 was because of Marc's redesigned indexing system.

But writing the answer to disk took two days. Marc and I went over Marc's output code to see why. We figured it out, and Marc improved the speed. But we found at the same time a bug: he wrote `size()` in one line where he meant `capacity()`. The result was that, even though the polynomials were all correctly written to disk, the index files (explaining which polynomial was stored where) were missing something like half their contents.

Marc fixed things instantly, and on Thursday evening December 21 we started a calculation mod 256 on `sage`. This computed 452,174 out of 453,060 rows of the character table in 14 hours, then `sage` crashed. We tried again starting late Friday afternoon, and actually finished with good output: the character table mod 256 was written to disk! Because we used multi-threading to speed up the computation, this run took just eleven hours.

On Saturday December 23 we started a calculation mod 255. This time `sage` crashed a third of the way through the computation. There was no one physically present to reboot it (apparently some kind of holiday in Seattle) so we retired for a bit (still having mod 256 as our only good output files).

Meanwhile Marc van Leeuwen had written code to combine Kazhdan-Lusztig polynomials from several moduli $m1, m2, \ldots$ into Kazhdan-Lusztig polynomials modulo $\mathrm{lcm}(m1, m2, \ldots)$. We tested the code on the first hundred million entries of the E_8 character table modulo 253, 255, and 256 (which we could calculate on smaller computers than `sage`), and it worked fine. When `sage` came back up on December 26, we got a character table mod 255 written to disk. At 1 a.m. on December 27, we started a run mod 253. About halfway through, `sage` crashed.

The experts I consulted assured me that the `atlas` software couldn't possibly be crashing `sage`. My own opinions about the causes of the crashes wavered between black helicopters from the NSA and Sasquatch. We resolved to keep our hands off `sage` until we were older and wiser: say for a year.

On Wednesday January 3 we were all one year older, which made perhaps thirty years of additional wisdom counting all the `atlas` people. This factor of thirty seemed like a suitable margin of safety, so that afternoon we started another computation mod 253. This finished in twelve hours.

The Chinese Remainder Calculation

By 4 a.m. Thursday January 4th we had output for three moduli (253, 255, and 256) with least common multiple 16,515,840: bigger (we had some hope) than all the coefficients of the Kazhdan-Lusztig polynomials. Marc van Leeuwen took unfair advantage of the time difference in Europe to start running his Chinese Remainder Theorem utility on the results. Its first task was to correlate the indices of the three output files, to determine which (of 1.1 billion) polynomials mod 253 corresponded to which mod 255. That finished in nine hours.

At that point we encountered another speed problem. The first version of Marc's software had a counter displaying the number of the polynomial to which the Chinese Remainder Theorem was being applied, to allow for monitoring progress.

Since there are more than a billion polynomials, this meant writing several billion characters to the display. It turns out that takes a (very) long time. So we started over on Friday morning January 5, with a counter that updates only every 4096 polynomials. Everything went nicely until `sage` crashed.

William Stein identified `sage`'s problem as a flaky hard drive. The situation for `atlas` was this: sitting on `sage`'s flaky hard drive were 100 gigabytes of output files, the Kazhdan-Lusztig polynomials modulo 253, 255, and 256 for E_8. Each of these files represented something like twelve hours of (multi-threaded) computation; the hundreds of hours of *unsuccessful* computations made them feel a great deal more valuable.

William Stein replaced the bad hard drive with a good one, on which he had made daily backups of all the work on `sage`. He did this more or less instantly: we still had our data files. I was already deeply indebted to his generosity in allowing the `atlas` group access to `sage`, but this raised him even higher in my esteem.

We restarted the Chinese Remainder calculation late Friday afternoon January 5.

Early Saturday morning, the disk file of polynomial coefficients mod 16515840 had grown to be about 7 billion bytes larger than it was supposed to be. Since it was a day with a y in it, I assumed that Marc van Leeuwen would be working. I asked him to find out what was wrong. He was visiting his family in the Netherlands for the weekend and had extremely limited access to the Internet.

The bug was (to my eyes) unbelievably subtle. Since the number of polynomials is about a billion, Marc's code represented the index of a polynomial by a 4-byte integer (perfectly good up to 4,294,967,295). At some point this integer needs to be multiplied by 5 (the number of bytes in the index entry for one polynomial); the result is put into an 8-byte integer, where it fits nicely. But when the polynomial number exceeds 858,993,459, and the multiplication is done in four bytes, it overflows. The result was that the code worked perfectly in any reasonable test (like the one we ran with a hundred million polynomials).

To complicate Marc's task further, the bug was not present in his most recent version of the code; what I was running on `sage` was a couple of days older.

So he was looking for a subtle bug that wasn't there, without Internet access to the machine where the problem occurred. It took him almost twenty hours to find and fix it (here of course I assume that he neither slept nor ate nor spoke to his family).

Marc's analysis showed that the bug came into play only around polynomial number 858 million; so all the coefficients (modulo 16,515,840) calculated before that were correct. The largest of these coefficients was 11,808,808, at polynomial number 818,553,156. (That is the polynomial displayed at the beginning of this article.) I was convinced that we'd find larger coefficients among the 350 million polynomials that were not correctly evaluated the first time and that we'd need a larger modulus than 16,515,840 to get them all.

So at 6 a.m. on Sunday January 7th I was able to restart Marc's current (and correct) Chinese Remainder Theorem utility, this time adding modulus 251. Of course nothing went wrong (because what could go wrong?), and the last polynomial was written to disk just before 9 a.m. Eastern time on Monday January 8.

What Next?

Sixty gigabytes is too much information to look at, even for nineteen mathematicians working in seamless harmony.[1] Here are some of the "clean and simple results" that I promised in the introduction. Attached to any representation are many beautiful geometric invariants. Knowledge of the character table allows us to compute some of them. We have computed the *Gelfand-Kirillov dimension* of each irreducible representation of E_8. This is an integer between 0 and 120 that measures how infinite-dimensional the representation is. Finite-dimensional representations are those of GK dimension 0, and *generic* representations (in a technical sense coming from the theory of automorphic forms) are those of GK dimension 120. The finite-dimensional representations were identified by Cartan and Weyl (Theorem 2 above) around 1930, and the generic representations by Kostant in 1978 (in both cases for all real reductive groups).

Now we can say for E_8 exactly what happens between these extremes. For instance, of the 453,060 (translation families of) representations we studied, there are exactly 392 of GK dimension 57. We can say which ones they are. (I chose 57 because it's the smallest possible dimension of a nontrivial homogeneous space Z_{57} for E_8. These 392 families of representations appear in sections of vector bundles over Z_{57}.) In the same way we can identify James Arthur's *special unipotent representations*, which conjecturally play a fundamental role in the theory of automorphic forms. There are 111 of these among the 453,060 representations.

In the longer term, our goal is to determine completely the unitary irreducible representations for the exceptional Lie groups. Our hope is that knowledge of the character table will allow us to make a computation of these unitary representations. Armed with a list of unitary representations, we can try to explain (most of) it using the Kirillov-Kostant orbit method, or Langlands' ideas about

[1] *This is a theoretical assertion. I have no practical experience with a team of nineteen mathematicians working in seamless harmony.*

functoriality, or perhaps even something entirely new.

Despite the length of this article, I have left out a great deal. I have said nothing about the meetings of the `atlas` group, where the insights of all of the mathematicians involved contributed to the shape of the software that Fokko was writing, and to our evolving understanding of the mathematics beneath it.

The greatest omission is the personal story of Fokko du Cloux. He was diagnosed with ALS (a progressive neurological disease) just after finishing the Kazhdan-Lusztig computation software in November of 2005. By February 2006 he had little use of his hands, and by May he was entirely paralyzed below his neck. But he continued to share his skills and insights into the mathematics and the programming—and his great joy at meeting a formidable mathematical challenge—with Jeff Adams and with Marc van Leeuwen and with me, until his death on November 10, 2006.

The `atlas` has introduced me to great mathematicians I barely knew, like Marc van Leeuwen and John Stembridge, and it has shown entirely new depths in people I knew well, like Jeff Adams and Dan Barbasch. So it's a very high bar…but what has been best of all, mathematically and personally, has been spending time with Fokko. It's *still* the best: every minute I spend on this project is a chance to think about him, and that's always good for a smile.

So thank you to Fokko, who did this all by himself.

Thank you to everyone who helped him—Marc van Leeuwen did more of that than anybody, but there were a lot of indispensable people.

I haven't had a *boss* since I worked in a lumberyard in the summer of 1972, until Jeff Adams. Thank you, boss!

I hope to be back here when we have some unitary representations to share.

Members of the Atlas of Lie Groups and Representations

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

We are very grateful to Brian Conrey and the American Institute of Mathematics, which provided financial support and a wonderful mathematical meeting place from the beginning of the `atlas` project; and to the National Science Foundation, which has provided support through FRG grant 0554278.

—D.V.