Computing coefficients of modular forms

(Work in progress; extension of results of Couveignes, Edixhoven et al.)

Peter Bruin

Mathematisch Instituut, Universiteit Leiden

Théorie des nombres et applications

CIRM, Luminy, 30 November 2009

Introduction

Let k and n be positive integers, and let f be a modular form of weight k for $\Gamma_1(n)$, with q-expansion

$$f = \sum_{m \geq 0} a_m(f) q^m.$$

It is known that f is determined by the $a_m(f)$ for

$$m \leq \frac{k}{12} [\mathrm{SL}_2(\mathbf{Z}) : \{\pm 1\}\Gamma_1(n)].$$

Question: given these $a_m(f)$, is it possible to efficiently compute $a_m(f)$ for large m?

This only seems reasonable to ask when given the factorisation of m: the recurrence relations for the $a_m(f)$ suggest that one could otherwise factor products of two large prime factors efficiently.

We may assume f is a Hecke eigenform, normalised such that $a_1(f) = 1$.

Theorem 1 (tentative for $n > 1$): There is a (probabilistic) algorithm that, given positive integers k and n with n square-free, a normalised eigenform f of weight k for $\Gamma_1(n)$, and an integer $m > 0$ in factored form, computes $a_m(f)$. If the generalised Riemann hypothesis for number fields is true, the algorithm runs in time polynomial in k, n and $\log m$.

For $n = 1$: proved by J.-M. Couveignes, S. J. Edixhoven, R. de Jong and F. Merkl (preprint, 2006/2009; to appear in the *Ann. Math. Studies* series).

For $n > 1$: work in progress, to appear in my thesis (2010).

Note: Our algorithm runs in time polynomial time in the input size, whereas existing algorithms (modular symbols) are exponential in $\log m$.

By the recurrence relation expressing $a_m(f)$ in the $a_p(f)$ for $p \mid m$ prime, we are reduced to the problem of computing $a_p(f)$ for prime numbers p.

Write $\mathbf{Q}(f)$ for the number field generated by the $a_m(f)$. By Deligne's bound $|\sigma(a_p(f))| \leq 2p^{(k-1)/2}$ for all $\sigma\colon \mathbf{Q}(f) \to \mathbf{C}$, it suffices to compute $a_p(f)$ modulo sufficiently many small primes λ of $\mathbf{Q}(f)$.

Remark: We need the generalised Riemann hypothesis to ensure the existence of a sufficient supply of such λ, uniformly in $\mathbf{Q}(f)$.

Theorem 2 (tentative for $n > 1$): There exists a (probabilistic) algorithm that, given positive integers k and n with n square-free, a normalised eigenform f over a finite field $\mathbf{F}$ and a prime number p, computes $a_p(f)$ in time polynomial in k, n and $\log p$.

The strategy for computing $a_p(f)$ for an eigenform f over a finite field $\mathbf{F}$ is to compute the Galois representation associated to f.

Let l be the characteristic of $\mathbf{F}$. There exists a unique semi-simple continuous representation

$$\rho_f : \mathrm{Gal}(\overline{\mathbf{Q}}/\mathbf{Q}) \to \mathrm{GL}_2(\mathbf{F})$$

that is unramified outside nl and such that the Frobenius conjugacy class at a prime $p \nmid nl$ has characteristic polynomial

$$t^2 - a_p(f)t + \epsilon(p)p^{k-1} \in \mathbf{F}[t].$$

In particular, $a_p(f)$ is the trace of a Frobenius at p under ρ_f.

What we want to compute

Let E_f be the finite Galois extension of $\mathbf{Q}$ such that ρ_f factors as

$$\rho_f \colon \mathrm{Gal}(\overline{\mathbf{Q}}/\mathbf{Q}) \twoheadrightarrow \mathrm{Gal}(E_f/\mathbf{Q}) \rightarrowtail \mathrm{GL}_2(\mathbf{F}).$$

Then by computing ρ_f we mean producing the following data:

- the multiplication table of E_f with respect to some $\mathbf{Q}$-basis $(b_1, \ldots, b_r)$ of E_f;

- for every element $\sigma \in \mathrm{Gal}(E_f/\mathbf{Q})$, the matrix of σ with respect to the basis $(b_1, \ldots, b_r)$ and the element $\rho_f(\sigma) \in \mathrm{GL}_2(\mathbf{F})$.

If ρ_f is reducible, then it is associated to an Eisenstein series and is easy to compute.

From now on we assume that $\rho_f\colon \mathrm{Gal}(\overline{\mathbf{Q}}/\mathbf{Q}) \to \mathrm{GL}_2(\mathbf{F})$ is irreducible. After twisting ρ_f by a power of the cyclotomic character, we may assume moreover that

$$2 \leq k \leq l+1.$$

Finally we may assume that $\mathbf{F}$ is generated by the $a_m(f)$.

Notation:

$$n' = \begin{cases} n & \text{if } k=2; \\ nl & \text{if } k>2; \end{cases}$$

$$\mathrm{X}_1(n') = \text{modular curve for } \Gamma_1(n')\text{-structures};$$

$$\mathrm{J}_1(n') = \text{Jacobian of } \mathrm{X}_1(n');$$

$$g = \mathrm{genus}(\mathrm{X}_1(n')) = \dim(\mathrm{J}_1(n')).$$

Let $\mathbf{T}_1(n') \subseteq \mathrm{End}\,\mathrm{J}_1(n')$ denote the Hecke algebra. By the work of various people (Mazur, Ribet, Gross, ...) there is a surjective homomorphism

$$\begin{aligned} \mathbf{T}_1(n') &\to \mathbf{F} \\ T_m &\mapsto a_m(f). \end{aligned}$$

Let $\mathfrak{m} \subset \mathbf{T}_1(n')$ be its kernel. Then the $\mathbf{F}[\mathrm{Gal}(\overline{\mathbf{Q}}/\mathbf{Q})]$-module

$$\mathrm{J}_1(n')[\mathfrak{m}](\overline{\mathbf{Q}})$$

is non-zero and 'usually' isomorphic to ρ_f (in general it has a composition chain consisting of copies of ρ_f).

Strategy for computing Galois representations

To find ρ_f, we are going to explicitly compute the $\mathbf{F}$-vector space scheme $\mathrm{J}_1(n')[\mathfrak{m}]$ over $\mathbf{Q}$. We do this by choosing a suitable closed immersion

$$\iota\colon \mathrm{J}_1(n')[\mathfrak{m}] \rightarrowtail \mathbf{A}^1_{\mathbf{Q}}.$$

The image of ι is defined by some non-zero polynomial $P_\iota \in \mathbf{Q}[x]$.

By "explicitly computing $\mathrm{J}_1(n')[\mathfrak{m}]$" we mean computing P_ι together with a collection of ring homomorphisms defining the $\mathbf{F}$-vector space scheme structure on $\operatorname{Spec} \mathbf{Q}[x]/(P_\iota)$.

From these data we can compute ρ_f by standard methods (mostly factorisation of polynomials over $\mathbf{Q}$).

Choosing a suitable map

Fix a point $O \in \mathrm{X}_1(n')(\mathbf{Q})$. Assume for simplicity that the Abel–Jacobi map

$$\begin{aligned} \mathrm{Sym}^g \mathrm{X}_1(n') &\twoheadrightarrow \mathrm{J}_1(n') \\ D &\mapsto [D - gO] \end{aligned}$$

is an isomorphism above $\mathrm{J}_1(n')[\mathfrak{m}]$. We choose a rational function

$$\psi \colon \mathrm{X}_1(n') \to \mathbf{P}^1(\mathbf{Q})$$

(e.g. a quotient of two modular forms of the same weight) Then we obtain a map

$$\psi_* \colon \mathrm{Sym}^g \mathrm{X}_1(n') \to \mathrm{Sym}^g \mathbf{P}^1_{\mathbf{Q}} \cong \mathbf{P}^g_{\mathbf{Q}}.$$

We choose ψ such that ψ_* is a closed immersion on the inverse image of $\mathrm{J}_1(n')[\mathfrak{m}]$ under the Abel–Jacobi map.

Choosing a suitable map

We next choose a suitable rational map

$$\lambda\colon \mathbf{P}^{g}_{\mathbf{Q}} \dashrightarrow \mathbf{A}^{1}_{\mathbf{Q}} \subset \mathbf{P}^{1}_{\mathbf{Q}}$$

that is a quotient of linear forms. We define our closed immersion

$$\iota\colon \mathrm{J}_1(n')[\mathfrak{m}] \rightarrowtail \mathbf{A}^{1}_{\mathbf{Q}}$$

as the arrow making the diagram

$$\begin{array}{ccccc}
\mathrm{Sym}^{g}\,\mathrm{X}_1(n') & \twoheadrightarrow & \mathrm{J}_1(n') & \supset & \mathrm{J}_1(n')[\mathfrak{m}] \\
{\scriptstyle\psi_*}\downarrow & & & & \\
\mathrm{Sym}^{g}\,\mathbf{P}^{1}_{\mathbf{Q}} & \xrightarrow{\sim} & \mathbf{P}^{g}_{\mathbf{Q}} & \overset{\lambda}{\dashrightarrow} & \mathbf{A}^{1}_{\mathbf{Q}}
\end{array}$$

commutative.

How to compute P_ι

Recall that we want to compute (among other things) the polynomial P_ι defining the image of the closed immersion

$$\iota\colon \mathrm{J}_1(n')[\mathfrak{m}] \rightarrowtail \mathbf{A}^1_{\mathbf{Q}}.$$

To compute P_ι, we use numerical approximation together with a bound on the heights of the coefficients of P_ι,

The polynomial P_ι can be approximated either using computations over the complex numbers (deterministically) or modulo many small prime numbers (probabilistically).

How to compute P_ι modulo prime numbers

For computing P_ι modulo a prime number p, one needs to be able to compute in the Jacobian of $\mathrm{X}_1(n')$ over finite fields of characteristic p: picking random elements, computing the Frobenius map, evaluating Hecke operators, etc.

For $n = 1$, one can use a (singular) plane model of $\mathrm{X}_1(5l)$ over $\mathbf{F}_p$ with singularities and apply algorithms by Couveignes for computing in the Jacobian of such a curve.

For $n \geq 1$, one can use the projective embedding of $\mathrm{X}_1(n')$ defined by modular forms of weight 2 and use algorithms of K. Khuri-Makdisi, Couveignes (adapted to this situation), C. Diem and myself.

For $n \geq 5$, the line bundle $\mathcal{L} = \omega^2$ of modular forms of weight 2 on $\mathrm{X}_1(n)$ over a field K (of characteristic not dividing n) gives a closed immersion

$$\mathrm{X}_1(n)_K \rightarrowtail \mathbf{P}\Gamma(\mathrm{X}_1(n), \mathcal{L}).$$

An effective divisor D with $\deg D = \deg \mathcal{L}$ can be represented as the subspace $\Gamma(\mathrm{X}_1(n), \mathcal{L}^2(-D))$. To such a D we associate the point $[\mathcal{L}(-D)]$ of $\mathrm{J}_1(n)(K)$.

Khuri-Makdisi has developed algorithms for computing with elements of the Jacobian represented in this way. Based in part on work of Couveignes and of Diem, I have shown that if K is finite, one can also compute Frobenius maps, Hecke correspondences, Kummer maps and Frey–Rück pairings. These can be used to compute $\mathrm{J}_1(n')[\mathfrak{m}]$ (i.e. compute P_ι) modulo prime numbers.

Height bounds

We use Arakelov intersection theory on the arithmetic surface $\mathrm{X}_1(n')$ to find bounds for the heights of the coefficients of the polynomial P_ι. Intersection numbers at infinite places can be expressed in terms of *canonical Green functions* of the Riemann surfaces $\mathrm{X}_1(n')(\mathbf{C})$.

We need to study the semi-stable reduction of $\mathrm{X}_1(n')$, and find bounds for canonical Green functions and for sup-norms of modular forms. Work of J. Jorgenson and J. Kramer, using spectral theory of automorphic forms for Fuchsian groups, implies that the latter quantities are bounded independently of n'.

Using methods similar to that of Jorgenson and Kramer, I have found bounds that could fairly easily be made explicit. These methods can be interpreted as based on the fact that the Green function is the constant term of the resolvent kernel of the Laplace operator.

Application: explicit realisations of Galois groups

The complex analytic method for computing modular Galois representations has been used by J. Bosman to compute various explicit polynomials over $\mathbf{Q}$ whose splitting fields have interesting Galois groups, such as $\mathrm{SL}_2(\mathbf{F}_{16})$ and $\mathrm{PSL}_2(\mathbf{F}_{49})$.

The algorithm is so far only practical in small cases. Instead of using explicit height bounds, Bosman verified the results using the fact that Serre's conjecture is true.

We expect that combining these complex analytic algorithms with the algorithms over finite fields (to be implemented) can be used to compute explicit realisations of more Galois groups.

Application: representation numbers of lattices

Let L be an even integral lattice of rank k, write

$$r_L(m) = \#\{x \in L \mid (x,x) = m\},$$

and let

$$\theta_L = \sum_{m \geq 0} r_L(2m) q^m \in \mathbf{Z}[q]$$

be the θ-series of L. Then θ_L is the q-expansion of a modular form of weight $k/2$ for $\Gamma_1(n)$, where n is the level of L.

Example: the Leech lattice (rank 24 and level 1). We know how to write its θ-series as a linear combination of E_{12} and Δ. Its representation numbers can therefore be computed by the work of Couveignes, Edixhoven, de Jong and Merkl.

Sums of squares

For $L = \mathbf{Z}^k$ with $(x, y) = 2\sum_{i=1}^{k} x_i y_i$, the θ-series $\theta_{\mathbf{Z}^k}$ is a modular form of weight $k/2$ for the group $\Gamma_1(4)$. From the identity

$$\theta_{\mathbf{Z}^k} = (\theta_{\mathbf{Z}})^k = (1 + 2q + 2q^4 + 2q^9 + \cdots)^k$$

we can quickly compute the first few coefficients of $\theta_{\mathbf{Z}^k}$. This has the following application to representing integers as sums of squares.

Expected theorem 3: There exists a (probabilistic) algorithm that, given an *even* integer $k \geq 0$ and an integer $m \geq 1$ in factored form, computes the number of ways in which m can be written as a sum of k squares, and that runs in time polynomial in k and $\log m$ under the generalised Riemann hypothesis for number fields.

A question about lattices

To compute coefficients of θ-series of a lattice in this way, one needs to input the first few coefficients of the θ-series into the algorithm.

It is known that finding the length of the shortest vector in a lattice is already a hard problem, but the proof of this does not seem to involve the level of the lattice. This raises the following question about counting short vectors in lattices.

Question: Does there exist an algorithm that, given a lattice L of rank k and level n and an integer m, computes $r_L(m)$ in time polynomial in k, n and m?