Computing modular Galois representations

An implementation of the finite field approach

Peter Bruin

Universiteit Leiden

Nederlandse Organisatie voor Wetenschappelijk Onderzoek

Atelier PARI/GP 2015

Bordeaux, 15 January 2015

Let X be a (smooth, complete, geometrically connected) curve of genus g over a finite field k.

Consider the Picard group

$$\begin{aligned}\operatorname{Pic}^0 X &= \{\text{divisors of degree 0}\}/\{\text{principal divisors}\}\\ &= \{\text{line bundles of degree 0}\}/\cong.\end{aligned}$$

This is also the group of k-rational points of the Jacobian variety J of X, which is an Abelian variety of dimension g over k.

Goal: compute efficiently in $\operatorname{Pic}^0 X = J(k)$ if g is large.

- Represent X, divisors on X and elements of $J(k)$
- Perform group operations in $J(k)$
- Pick random elements of $X(k)$ and $J(k)$
- Evaluate the Frobenius automorphism of $J(k')$ for finite extensions k'/k
- Evaluate Kummer maps $J(k)/nJ(k) \to J(k)[n]$
- Evaluate Frey–Rück (Tate) pairings $J(k)[n] \times J(k)/nJ(k) \to \mu_n(k)$

Using these operations, one can find a basis of $J(k)[l]$, with $l \neq \operatorname{char} k$ a prime number, in time polynomial in g, $\log \#k$ and l, provided we know the zeta function of X.

Various approaches have been developed for computing in $\mathrm{Pic}^0 X$, based on different ways to represent X and divisors on it:

(1) as a (ramified) covering of $\mathbf{P}^1$;

(2) as a curve in $\mathbf{P}^2$ (necessarily singular for general X);

(3) by an embedding into a projective space of relatively high dimension.

We will only consider the third approach. There exist efficient algorithms, due in the first place to K. Khuri-Makdisi (2004) and extended in my thesis (2010).

Khuri-Makdisi's methods are tailored for modular curves. For example, if $n \geq 5$ the required representation of $\mathrm{X}_1(n)$ over k can be obtained from a basis of q-expansions for the space of modular forms of weight 2 for $\Gamma_1(n)$ over k.

Following Khuri-Makdisi, we choose a line bundle $\mathcal{L}$ on X with

$$\deg \mathcal{L} \geq 2g + 1.$$

Let $\Gamma(X, \mathcal{L})$ denote its space of global sections. (For us, this is just a space of modular forms.)

Besides $\Gamma(X, \mathcal{L})$, we store the spaces $\Gamma(X, \mathcal{L}^i)$ for $1 \leq i \leq 7$ and the multiplication maps between them.

This gives a representation of X in terms of linear algebra.

Remark: We never need to write down equations (although they are implicit in the data and can be extracted if needed).

We represent points on X by hyperplanes in $\Gamma(X, \mathcal{L})$: to $x \in X$ we associate the subspace

$$V_x = \{f \in \Gamma(X, \mathcal{L}) \mid f(x) = 0\} \subset \Gamma(X, \mathcal{L}).$$

This construction gives a projective embedding

$$\begin{aligned} X &\rightarrowtail \mathbf{P}\Gamma(X, \mathcal{L}) \\ x &\mapsto V_x. \end{aligned}$$

Remark: after choosing a basis $(f_0, \ldots, f_n)$ of $\Gamma(X, \mathcal{L})$, this embedding looks like

$$\begin{aligned} X &\rightarrowtail \mathbf{P}^n_k \\ x &\mapsto (f_0(x) : f_1(x) : \ldots : f_n(x)). \end{aligned}$$

However, we prefer to represent x by the hyperplane $V_x \subset \Gamma(X, \mathcal{L})$ rather than by a vector.

Similarly, we represent effective divisors D on X by subspaces of the form

$$\begin{aligned} V_D^i &= \Gamma(X, \mathcal{L}^i(-D)) \\ &= \{f \in \Gamma(X, \mathcal{L}^i) \mid f \text{ vanishes on } D\} \end{aligned}$$

for i large enough such that $\deg \mathcal{L}^i(-D) \geq 2g + 1$.

Remark: This comes down to embedding the d-th symmetric power $\operatorname{Sym}^d X$ (variety of effective divisors of degree d) into the Grassmannian variety of subspaces of codimension d in $\Gamma(X, \mathcal{L}^i)$:

$$\operatorname{Sym}^d X \rightarrowtail \operatorname{Gr}^d \Gamma(X, \mathcal{L}^i).$$

Khuri-Makdisi's algorithms are based on two fundamental results:

Lemma (multiplication): We can compute V^{i+j}_{D+E} from V^i_D and V^j_E by

$$\begin{aligned} V^{i+j}_{D+E} &= V^i_D \cdot V^j_E \\ &= \operatorname{span}\{vw \mid v \in V^i_D, w \in V^j_E\} \subset \Gamma(X, \mathcal{L}^{i+j}) \end{aligned}$$

Lemma (division): We can compute V^i_D from V^{i+j}_{D+E} and V^j_E by

$$\begin{aligned} V^i_D &= V^{i+j}_{D+E} \div V^j_E \\ &= \{v \in \Gamma(X, \mathcal{L}^i) \mid vV^j_E \subseteq V^{i+j}_{D+E}\}. \end{aligned}$$

These allow us to add and subtract divisors and to test for linear equivalence.

Write

$$d = \deg \mathcal{L} \quad (\geq 2g + 1).$$

Elements of $J(k)$ are represented by effective divisors of degree d as follows:

$$\begin{aligned} \{\text{effective divisors of degree } d \text{ on } X\} &\to J(k) \\ D &\mapsto \text{isomorphism class of } \mathcal{L}(-D). \end{aligned}$$

Addition and negation in J can be built up from $0 \in J(k)$ and the operation

$$\mathrm{addflip}\colon (x, y) \mapsto -x - y.$$

Let $x, y \in J(k)$ be represented by effective divisors D, E. Then $-x-y$ is represented by any effective divisor F such that $\mathcal{L}^3(-D-E-F) \cong \mathcal{O}_X$.

We would like to pick uniformly random elements from the finite sets $X(k)$ and $J(k)$.

Algorithm: Choose a uniformly random hyperplane H in $\mathbf{P}\Gamma(X, \mathcal{L})$. Compute the set $\{x_1, \ldots, x_r\}$ of rational points in $H \cap X$. With probability $r/\deg \mathcal{L}$, pick one of the x_i; else start over.

Using a similar approach, we can pick uniformly random prime divisors on X.

Uniformly random effective divisors of a given degree m can be built up from prime divisors as follows. First select the "decomposition type" (degrees and multiplicities of prime divisors) of a uniformly random effective divisor of degree m using the zeta function of X. Then pick a uniformly random such divisor having this decomposition type.

If k' is a finite extension of k, we can compute the Frobenius map

$$F\colon J(k') \xrightarrow{\sim} J(k')$$

by applying the $\#k$-th power map on matrix entries with respect to k-bases.

If n is coprime to $\operatorname{char} k$ and $J[n]$ is k-rational, we can compute the Kummer isomorphism

$$K\colon J(k)/nJ(k) \xrightarrow{\sim} J(k)[n]$$

coming from Galois cohomology of $0 \to J[n] \to J \xrightarrow{n} J \to 0$. Under the weaker assumption that $k^\times$ contains the n-th roots of unity, we can compute the Frey–Rück (Tate) pairing

$$[\ \ ,\ \ ]_n\colon J(k)/nJ(k) \times J(k)[n] \to \mu_n(k).$$

(Based in part on work of Couveignes, transferred to our setting.)

Application: computing modular Galois representations

Let f be a normalised Hecke eigenform, let K be the number field generated by the coefficients of f, let λ be a finite place of K with residue field $\mathbf{F}_\lambda$. There are associated semi-simple Galois representations

$$\rho_{f,\lambda}\colon \mathrm{Gal}(\bar{\mathbf{Q}}/\mathbf{Q}) \to \mathrm{GL}_2(K_\lambda),$$
$$\bar\rho_{f,\lambda}\colon \mathrm{Gal}(\bar{\mathbf{Q}}/\mathbf{Q}) \to \mathrm{GL}_2(\mathbf{F}_\lambda).$$

We want to compute $\bar\rho_{f,\lambda}$ in the sense of giving a finite Galois extension $L/\mathbf{Q}$ together with an embedding $\mathrm{Gal}(L/\mathbf{Q}) \rightarrowtail \mathrm{GL}_2(\mathbf{F}_\lambda)$.

If $\bar\rho_{f,\lambda}$ is irreducible, then after twisting it occurs in $\mathrm{J}_1(n)[l](\bar{\mathbf{Q}})$ for a suitable n, where $l = \operatorname{char}\lambda$ and $\mathrm{J}_1(n)$ is the Jacobian of $\mathrm{X}_1(n)$. This allows us to reduce the problem of computing $\bar\rho_{f,\lambda}$ as follows: given a maximal ideal $\mathfrak{m}$ of the Hecke algebra acting on $\mathrm{J}_1(n)$, with residue field $\mathbf{F}$, compute $\mathrm{J}_1(n)[\mathfrak{m}]$.

(Project of Couveignes, Edixhoven et al.; “Schoof-like” algorithm.)

We choose a suitable embedding of $\mathbf{Q}$-schemes

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

Then $\operatorname{im}\iota$ is defined by a polynomial over $\mathbf{Q}$; the induced $\mathbf{F}$-vector space scheme structure on $\operatorname{im}\iota$ is also given by polynomials over $\mathbf{Q}$.

Strategy for computing $\mathrm{J}_1(n)[\mathfrak{m}]$: find these polynomials either numerically over $\mathbf{C}$ or modulo p for sufficiently many small prime numbers p, and then reconstruct $\operatorname{im}\iota$ over $\mathbf{Q}$.

Remark: to know how much precision/how many p we need to ensure correctness, one needs a bound on the heights of the coefficients of the polynomials. Such a bound can be derived (with a lot of work) if one chooses ι carefully.

PARI implementation of the finite field approach (work in progress)

The program (ca. 5700 lines of C code using the PARI library) consists of four modules:

• `libpari-extra` – various utility functions: linear algebra, finite algebras, extensions of finite fields (small characteristic, i.e. `Fl` and `Flxq`)

• `modular` – a toy implementation of modular symbols. Hopefully this will soon be switched to the new PARI `ms*` functions (main new ingredient needed: modular symbols for $\Gamma_1(n)$).

• `jacobian` – general curves over finite fields, Jacobians, operations on them (using Khuri-Makdisi's algorithmic representation)

• `modgalrep` – modular curves and Jacobians; Galois representations

Linear algebra (over finite fields) is currently the main bottleneck.