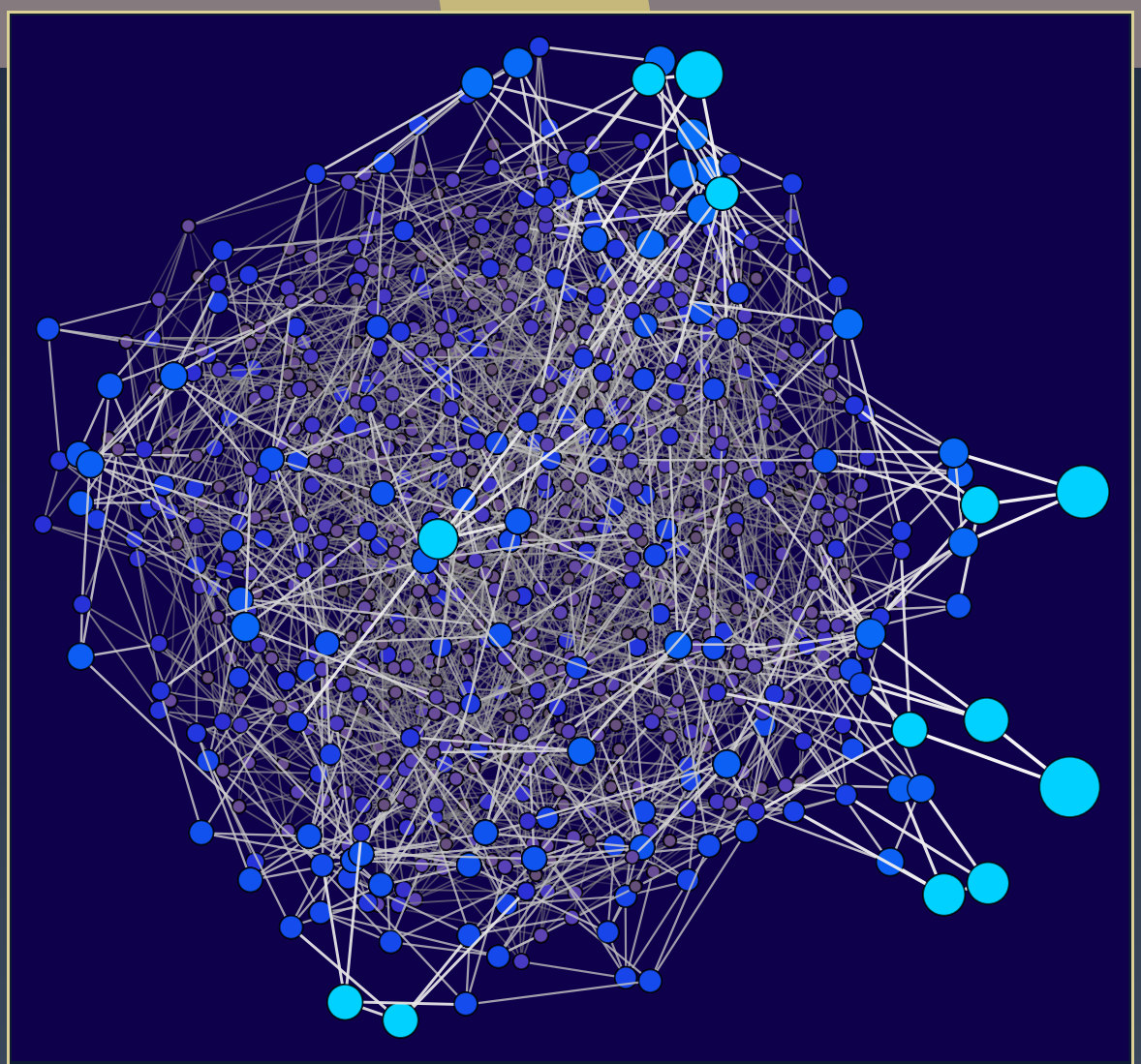
THE OPEN BOOK SERIES 4

ANTS XIV
Proceedings of the Fourteenth Algorithmic Number Theory Symposium

Counting points on superelliptic curves in average polynomial time

Andrew V. Sutherland



msp

THE OPEN BOOK SERIES **4** (2020)
Fourteenth Algorithmic Number Theory Symposium
https://doi.org/10.2140/obs.2020.4.403

Counting points on superelliptic curves in average polynomial time

Andrew V. Sutherland

In memory of Peter L. Montgomery.

We describe the practical implementation of an average polynomial-time algorithm for counting points on superelliptic curves defined over $\mathbb{Q}$ that is substantially faster than previous approaches. Our algorithm takes as input a superelliptic curve $y^m = f(x)$ with $m \geq 2$ and $f \in \mathbb{Z}[x]$ any squarefree polynomial of degree $d \geq 3$, along with a positive integer N. It can compute $\#X(\mathbb{F}_p)$ for all $p \leq N$ not dividing $m\mathrm{lc}(f)\mathrm{disc}(f)$ in time $O(md^3 N \log^3 N \log\log N)$. It achieves this by computing the trace of the Cartier–Manin matrix of reductions of X. We can also compute the Cartier–Manin matrix itself, which determines the p-rank of the Jacobian of X and the numerator of its zeta function modulo p.

1. Introduction

Let X/k by a smooth projective curve of genus $g > 0$ whose function field is defined by an equation of the form

$$y^m = f(x)$$

with $m > 1$ prime to the characteristic p of k and $f \in k[x]$ a squarefree polynomial of degree $d \geq 3$. We shall call such a curve X a superelliptic curve. We note that not all authors require f to be squarefree or $p \nmid m$, while others require d and m to be coprime; our definition follows the convention in [21; 27] and is equivalent to the class of cyclic covers of $\mathbb{P}^1$ considered in [2; 13]. One can compute the genus of X as

$$g = \frac{(d-2)(m-1) + m - \gcd(m,d)}{2} \tag{1}$$

via the Riemann–Hurwitz formula. Well-known examples of superelliptic curves include elliptic curves, hyperelliptic curves, Picard curves, and Fermat curves.

The author was supported by Simons Foundation grant 550033.
MSC2010: primary 11G20; secondary 11M38, 11Y16, 14G10.
Keywords: superelliptic curve, Cartier–Manin matrix, Hasse–Witt matrix, average polynomial-time.

We are primarily interested in $k = \mathbb{Q}$ where X has an associated L-function $L(X, s) = \sum a_n n^{-s}$ that we would like to "compute". For us this means computing the integers a_n for all n up to a bound N that is large enough for us to approximate special values of $L(X, s)$ to high precision, and to compute upper bounds on its analytic rank that we can reasonably expect to be sharp. This requires N to be on the order of the square root of the conductor of the Jacobian of X, and in practice we typically take N to be about 30 times this value.

The fact that $L(X, s)$ is defined by an Euler product implies that it suffices to compute a_n for prime powers $n \le N$. Nearly all of the prime powers $n \le N$ are in fact primes p, so this task is overwhelmingly dominated by the time to compute a_p for primes $p \le N$. Indeed, even if we spend $O(p^e \log^2 p)$ time computing each $a_{p^e} \le N$ with $e > 1$ (which for primes of good reduction can be achieved by naïve point-counting), we will have spent only $O(N \log N)$ time, which is roughly the time it takes just to write down all the a_n for $n \le N$. For primes of good reduction for X, including all $p \nmid m \operatorname{lc}(f) \operatorname{disc}(f)$,[1] we may compute a_p as

$$a_p = p + 1 - \#X(\mathbb{F}_p);$$

in other words, by counting points on the reduction of X modulo p. See [6] for a discussion of how primes of bad reduction may be treated. Alternatively, if one is willing to assume that the Hasse–Weil conjecture for $L(X, s)$ holds, one can use the knowledge of a_n at powers of good primes to determine the a_n at powers of bad primes (and in particular, the primes $p|m$ not treated by [6]) by using the functional equation to rule out all but one possibility; see [3, §5] for a discussion of this approach when $g = 2$.

Another motivation for computing a_p for good primes $p \le N$ is to compute the sequence of normalized Frobenius traces $a_p/\sqrt{p}$ that appear in generalizations of the Sato–Tate conjecture. The moments of this distribution encode certain arithmetic invariants of X, including, for example, the rank of the endomorphism ring of its Jacobian [9, Proposition 1], as well as information about its Sato–Tate group [11; 22]. Indeed, the initial motivation for this work (and its first application) was to compute Sato–Tate distributions for the genus 3 superelliptic curves with $(m, d) \in \{(3, 4), (4, 3), (4, 4)\}$ that arise as smooth plane quartics in the database described in [25] and played a role in the recent classification of Sato–Tate groups of abelian threefolds [12]. The sequence of normalized Frobenius traces can also be used to numerically investigate the error term in the Sato–Tate conjecture, and in particular, predictions regarding its leading constant [7]. The ability to efficiently compute many integer values of a_p also supports investigations of generalizations of the Lang–Trotter conjecture, as well as a recent question of Serre regarding the density of "record" primes, those with $-a_p > 2g\sqrt{p} - 1$ (personal communication, 2019).

The algorithm we present here does more than just compute a_p. Following the approach of [15; 16; 17], which treated the case of hyperelliptic curves, for each good prime p we compute a $g \times g$ matrix A_p giving the action of the Cartier–Manin operator on a basis for the space of regular differentials of the reduction of X modulo p; see Section 2 for details. The matrix A_p is the transpose of the Hasse–Witt

[1] For $m|d$ some good primes may divide $\operatorname{lc}(f)$, but to simplify the presentation we exclude them here.

matrix. Like the Hasse–Witt matrix, it satisfies

$$\det(I - T A_p) \equiv L_p(T) \bmod p,$$

where $L_p(T)$ is the integer polynomial that appears in both the Euler product $L(X, s) = \prod_p L_p(p^{-s})^{-1}$ and the numerator of the zeta function of the reduction of X modulo p:

$$Z_p(T) := \exp\left(\sum_{n\geq 1} \#X(\mathbb{F}_{p^n})\frac{T^n}{n}\right) = \frac{L_p(T)}{(1-T)(1-pT)}.$$

In particular, we have $a_p \equiv \operatorname{tr} A_p \bmod p$, and for $p > 16g^2$ this uniquely determines $a_p \in \mathbb{Z}$, since $|a_p| \leq 2g\sqrt{p}$, by the Weil bounds. The matrix A_p is also of independent interest, since it can be used to compute the p-rank of the reduction of X modulo p, something that cannot be deduced solely from $L_p(T)$.

Our main result is the following theorem, in which $\|f\| = \log \max_i |f_i|$ denotes the logarithmic height of a nonzero integer polynomial $f(x) = \sum_i f_i x^i$.

Theorem 1. *Given a superelliptic curve $X : y^m = f(x)$ with $f \in \mathbb{Z}[x]$ of degree d and $N \in \mathbb{Z}_{>0}$, the algorithm* ComputeCartierManinMatrices *outputs the Cartier–Manin matrices A_p of the reductions of X modulo all primes $p \leq N$ not dividing $m \operatorname{lc}(f) \operatorname{disc}(f)$. If we assume m, d, $\|f\|$ are bounded by $O(\log N)$ the algorithm runs in $O(m^2 d^3 N \log^3 N)$ time using $O(md^2 N)$ space; it can alternatively compute Frobenius traces $a_p \in \mathbb{Z}$ for $p \leq N$ in time $O(md^3 N \log^3 N)$.*

Remark 2. The assumption $m, d, \|f\| = O(\log N)$ ensures that the complexity of multiplying the integer matrices used in the algorithm is dominated by the cost of computing FFT transforms of the matrix entries, which eliminates any dependence on the exponent ω of matrix multiplication; one can replace d^3 with $d^{\omega+1}$ and then remove this assumption. We note that our complexity bound relies on the recently improved $\mathsf{M}(n) = n \log n$ bound on integer multiplication [18]. While the algorithm that achieves this bound is not practical, many FFT-based implementations effectively achieve this growth rate within the feasible range of computation, which for our purposes, is certainly limited to integers that fit in random access memory; see [26, Algorithm 8.25], for example.

We also obtain an algorithm that can be used to compute A_p for a single superelliptic curve $X/\mathbb{F}_p$. The asymptotic complexity is comparable to that achieved in [2] which describes the algorithm that is now implemented in version 9 of Sage. We include this result because it contains several components that are used by the average polynomial-time algorithm we present. We should emphasize that the algorithm in [2] can compute $L_p(T) \bmod p^n$ for any $n \geq 1$, and taking n sufficiently large yields $L_p \in \mathbb{Z}[T]$, whereas we focus solely on the case $n = 1$ (we gain a small but not particularly significant performance advantage in this case).

Theorem 3. *Given a superelliptic curve $X : y^m = f(x)$ with $f \in \mathbb{F}_p[x]$ of degree d, the algorithm* ComputeCartierManinMatrix *is able to compute the Cartier–Manin matrix of X using $O(p^{1/2} m d^2 \log p)$*

space in $O(p^{1/2}m(d^{\omega+1}+d^3\log p)\log p(\log\log p))$ *time, and also using* $O((md+d^2)\log p)$ *space in* $O((p+d)md^2\log p\log\log p)$ *time.*

In the article [2] noted above the authors consider a particular curve

$$X : y^7 = x^3 + 4x^2 + 3x - 1$$

for which they estimate that it would take approximately six months (on a single core) for their algorithm to compute the L-polynomials $L_p(T)$ for all primes $p \le 2^{24}$ of good reduction. This is an improvement over an estimated three years for an earlier algorithm due to Minzlaff [20] that is implemented in Magma. Computing $L_p(T)$ mod p is an easier problem that would likely take about a week or so using the algorithm in [2], based on timings taken using a representative sample of $p \le 2^{24}$. The algorithm we present here can accomplish this task in half an hour, and less than ten minutes if we only compute Frobenius traces.

See Tables 1 and 2 in Section 7 for detailed performance comparisons for various shapes of superelliptic curves.

2. The Cartier operator

For background on differentials of algebraic function fields we refer the reader to [8, §2] and [23, §4]. Let K be a function field of one variable over a perfect field k of characteristic $p > 0$ that we assume is the full field of constants of K. Let Ω_K denote its module of differentials, which we identify with its module of Weil differentials via [23, Definition 4.17] and [23, Remark 4.3.7]. Let $x \in K$ be a separating element, so that $K/k(x)$ is a finite separable extension, and let K^p denote the subfield of p-th powers. Then $(1, x, \dots, x^{p-1})$ is a basis for K as a K^p-vector space, and every $z \in K$ has a unique representation of the form

$$z = z_0^p + z_1^p x + \cdots + z_{p-1}^p x^{p-1},$$

with $z_0, \dots, z_{p-1} \in K^p$, and every rational differential form $\omega = z\,dx$ can be uniquely written in the form

$$\omega = (z_0^p + z_1^p x + \cdots z_{p-1}^p x^{p-1})dx.$$

The (modified) *Cartier operator* $\mathcal{C} : \Omega_K \to \Omega_K$ is then defined by

$$\mathcal{C}(\omega) = z_{p-1}dx.$$

The Cartier operator is uniquely characterized by the following properties:

(1) $\mathcal{C}(\omega_1 + \omega_2) = \mathcal{C}(\omega_1) + \mathcal{C}(\omega_2)$ for all $\omega_1, \omega_2 \in \Omega_K$.

(2) $\mathcal{C}(z^p\omega) = z\,\mathcal{C}(\omega)$ for all $z \in K$ and $\omega \in \Omega_K$.

(3) $\mathcal{C}(dz) = 0$ for all $z \in K$.

(4) $\mathcal{C}(dz/z) = dz/z$ for all $z \in K^\times$.

In particular, it does not depend on our choice of a separating element x. Moreover, it maps regular differentials to regular differentials and thus restricts to an operator on the space

$$\Omega_K(0) = \{\omega \in \Omega_K : \omega = 0 \text{ or } \operatorname{div}(\omega) \geq 0\},$$

which we recall is a k-vector space whose dimension g is equal to (and often used as the definition of) the genus of K; see [23, Example 4.12-17] for these and other standard facts about the Cartier operator.

Definition 4. Let $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_g)$ be a basis for $\Omega_K(0)$ and define $a_{ij} \in k$ via

$$\mathcal{C}(\omega_j) = \sum_{i=1}^{g} a_{ij}\omega_i.$$

The Cartier–Manin matrix of K (with respect to $\boldsymbol{\omega}$) is the matrix $A = [a_{ij}] \in k^{g\times g}$.

If X/k is a smooth projective curve with function field $k(X) = K$, we also call A the Cartier–Manin matrix of X. This matrix is closely related to the Hasse–Witt matrix B of X, which is defined as the matrix of the p-power Frobenius operator acting on $H^1(X, \mathcal{O}_X)$ with respect to some basis. As carefully explained in [1], the matrices A and B can be related via Serre duality, and for a suitable choice of basis one finds that $B = [a_{ij}^p]^{\mathsf{T}}$. In the case of interest to us $k = \mathbb{F}_p$ is a prime field and the Cartier–Manin and Hasse–Witt matrices are simply transposes of each other, and hence have the same rank and characteristic polynomials, but we shall follow the warning/request of [1] and call A the Cartier–Manin matrix, although one can find examples in the literature where A is called the Hasse–Witt matrix (see [1] for a list).

We shall apply the method of Stöhr–Voloch [24] to compute the Cartier–Manin matrix of a smooth projective curve X with function field $K = k(X)$. Let us write K as $k(x)[y]/(F)$, where $x \in X$ is a separating element and y is an integral generator for the finite separable extension $K/k(x)$ with minimal polynomial $F \in k[x][y]$. We now define the differential operator

$$\nabla = \frac{\partial^{2p-2}}{\partial x^{p-1}\partial y^{p-1}}$$

which maps $x^{(i+1)p-1}y^{(j+1)p-1}$ to $x^{ip}y^{jp}$ and annihilates monomials not of this form; it thus defines a semilinear map $\nabla : K \to K^p$. Writing F_y for $\frac{\partial}{\partial y}F \in k[x, y]$, for any $h \in K$ we have the identity

$$\mathcal{C}\left(h\frac{dx}{F_y}\right) = (\nabla(F^{p-1}h))^{1/p}\frac{dx}{F_y}, \tag{2}$$

given by [24, Theorem 1.1]. If we choose a basis for $\Omega_X(0)$ using regular differentials of the form hdx/F_y, we can compute the action of the Cartier operator on this basis via (2). To construct such a basis we shall use differentials of the form

$$\omega_{k\ell} = x^{k-1}y^{\ell-1}\frac{dx}{F_y}, \qquad (k, \ell \geq 1, \quad k + \ell \leq \deg(F) - 1). \tag{3}$$

Writing $F(x,y)^{p-1} = \sum_{i,j} F_{ij}^{p-1} x^i y^j$ (defining $F_{i,j}^{p-1} \in k$ for all $i, j \in \mathbb{Z}$), for $k, \ell \geq 1$ one finds that

$$\nabla\left(\sum_{i,j\geq 0} F_{ij}^{p-1} x^{i+k-1} y^{j+\ell-1}\right) = \sum_{i,j\geq 1} F_{ip-k,\,jp-\ell}^{p-1} x^{(i-1)p} y^{(j-1)p}. \tag{4}$$

Now $F_{ip-k,\,jp-\ell}^{p-1}$ is nonzero only if we have $(i+j)p - (k+\ell) \leq (p-1)\deg(F)$, and $k+\ell \leq \deg(F)-1$, so we can restrict the sum on the RHS to $i+j \leq \deg(F)-1$. From (2) and (4) we obtain

$$\mathcal{C}(\omega_{k\ell}) = \sum_{i,j\geq 1} (F_{ip-k,\,jp-\ell}^{p-1})^{1/p} \omega_{ij}. \tag{5}$$

When X is a smooth plane curve the complete set of ω_{ij} defined in (3) is a basis for $\Omega_K(0)$ and we can read off the entries of the Cartier–Manin matrix for X directly from (5). In general not all of the ω_{ij} necessarily lie in $\Omega_K(0)$, some of them might not be regular, but the subset that do (those corresponding to adjoint polynomials) form a basis for $\Omega_K(0)$; see [14; 24]. In the case of superelliptic curves this subset is given explicitly by Lemma 6.

Definition 5. For $a, b \in \mathbb{Z}$ with $b > 0$, let $a \operatorname{rem} b = a - b\lfloor a/b \rfloor$ denote the unique integer in the interval $[0, b-1] \cap (a + b\mathbb{Z})$.

Lemma 6. *Let k be a perfect field of positive characteristic p, let X/k be a superelliptic curve defined by $F(x,y) = y^m - f(x) = 0$, let $d = \deg f$, and for $i, j \geq 1$ let $\omega_{ij} = x^{i-1}y^{j-1}dx/F_y \in \Omega_K$, where $K = k(x)[y]/(F)$ is the function field of X. Then the set*

$$\boldsymbol{\omega} = \{\omega_{ij} : mi + dj < md\}$$

is a k-basis for $\Omega_K(0)$, with $1 \leq i < d - \lfloor d/m \rfloor$ and $1 \leq j < m - \lfloor m/d \rfloor$. Moreover, if we define

$$d_j = d - \lfloor dj/m \rfloor - 1 \quad \text{and} \quad m_i = m - \lfloor mi/d \rfloor - 1, \tag{6}$$

then the $\omega_{ij} \in \boldsymbol{\omega}$ are precisely those for which $1 \leq i \leq d_j$ and $1 \leq j \leq m_i$.

Proof. Note that $\omega_{ij} = \frac{1}{m} x^{i-1} y^{j-m} dx$, with $p \nmid m$. It follows from [21, Proposition 3.8] (which treats $X/\mathbb{C}$ but whose proof also works for X/k and can be independently derived using the methods of [14]) that the set

$$\{x^{i-1}y^{-k}dx : 1 \leq i < d,\ 1 \leq k \leq m-1,\ dk - mi \geq \gcd(m,d)\}$$

is a basis for $\Omega_K(0)$. Taking $k = m - j$ and rearranging yields the basis

$$\boldsymbol{\omega} = \{\omega_{ij} : mi + dj \leq md - \gcd(m,d)\} = \{\omega_{ij} : mi + dj < md\},$$

and the bounds on i and j immediately follow. □

For X/k defined by $F(x,y) = f(x) - y^m = 0$, if we let f_a^n denote the coefficient of x^a in $f(x)^n$ then

$$F_{ab}^{p-1} = \begin{cases} f_a^{p-1-b/m} & \text{if } m \mid b \text{ and } b \leq m(p-1), \\ 0 & \text{otherwise,} \end{cases}$$

(here we have used $\binom{p-1}{e}(-1)^e \equiv 1 \bmod p$), thus for all $1 \le i, k < d$ and $1 \le j, \ell < m$ we have

$$F^{p-1}_{ip-k,\,jp-\ell} = \begin{cases} f^{p-1-(jp-\ell)/m}_{ip-k} & \text{if } m \mid (jp-\ell), \\ 0 & \text{otherwise.} \end{cases}$$

Now $1 \le j, \ell < m$ and $p \nmid m$, so whenever

$$F^{p-1}_{ip-k,\,jp-\ell} \neq 0,$$

we must have $\ell = jp \operatorname{rem} m > 0$ and

$$n_j = p - 1 - (jp-\ell)/m = \frac{(m-j)p-(m-\ell)}{m} = p - 1 - \lfloor jp/m \rfloor. \tag{7}$$

Let us order the basis for $\Omega_K(0)$ given by Lemma 6 as $\boldsymbol{\omega} = (\omega_{11}, \omega_{21}, \ldots, \omega_{12}, \ldots)$ with the ω_{ij} ordered first by j and then by i. The Cartier–Manin matrix of X can then be described in block form with blocks indexed by j and ℓ containing entries indexed by i and k:

$$\begin{aligned} A_p &= [B^{j\ell}]_{j\ell} && 1 \le j, \ell \le \mu = m_1 = m - \lfloor m/d \rfloor - 1, \\ B^{j\ell} &= [(b^{j\ell}_{ik})^{1/p}]_{ik} && 1 \le i \le d_j \text{ and } 1 \le k \le d_\ell, \\ b^{j\ell}_{ik} &= \begin{cases} f^{n_j}_{ip-k} & \text{if } (jp-\ell)/m \in \mathbb{Z}_{\ge 0}, \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{8}$$

The diagonal blocks $B^{j,j}$ are square but the others typically will not be square, since the bound on i depends on j while the bound on k depends on ℓ. We also note that there is at most one nonzero $B^{j\ell}$ in each row j, and in each column ℓ of $[B^{j\ell}]_{j\ell}$, since any nonzero $B^{j\ell}$ must have $\ell \equiv jp \bmod m$ (there will be no nonzero $B^{j\ell}$ for j if no $\ell \le \mu$ satisfies $\ell \equiv jp \bmod m$; this happens, for example, when $j = 1$ and $d = m = 5$ with $p \equiv 4 \bmod 5$).

Example 7. For $m = 5$ and $d = 3$ we have $g = 4$, and the 4×4 matrix A_p consists of $3 \times 3 = 9$ blocks: one 2×2, two 2×1, two 1×2, and four 1×1. For $k = \mathbb{F}_p$, the matrices A_p for $p \equiv 1, 2, 3, 4 \bmod 5$ are

$$\begin{pmatrix} f^{(4p-4)/5}_{p-1} & f^{(4p-4)/5}_{p-2} & 0 & 0 \\ f^{(4p-4)/5}_{2p-1} & f^{(4p-4)/5}_{2p-2} & 0 & 0 \\ 0 & 0 & f^{(3p-3)/5}_{p-1} & 0 \\ 0 & 0 & 0 & f^{(2p-2)/5}_{p-1} \end{pmatrix}, \quad \begin{pmatrix} 0 & 0 & f^{(4p-3)/5}_{p-1} & 0 \\ 0 & 0 & f^{(4p-3)/5}_{2p-1} & 0 \\ 0 & 0 & 0 & 0 \\ f^{(2p-4)/5}_{p-1} & f^{(2p-4)/5}_{p-2} & 0 & 0 \end{pmatrix},$$

$$\begin{pmatrix} 0 & 0 & 0 & f^{(4p-2)/5}_{p-1} \\ 0 & 0 & 0 & f^{(4p-2)/5}_{2p-1} \\ f^{(3p-4)/5}_{p-1} & f^{(3p-4)/5}_{p-2} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & f^{(3p-2)/5}_{p-1} \\ 0 & 0 & f^{(2p-3)/5}_{p-1} & 0 \end{pmatrix}.$$

For $m = 3$ and $d = 5$ we also have $g = 4$ but now the 4×4 matrix A_p consists of $2 \times 2 = 4$ blocks:

one 3×3, one 3×1, one 1×3, and one 1×1. For $k=\mathbb{F}_p$ the matrices A_p for $p\equiv 1, 2 \bmod 3$ are

$$\begin{pmatrix} f_{p-1}^{(2p-2)/3} & f_{p-2}^{(2p-2)/3} & f_{p-3}^{(2p-2)/3} & 0\\ f_{2p-1}^{(2p-2)/3} & f_{2p-2}^{(2p-2)/3} & f_{2p-3}^{(2p-2)/3} & 0\\ f_{3p-1}^{(2p-2)/3} & f_{3p-2}^{(2p-2)/3} & f_{3p-3}^{(2p-2)/3} & 0\\ 0 & 0 & 0 & f_{p-1}^{(p-1)/3} \end{pmatrix}, \quad \begin{pmatrix} 0 & 0 & 0 & f_{p-1}^{(2p-1)/3}\\ 0 & 0 & 0 & f_{2p-1}^{(2p-1)/3}\\ 0 & 0 & 0 & f_{3p-1}^{(2p-1)/3}\\ f_{p-1}^{(p-2)/3} & f_{p-2}^{(p-2)/3} & f_{p-3}^{(p-2)/3} & 0 \end{pmatrix}.$$

In both cases $\operatorname{tr} A_p = 0$ for $p \not\equiv 1 \bmod m$, but this is not true in general (consider $m=4$ and $d=3$, for example).

The block form of the Cartier–Manin matrix A_p given by (8) implies the following theorem, which plays a key role in our algorithm for computing A_p and may also be of independent interest.

Theorem 8. *Let $X : y^m = f(x)$ be a superelliptic curve over a perfect field of characteristic $p>0$ with $d=\deg(f)$. Let $\boldsymbol{\omega}$ be the basis of $\Omega_{k(X)}(0)$ given by Lemma 6, and for $1\le j\le m_1 = m-\lfloor m/d\rfloor - 1$, let $\boldsymbol{\omega}_j = \{\omega_{ij'}\in\boldsymbol{\omega} : j'=j\}$. For $1\le j\le m_1$ the Cartier operator maps the subspace spanned by $\boldsymbol{\omega}_j$ to the subspace spanned by $\boldsymbol{\omega}_\ell$, with $\ell\equiv jp \bmod m$, and this action is given by the matrix $B^{j\ell}$ defined in (8). In particular, when $p\equiv 1 \bmod m$ the Cartier operator fixes each of the subspaces spanned by $\boldsymbol{\omega}_j$.*

Proof. This is an immediate consequence of (8). □

Remark 9. In [5, Lemma 5.1] Bouw gives formulas for the coefficients of the Hasse–Witt matrix of a general cyclic cover $Y : y^m = f(x)$ of $\mathbb{P}^1$ in terms of the (possibly repeated) roots of the polynomial $f\in k[x]$, where k is an algebraically close field of characteristic p. When f is squarefree, Bouw's formulas agree with (8), after taking into account the transposition needed to get the Cartier–Manin matrix and a possible change of basis (I'm grateful to Wanlin Li and John Voight for bringing this to my attention). One can compute analogs of the formulas in (8) to handle f that are not squarefree that take into account the multiplicities of its root, but we do not consider this case here. Note that the genus of Y and therefore the dimensions of A_p will be less than that given by (1) when f is not squarefree, so while the formulas may be more involved, the problem is computationally easier.

3. Linear recurrences

The results of the previous section imply that to compute the Cartier–Manin matrix A_p of a superelliptic curve $X : y^m = f(x)$ over $\mathbb{F}_p$ it suffices to compute certain coefficients of certain powers of $f(x)$. In this section we derive linear recurrences that allow us to do this efficiently, both when $f\in\mathbb{F}_p[x]$ and when $f\in\mathbb{Z}[x]$ and we wish to compute certain coefficients of certain powers of the reduction of f modulo many primes p. In this section we generalize [17, §2], which treated the case $m=2$, in which case $A_p = B$ consists of a single block B^{11} (so $j=\ell=1$), the powers f^n that appear in the matrix entries are always the same ($n=(p-1)/2$), and every prime $p\nmid m$ is congruent to 1 modulo m. Here we allow all of these parameters to vary.

Let $f \in \mathbb{Z}[x]$ be a squarefree polynomial of degree $d \geq 3$, which we shall write as $f(x) = x^c h(x)$ with $c = 0, 1$ and $h(0) \neq 0$ (note that $x^2 \nmid f$).[2] Let $h(x) = \sum_{i=0}^{r} h_i x^i$, and for $n \geq 1$ let h_i^n denote the coefficient of x^i in $h(x)^n$. As shown in [17, §2], the identities $h^{n+1} = h \cdot h^n$ and $(h^{n+1})' = (n+1)h^n$ yield the linear relation

$$\sum_{i=0}^{r}((n+1)i - k)h_i h_{k-i}^n = 0, \tag{9}$$

which is valid for all $k \in \mathbb{Z}$ and $n \in \mathbb{Z}_{\geq 0}$. Observing that $n_j = ((m-j)p - (m-\ell))/m$ is the exponent on f in every entry of the nonzero block $B^{j\ell}$ defined in (8), let us set $n = n_j$ and rewrite (9) as

$$0 = \sum_{i=0}^{r}((m-j)p + \ell)i - mk)h_i h_{k-i}^{n_j} \equiv \sum_{i=0}^{r}(\ell i - mk)h_i h_{k-i}^{n_j} \bmod p, \tag{10}$$

which is valid for all $k \in \mathbb{Z}$. We now define

$$v_k^{n_j} := [h_{k-r+1}^{n_j}, \ldots, h_k^{n_j}] \in \mathbb{Z}^r,$$

and put $s = p - 1 - cn_j$. The entries of $v_s^n \bmod p$ suffice to compute the first row of block $B^{j\ell}$ in A_p; note that n (and potentially s) depend on j and will vary from block to block. We have $v_0^{n_j} = [0, \ldots, 0, h_0^{n_j}] = h_0^{n_j} v_0^0$, where $v_0^0 = [0, \ldots, 0, 1]$. Noting that $s < p$ and $p \nmid m$ and $p \nmid h_0$ (since f is squarefree), solving for h_k^n in (10) yields

$$v_s^{n_j} \equiv \frac{v_0^{n_j}}{(mh_0)^s s!}\prod_{i=0}^{s-1} M_i^\ell \equiv m^{cn_j} h_0^{(c+1)n_j}(-1)^{cn_j+1}(cn_j)! v_0^0 \prod_{i=0}^{s-1} M_i^\ell \bmod p, \tag{11}$$

where

$$M_{i-1}^\ell := \begin{bmatrix} 0 & \cdots & 0 & (\ell r - mi)h_r \\ mih_0 & \cdots & 0 & (\ell(r-1) - mi)h_{r-1} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & mih_0 & (\ell - mi)h_1 \end{bmatrix} \tag{12}$$

is an integer matrix that depends on the integers i, ℓ, m and the polynomial h of degree r, but is independent of p. This independence is the key to obtaining an average polynomial-time algorithm.

Remark 10. Alternatively, if we define $w_k^n = [h_{k+r-1}^{n_j}, h_{k+r-2}^{n_j}, \ldots, h_k^{n_j}]$ and $t = d_j p - d_\ell - cn_j$, the entries of w_t^n suffice to compute the last row of block $B^{j\ell}$ in A_p. Equivalently, if we put $\tilde{h}(x) = x^r h(1/x)$ (in other words, reverse the coefficients of h) and define $\tilde{v}_k^n$ in terms of $\tilde{h}^n$ as above, it suffices to compute $\tilde{v}_{\tilde{s}}^n$ where

$$\tilde{s} = rn_j - t = dn_j - d_j p + d_\ell = p - 1 - \lfloor (dj \operatorname{rem} m)p/m \rfloor. \tag{13}$$

When $m \nmid dj$ we will have $\tilde{s} < s$ if $c = 0$ (and possibly even if $c = 1$), in which case we can compute the last row more efficiently than the first.

[2] The reader may wish to assume $c = 0$ and $f = h$ on a first reading.

We have shown how to compute the first (or last) row of each of the blocks $B^{j\ell}$ that appear in the Cartier–Manin matrix of the superelliptic curve X (either for $X/\mathbb{F}_p$ or for the reductions of $X/\mathbb{Q}$ modulo varying primes p) by computing reductions of products of integer matrices modulo primes. To compute the remaining rows in the same fashion would require working modulo powers of primes, which is something we wish to avoid. In the next section we show how to efficiently reduce the computation of the remaining rows to the computation of the first row using translated curves, which allows us to always work modulo primes.

4. Translation tricks

Let $X : y^m = f(x)$ be a superelliptic curve over $\mathbb{F}_p$ of genus g, with $d = \deg(f)$. Let A_p be the Cartier–Manin matrix A_p, and for $a \in \mathbb{F}_p$, let $A_p(a)$ be the Cartier–Manin matrix of the translated curve $X_a : y^m = f(x+a)$, whose blocks we denote $B^{j\ell}(a)$ with entries $b_{ik}^{j\ell}(a)$. We omit the exponent $1/p$ that appears in (8) because we are now working over $\mathbb{F}_p$. The curve X_a is isomorphic to X, which forces A_p and $A_p(a)$ to be conjugate, but these matrices are typically not equal. Our objective in this section is to show that we can compute $B^{j\ell}$ by solving a linear system that involves the entries that appear in just the first rows of $B^{j\ell}(a)$, where a ranges over $d_j = d - \lfloor dj/m \rfloor - 1$ distinct values of $a \in \mathbb{F}_p$. Note that $B^{j\ell}$ has d_j rows and d_ℓ columns, and we recall from (8) that the $g \times g$ matrix A_p is made up of μ^2 blocks $B^{j\ell}$, where $\mu = m_1 = m - \lfloor m/d \rfloor - 1$, and we have $d_1 + \cdots + d_\mu = g$. We shall assume $p \geq d$, so that $d_j < d$ distinct values of a exist in $\mathbb{F}_p$; for $p < d$ we can easily compute A_p directly from (8).

The results in this section generalize [17, §5], which treated the case $m = 2$, where $\mu = 1$ and $A = B^{11}$. In our current setting A_p consists of $\mu \times \mu$ rectangular blocks $B^{j\ell}$ that need not be square.

For $a \in \mathbb{F}_p$ and $1 \leq j \leq \mu$ we define the upper triangular $d_j \times d_j$ matrix

$$T^j(a) = [t_{ik}^j(a)]_{ik}, \qquad t_{ik}^j(a) = \binom{k-1}{i-1} a^{k-i}, \qquad 1 \leq i, k \leq d_j.$$

We also define $T(a)$ to be the $g \times g$ block diagonal matrix with the matrices $T^j(a)$ on the diagonal, for $1 \leq j \leq \mu$. We note that

$$T^j(a)^{-1} = T^j(-a)$$

and $T(a)^{-1} = T(-a)$, as the reader may verify (or see the proof below).

Lemma 11. *For $a \in \mathbb{F}_p$ we have $B^{j\ell}(a)T^\ell(a) = T^j(a)B^{j\ell}$ for all $1 \leq j, \ell \leq \mu$, and $A_p(a) = T(a)A_pT(-a)$.*

Proof. From the block structure of A_p given by (8) it is clear that the first statement implies the second. Let $\boldsymbol{\omega}(a) = \{\omega_{ij}(a)\}$ be the basis given by Lemma 6 for X_a and define $\boldsymbol{\omega}_j(a) = \{\omega_{ij'}(a) \in \boldsymbol{\omega} : j' = j\}$. By Theorem 8, the Cartier operator of X maps the subspace spanned by $\boldsymbol{\omega}_j$ to the subspace spanned by $\boldsymbol{\omega}_\ell$ via the matrix $B^{j\ell}$, and the Cartier operator of X_a maps the subspace spanned by $\boldsymbol{\omega}_j(a)$ to the subspace spanned by $\boldsymbol{\omega}_\ell(a)$ via the matrix $B^{j\ell}(a)$. We just need to check that the matrices $T^\ell(a)$ and $T^j(a)$ correspond to the change of basis that occurs when we replace x with $x+a$. Noting that $d(x+a) = dx$

and $F(x+a)_y = F(x)_y$, we have

$$\omega_{kj}(a) = (x+a)^{k-1}y^{j-1}dx/F_y = \sum_{i=1}^{k}\binom{k-1}{i-1}a^{k-i}x^{i-1}y^{j-1}dx/F_y$$
$$= \sum_{i=1}^{k} t_{ik}^{j}(a)\omega_{ij} = \sum_{i=1}^{d_j} t_{ik}^{j}(a)\omega_{ij},$$

and it follows that $\boldsymbol{\omega}_j(a) = T^j(a)\boldsymbol{\omega}_j$ (here we are viewing $\boldsymbol{\omega}_j$ and $\boldsymbol{\omega}_j(a)$ as column vectors). This holds for any j, including ℓ, and the lemma follows. □

Let us now consider the computation of the $d_j \times d_\ell$ block $B^{j\ell}$. Computing the k-th entry in the first row of both sides of the identity $B^{j\ell}(a)T^\ell(a) = T^j(a)B^{j\ell}$ given by Lemma 11 yields

$$\sum_{s=1}^{d_\ell} b_{1s}^{j\ell}(a)t_{sk}^{\ell}(a) = \sum_{t=1}^{d_j} t_{1t}^{j}(a)b_{tk}^{j\ell},$$

which defines a linear equation with d_j unknowns $b_{tk}^{j\ell}$ in terms of the $b_{1s}^{j\ell}(a)$ and matrices $T^j(a)$ and $T^\ell(a)$ we assume are known. Taking d_j distinct values of a, say $(a_1, \ldots, a_{d_j})$, yields a linear system with d_j equations and d_j unknowns that we can solve because the $d_j \times d_j$ matrix $[t_{1t}^{j}(a_i)]_{it} = [a_i^{t-1}]_{it}$ is an invertible Vandermonde matrix $V(a_1, \ldots, a_{d_j})$. If we now define the $d_j \times d_\ell$ matrix

$$B_1^{j\ell}(a_1, \ldots, a_{d_j}) = [b_{1s}^{j\ell}(a_i)]_{is} \tag{14}$$

and let $W_1^{j\ell}$ be the $d_j \times d_\ell$ matrix whose i-th row is the i-th row of $B_1^{j\ell}$ times $T^\ell(a_i)$, we can compute $B^{j\ell}$ as

$$B^{j\ell} = V(a_1, \ldots, a_{d_j})^{-1} W_1^{j\ell}. \tag{15}$$

Remark 12. If we use Remark 10 to compute the last row of $B^{j\ell}$ we can compute the first row of $B^{j\ell}(a_i)$ for $a_1, \ldots, a_{d_j-1}$ and use (15) to deduce the last row of $W_1^{j\ell}$ from the last row of $B^{j\ell}$. One might suppose that we could instead compute the last rows of the $B^{j\ell}(a_i)$ instead of their first rows, but this is not enough to deduce $B^{j\ell}$.

Lemma 13. *Let $X : y^m = f(x)$ be a superelliptic curve over $\mathbb{F}_p$ with $d = \deg(f)$, and let $a_1, \ldots, a_{d_1}$ be distinct elements of $\mathbb{F}_p$, where $d_1 = d - \lfloor d/m \rfloor - 1$. Given the matrices $B_1^{j\ell}(a_1, \ldots, a_{d_j})$ for $1 \le j \le \mu = m_1 = m - \lfloor m/d \rfloor - 1$ with $\ell \equiv jp \bmod m$, we can compute the Cartier–Manin matrix A_p of X using $O(md^3)$ ring operations in $\mathbb{F}_p$ and space for $O(md + d^2)$ elements of $\mathbb{F}_p$.*

Proof. We can compute $V(a_1, \ldots, a_{d_j})^{-1}$ using $O(d_j^2)$ ring operations in $\mathbb{F}_p$ [10], and we can compute $T^\ell(a_i)$ in $O(d_j^2)$ ring operations (using $\binom{k}{i} = \binom{k-1}{i-1} + \binom{k-1}{i}$). The computation of $W^{j\ell}$ requires $O(d_j d_\ell^2)$ $\mathbb{F}_p$-operations, and the matrix product in (14) uses $O(d_j^2 d_l)$ ring operations, so it takes $O(d_j^2 d_\ell + d_\ell d_j^2) = O(d^3)$ ring operations to compute each $B^{j\ell}$. There are at most $\mu < m$ nonzero $B^{j\ell}$ to compute, so the total cost of computing A_p given the matrices $B_1^{j\ell}(a_1, \ldots, a_{d_j})$ is $O(md^3)$ ring operations in $\mathbb{F}_p$ while storing $O(md + d^2)$ elements of $\mathbb{F}_p$. □

Remark 14. In terms of the genus $g \sim md/2$, the bound $O(md^3)$ is equivalent to $O(gd^2)$, which is always bounded by $O(g^3)$ but can be as small as $O(g)$ if $d = O(1)$ (this assumes we use a sparse representation of A_p).

Remark 15. In addition to playing a key role in our strategy for computing A_p, using translated curves can improve performance, as noted in the case of hyperelliptic curves in [17, §6.1]. In particular, if $f(x)$ has a rational root a then the translated curve $X_a : y^m = f(x+a) = xh(x)$ will have $r = d-1$ and $c = d-r = 1$, reducing both the dimension r and number $t = p-1-cn$ of matrices M_k^ℓ that appear in the product in (11). It thus makes sense to choose our distinct translation points a to be roots of $f(x)$ whenever possible. Additionally, if d is divisible by m and $f(x)$ has a rational root a, we can replace X with $X' : y^m = x^d f(1/x+a) = g(x)$, where $g(x)$ has degree $d-1$, and this also applies to all translated curves $X'_{a'}$. This applies both locally (over $\mathbb{F}_p$) and globally (over $\mathbb{Q}$).

5. Accumulating remainder trees and forests

In this section we briefly recall some background on accumulating remainder trees and related complexity bounds. Given a sequence of $r \times r$ matrices $M_0, \dots, M_{N-1}$ and a sequence of coprime integers $m_1, \dots, m_N$ we wish to compute the sequence of reduced partial products

$$A_k = M_0 \cdots M_{k-1} \bmod m_k$$

for $1 \le k \le N$. For $0 \le k \le N/2$ let $B_k = M_{2k}M_{2k+1}$ and $b_k = m_{2k}m_{2k+1}$, where $M_N = M_{N+1} = I$ and $m_0 = m_{N+1} = 1$. Then $A_1 = M_0 \bmod m_1$, and if we recursively compute $C_k = B_0 \cdots B_{k-1} \bmod b_k = M_0 \cdots M_{2k-1} \bmod m_{2k}m_{2k+1}$ for $1 \le k \le N/2$, we can then compute

$$A_{2k} = C_k \bmod m_{2k} \quad \text{and} \quad A_{2k+1} = C_k M_{2k} \bmod m_{2k+1},$$

omitting C_{2k+1} when $k = N/2$. This is the RemainderTree algorithm given in [16]. In our setting we actually want to compute products of the form $V \prod_k M_k$ that involve a row vector V, and for this problem the RemainderForest algorithm in [16] achieves an improved time (and especially) space complexity by splitting the remainder tree into 2^κ-subtrees, for a suitable choice of κ. We record the following result from [17], in which $\|x\|$ denotes the logarithm of the largest absolute value appearing in nonzero integer matrix or integer vector x, including the case where x is a single nonzero integer.

Theorem 16 [17]. *Given $V \in \mathbb{Z}^r$, $M_1, \dots, M_N \in \mathbb{Z}^{r\times r}$, and $m_1, \dots, m_N \in \mathbb{Z}$, let $n = \lceil \log_2 N \rceil$, let B be an upper bound on $\|\prod_{j=1}^N m_j\|$ such that $B/2^\kappa$ is an upper bound on $\|\prod_{j=st}^{st+t-1} m_j\|$ for $1 \le s \le N/t$, where $t := 2^{n-\kappa}$. Let B' be an upper bound on $\|V\|$, and let H be an upper bound on $\|m_k\|, \|A_k\|$ for $1 \le k \le N$, such that $\log r \le H$, and assume that $r = O(\log N)$. The* RemainderForest *algorithm computes the vectors $V_k = VM_1 \cdots M_k \bmod m_k \in (\mathbb{Z}/m_k\mathbb{Z})^r$ for $1 \le k \le N$ in*

$$O(r^2\,\mathsf{M}(B+NH)(n-\kappa) + 2^\kappa r^2\,\mathsf{M}(B) + r\,\mathsf{M}(B'))$$

time using space bounded by

$$O(2^{-\kappa}r^2(B+NH)(n-\kappa) + r(B+B')).$$

This theorem implies the following corollary, which is all we shall use.

Corollary 17. *Fix an absolute constant $c > 0$. Let N be a positive integer, let $m_1, \dots, m_N$ be a sequence of positive coprime integers with $\log m_k \le c \log N$, let $M_0, \dots, M_{N-1} \in \mathbb{Z}^{r\times r}$ be integer matrices with r, $\|M_k\| \le c \log N$, and let $v_0 \in \mathbb{Z}^r$ be a row vector with $\|v_0\| = cN \log N$. We can compute the vectors*

$$v_k = v_0 \prod_{i=0}^{k-1} M_i \bmod m_k$$

for $1 \le k \le N$ in $O(r^2 N \log^3 N)$ time using $O(r^2 N)$ space.

Proof. Applying Theorem 16 with $\kappa = 2 \log\log N$, $B = cN \log N$, $B' = c \log N$, and $H = c \log N$, yields an $O(r^2\, \mathsf{M}(N \log N) \log N)$ time bound using $O(r^2 N)$ space. Now apply $\mathsf{M}(N) = O(N \log N)$ from [18]. □

6. Algorithms

We now give our algorithms for computing the Cartier–Manin matrix A_p of a superelliptic curve $X/\mathbb{F}_p$ and for the reductions of a superelliptic curve $X/\mathbb{Q}$ modulo good primes $p \le N$. In the descriptions below, expressions of the form "$a \operatorname{rem} m$" denote the least nonnegative remainder in Euclidean division of a by m. As above we assume X is defined by $y^m = f(x)$ with $f(x)$ squarefree of degree $d \ge 3$. We define $\mu = m - \lfloor m/d \rfloor - 1$, and for $1 \le j \le \mu$ we put $d_j = d - \lfloor dj/m \rfloor - 1$, with $d_1 \ge d_2 \ge \cdots d_\mu$ as in (6). Recall that the genus of X is $g = ((d-2)(m-1) + m - \gcd(m,d))/2$, as in (1).

Algorithm (ComputeCartierManinMatrix). Given $m \ge 2$ and squarefree $f \in \mathbb{F}_p[x]$ of degree $3 \le d \le p$ with $p \nmid m$, compute the Cartier–Manin matrix $A_p \in \mathbb{F}_p^{g\times g}$ of $X : y^m = f(x)$ as follows:

(1) Fix distinct $a_1, \dots, a_{d_1} \in \mathbb{F}_p$ that include as many roots of $f(x)$ as possible.

(2) For j from 1 to μ such that $\ell = jp \operatorname{rem} m \le \mu$:

  (a) For i from 1 to d_j:

    (i) Let $f(x + a_i) = x^c h(x) \in \mathbb{F}_p[x]$ with $c \in \{0, 1\}$ and put $r = \deg(h)$.

    (ii) Set $n = ((m-j)p - (m-\ell))/m \in \mathbb{Z}$ and $s = p - 1 - cn$.

    (iii) Compute $w_s = v_0^0 \prod_{i=0}^{s-1} M_i^\ell \in \mathbb{F}_p^r$, with $M_i^\ell \in \mathbb{F}_p^{r\times r}$ as in (12), and $u_s = s! \in \mathbb{F}_p$.

    (iv) Compute $\alpha = v_s^n = m^{-s} h_0^{n-s} u_s^{-1} w_s \in \mathbb{F}_p^r$ via (11).

    (v) Let $b_1^{j\ell}(a_i) = [\alpha_r, \alpha_{r-1}, \dots \alpha_{r-d_\ell+1}] \in \mathbb{F}_p^{d_\ell}$.

  (b) Let $B_1^{j\ell} \in \mathbb{F}_p^{d_j \times d_\ell}$ be the matrix with i-th row $b_1^{j\ell}(a_i)$ as in (14) and use $B_1^{j\ell}$ to compute $B^{j\ell} \in \mathbb{F}_p^{d_j \times d_\ell}$ via (15).

(3) Output $A_p = [B^{j\ell}]_{j\ell} \in \mathbb{F}_p^{g\times g}$ defined as in (8), with $B^{j\ell} = 0$ for $\ell \not\equiv jp \bmod m$.

There are two ways to compute w_s in step (iii). One is to compute s vector-matrix products $w_{i+1} = w_i M_i^\ell$ starting with $w_0 = [0, \dots, 0, 1] \in \mathbb{F}_p^r$, which can be accomplished using $O(pr)$ ring operations

in $\mathbb{F}_p$ using $O(r \log p)$ space (note that M_i^ℓ has only $2r-1$ nonzero entries). Alternatively one can use the Bostan–Gaudry–Schost algorithm [4], which uses an optimized interpolation/evaluation approach to compute products of matrices over polynomial rings evaluated along an arithmetic progression; in our setting we view the M_i^ℓ as matrices of linear polynomials in i evaluated along the arithmetic progression $i = 0, 1, 2, \ldots, s-1$. This involves $O(p^{1/2}(r^\omega + r^2 \log p)$ ring operations in $\mathbb{F}_p$ using $O(r^2 p^{1/2})$ space, via [4, Theorem 8] and [19], and we can similarly compute $u_s = s!$ (but note that $u_s = -1$ in the typical case where $c = 0$).

We now prove Theorem 3, which we restate here for convenience.

Theorem 3. *Given a superelliptic curve $X : y^m = f(x)$ with $f \in \mathbb{F}_p[x]$ of degree d, the algorithm* ComputeCartierManinMatrix *is able to compute the Cartier–Manin matrix of X using $O(p^{1/2} m d^2 \log p)$ space in $O(p^{1/2} m (d^{\omega+1} + d^3 \log p) \log p (\log\log p))$ time, and also using $O((md + d^2) \log p)$ space in $O((p+d) m d^2 \log p \log\log p)$ time.*

Proof. The theorem follows from Lemma 13, provided that we can compute the matrices $B_1^{j\ell}(a_1, \ldots, a_{d_j})$ within the stated complexity bounds. This computation is dominated by the cost of step (iii), which is executed $O(md)$ times. The cost of a ring operation in $\mathbb{F}_p$ can be bounded by $O(\mathsf{M}(\log p))$ via [26, Theorem 9.9], which is $O(\log p \log\log p)$, by [18]. The Bostan–Gaudry–Schost approach yields a bit-complexity of

$$O(p^{1/2}(d^\omega + d^2 \log p) \log p \log\log p)$$

time and $O(d^2 p^{1/2} \log p)$ space per iteration, and the vector-matrix multiplication approach yields a bit-complexity of $O(pd \log p \log\log p)$ and $O(d \log p)$ space per iteration; the theorem follows. □

We now present our main result, an average polynomial-time algorithm to compute the Cartier–Manin matrices of the reductions of a superelliptic curve $X/\mathbb{Q}$ at all good primes $p \le N$.

Algorithm (ComputeCartierManinMatrices). Given $m \ge 2$ and squarefree $f \in \mathbb{Z}[x]$ of degree $d \ge 3$, compute the Cartier–Manin matrices A_p of the reductions of $X : y^m = f(x)$ modulo primes $p \le N$ with $p \nmid m \operatorname{lc}(f) \operatorname{disc}(f)$ as follows:

(1) For primes $p \le N$ with $p \nmid m \operatorname{lc}(f) \operatorname{disc}(f)$ initialize $A_p \in \mathbb{F}_p^{g \times g}$ to the zero matrix.

(2) Fix distinct $a_1, \ldots, a_{d_1} \in \mathbb{Z}$ that include as many roots of f as possible.

(3) For each pair of integers $j, \ell \in [1, \mu]$:

  (a) Compute the set $P = \{p_1, p_2, \cdots\}$ of primes $p \le N$ with $jp \equiv \ell \bmod m$ such that $p \nmid m \operatorname{lc}(f) \operatorname{disc}(f)$ and $a_1, \ldots, a_{d_1}$ are distinct modulo p.

  (b) If the set P is empty proceed to the next pair j, ℓ.

  (c) For i from 1 to d_j:

    (i) Let $f(x + a_i) = x^c h(x) \in \mathbb{Z}[x]$ with $c \in \{0, 1\}$ and put $r = \deg(h)$.

(ii) Let $N' = N$ if $c = 0$ and $N' = \lfloor (jN - \ell)/m \rfloor$ otherwise.

(iii) Define coprime moduli $m_1, \dots, m_{N'}$ as follows:

If $c = 0$ then $m_k = k + 1$ for $k + 1 \in P$.

If $c = 1$ then $m_k = (mk + \ell)/j$ for $(mk + \ell)/j \in P$.

For any m_k not defined above, let $m_k = 1$.

For $p \in P$ let $k(p)$ denote the index k of the m_k for which $m_k = p$.

(iv) Compute $w_k = v_0^0 \prod_{i=0}^{k-1} M_i^\ell \bmod m_k$ and $u_k = k! \bmod m_k$ for $1 \le k \le N'$ as in Corollary 17.

(v) For $p \in P$ use $w_{k(p)}$, $u_{k(p)}$ to compute $b_1^{j\ell}(a_i) \in \mathbb{F}_p^{d_\ell}$ as in COMPUTECARTIERMANINMATRIX.

(d) For $p \in P$, let $B_1^{j\ell} \in \mathbb{F}_p^{d_j \times d_\ell}$ have rows $b_1^{j\ell}(a_i) \in \mathbb{F}_p^{d_\ell}$ as in (14), use $B_1^{j\ell}$ to compute $B^{j\ell} \in \mathbb{F}_p^{d_j \times d_\ell}$ via (15), and set the j, ℓ block of A_p to $B^{j\ell}$ as in (8).

(4) Let S be the set of primes $p \le N$ satisfying $p \nmid m \operatorname{lc}(f) \operatorname{disc}(f)$ for which the $a_1, \dots a_{d_1}$ are not distinct modulo p. For $p \in S$ compute A_p using algorithm COMPUTECARTIERMANINMATRIX if $p \ge d$ and otherwise compute A_p directly from (8) by extracting coefficients of powers of $f \in \mathbb{F}_p[x]$.

(5) Output $A_p \in \mathbb{F}_p^{g \times g}$ for all primes $p \le N$ such that $p \nmid m \operatorname{lc}(f) \operatorname{disc}(f)$.

Remark 18. To compute Frobenius traces $a_p \in \mathbb{Z}$, we modify step (3) to loop over integers $j = \ell \in [1, \mu]$ and output just the traces of the A_p in step (5). This gives the traces of Frobenius $a_p \bmod p$. For $p > 16g^2$ these determine $a_p \in \mathbb{Z}$, by the Weil bounds, and for $p \le 16g^2$ we can compute

$$a_p = p + 1 - \#X(\mathbb{F}_p)$$

by enumerating values of $f(x)$ and looking them up in a precomputed table of m-th powers.

Remark 19. The loop in step (c) is executed (up to) μg times. Each of these computations is completely independent of the others, which makes it easy to efficiently distribute the work across μg threads. In principal one can also parallelize the integer matrix multiplications performed by the REMAINDERFOREST algorithm in step (iv), but in practice it is extremely difficult to do this efficiently.

We now prove Theorem 1, which we restate for convenience.

Theorem 1. *Given a superelliptic curve $X : y^m = f(x)$ with $f \in \mathbb{Z}[x]$ of degree d and $N \in \mathbb{Z}_{>0}$, the algorithm* COMPUTECARTIERMANINMATRICES *outputs the Cartier–Manin matrices A_p of the reductions of X modulo all primes $p \le N$ not dividing $m \operatorname{lc}(f) \operatorname{disc}(f)$. If we assume m, d, $\|f\|$ are bounded by $O(\log N)$ the algorithm runs in $O(m^2 d^3 N \log^3 N)$ time using $O(md^2 N)$ space; it can alternatively compute Frobenius traces $a_p \in \mathbb{Z}$ for $p \le N$ in time $O(md^3 N \log^3 N)$.*

Proof. The total time to compute all the sets P using a sieve is bounded by $O(N \log N)$ time using $O(N)$ space, and this also bounds the total time and space for steps (i), (ii), (iii), under our assumption that $m, d, \|f\| = O(\log N)$. Corollary 17 yields an $O(d^2 N \log^3 N)$ bound on each of the $O(m^2 d)$ iterations of step (iv). This yields the claimed time bound of $O(m^2 d^3 N \log^3 N)$ for step (c), which we claim

dominates. Lemma 13 implies that the total cost of step (d) is bounded by $O(\pi(N)m^2d^3\log N)$, which is negligible, as is the cost of the rest of the algorithm. Note that the cardinality of the set S in step (4) is at worst quadratic in d and $\log(N)$ under our assumption $\|f\| = O(\log N)$, so we can easily afford the calls to ComputeCartierManinMatrix and use a brute force approach to compute A_p for primes $p < d$ of good reduction.

The space bound follows from the bound in Corollary 17, which covers step (iv) (it is easy to see that all of the other steps fit within the claimed bound).

To compute Frobenius traces $a_p \in \mathbb{Z}$ we apply Remark 18 and note that restricting to $j = \ell$ in step (3) reduces the number of iterations of the main loop by a factor of m. The cost of computing $\#X(\mathbb{F}_p)$ by looking up values of $f(x)$ in a table of m-th powers is $O(pd)$ ring operations in $\mathbb{F}_p$. The total time to compute $a_p = p + 1 - \#X(\mathbb{F}_p)$ for good $p \le 16g^2$ is then

$$O(dg^2\pi(g^2)\log g \log\log g) = O(d(\log N)^4 \log\log N),$$

which is negligible. □

7. Supplementary material

Tables 1 and 2 compare the performance of the average polynomial-time algorithm ComputeCartierManinMatrices with the $\tilde{O}(p^{1/2})$ algorithm for computing zeta functions of cyclic covers implemented in Sage version 9.0. The Sage implementation provides the function CyclicCover which takes an integer m and a squarefree polynomial $f \in \mathbb{F}_p[x]$ and returns an object that represents a superelliptic curve $y^m = f(x)$ over $\mathbb{F}_p$. Invoking the frobenius_matrix method of this object with the p-adic precision set to 1 yields a matrix that encodes essentially the same information as the Cartier–Manin matrix A_p; in particular it determines the p-rank of X and its zeta function modulo p.

Each table lists the genus g and invariants m and d of a superelliptic curve $X : y^m = f(x)$ defined over $\mathbb{Q}$ with $f \in \mathbb{Z}[x]$ of degree d. There is a row for every pair $m \ge 2$ and $d \ge 3$ for which $m^2d^3 \le 6^5$, which includes all superelliptic curves of genus $g \le 5$ as well as plane quintics and sextics, and other curves of genus up to 15. The times listed are average times in milliseconds for primes $p \le N$ for increasing values of N. For each N three times are listed: one to compute Frobenius matrices using Sage, one to compute Cartier–Manin matrices using the algorithm ComputeCartierManinMatrices, and one to compute Frobenius traces via Remark 18. For the Sage timings we only computed Frobenius matrices for every n-th good prime $p \le N$ with n chosen so that the computation would complete in less than a day (many of the computations would have taken months otherwise).

In Table 1 we show timings with $f \in \mathbb{Z}[x]$ having coefficients $f_{d+1-n} = p_n$ for $1 \le n \le d$, where p_n is the n-th prime. These polynomials are all irreducible, so our algorithm was unable to choose any a_i to be roots of f; this is the generic situation, and the worst case for our algorithm. In Table 2 we show timings with $f \in \mathbb{Z}[x]$ a product of linear factors, which represents the best case for our algorithm.

			$N = 2^{20}$			$N = 2^{24}$			$N = 2^{28}$		
g	m	d	sage	matrix	trace	sage	matrix	trace	sage	matrix	trace
1	2	3	27	0.05	0.05	67	0.13	0.13	230	0.30	0.30
1	2	4	41	0.17	0.16	120	0.42	0.42	454	0.95	0.93
1	3	3	46	0.08	0.08	141	0.20	0.20	499	0.48	0.49
2	2	5	55	0.38	0.38	163	0.92	0.92	580	2.02	2.01
2	2	6	83	0.73	0.74	280	1.77	1.77	1070	3.89	3.92
3	2	7	112	1.30	1.29	307	3.19	3.12	1217	6.47	6.71
3	2	8	169	2.15	2.07	528	5.02	4.94	2106	10.20	10.57
3	3	4	61	0.53	0.26	178	1.38	0.70	702	3.14	1.63
3	4	3	58	0.14	0.15	165	0.37	0.37	601	0.89	0.89
3	4	4	101	0.44	0.44	343	1.14	1.14	1283	2.55	2.63
4	2	9	194	3.22	3.24	576	7.65	7.70	2214	16.12	15.90
4	2	10	319	4.78	4.65	974	11.10	10.98	3693	22.13	22.79
4	3	5	93	1.29	0.65	287	3.37	1.67	1105	7.64	3.68
4	3	6	152	2.59	1.28	535	6.34	3.20	2121	14.04	7.07
4	5	3	68	0.40	0.13	200	1.19	0.40	778	2.96	0.99
4	6	3	112	0.24	0.24	313	0.64	0.64	1184	1.53	1.53
5	2	11	361	7.04	7.06	1024	16.57	16.30	3695	33.61	33.32
5	2	12	555	9.56	9.54	1537	21.84	22.23	5820	45.98	45.65
6	3	7	200	4.61	2.32	632	11.53	5.52	2360	24.18	12.18
6	4	5	130	1.71	1.08	424	4.37	2.73	1658	9.86	5.88
6	5	4	113	1.29	0.42	344	3.76	1.25	1358	9.08	3.03
6	5	5	201	3.06	1.02	671	8.98	2.92	2749	19.39	6.64
6	7	3	94	0.68	0.17	290	2.24	0.56	1146	5.57	1.39
7	3	8	294	8.17	4.05	835	19.07	9.38	3279	40.32	20.49
7	3	9	437	12.77	6.32	1462	28.54	14.50	5567	61.82	29.67
7	4	6	232	3.42	2.12	806	8.58	5.21	3160	18.99	11.54
7	6	4	153	1.08	0.77	524	2.79	2.00	2112	6.46	4.55
7	8	3	111	0.60	0.29	366	1.72	0.83	1333	4.32	2.00
7	9	3	140	0.82	0.26	479	2.64	0.82	1870	6.77	2.03
9	4	7	302	6.49	3.94	941	15.10	9.42	3566	32.97	20.43
9	7	4	156	2.77	0.56	510	9.14	1.78	2012	20.90	4.21
9	8	4	231	1.85	0.92	720	5.43	2.57	2941	12.58	6.12
9	10	3	137	0.76	0.34	429	2.29	1.01	1694	5.82	2.50
10	5	6	265	8.08	2.02	840	22.89	5.62	3256	51.62	12.42
10	6	5	206	2.51	1.83	701	6.28	4.61	2700	14.07	9.88
10	6	6	379	5.05	3.49	1278	11.95	8.59	5202	26.43	18.72
10	11	3	158	1.77	0.25	501	6.11	0.88	1878	15.32	2.12
10	12	3	187	0.80	0.49	636	2.35	1.39	2558	5.87	3.45
12	7	5	246	6.75	1.33	840	20.80	4.13	3228	48.09	9.23
12	9	4	199	2.88	0.87	657	8.87	2.64	2655	21.75	6.24
12	13	3	175	2.43	0.29	616	8.24	1.03	2244	20.02	2.49
13	10	4	264	2.90	1.09	1008	8.62	3.17	3762	20.08	7.47
13	14	3	193	1.58	0.43	619	5.01	1.36	2430	12.79	3.40
13	15	3	235	1.69	0.46	811	5.54	1.45	3238	13.99	3.72
15	11	4	252	6.29	0.79	839	22.76	2.84	3334	52.85	6.59
15	16	3	223	1.79	0.53	733	5.66	1.63	2805	14.16	4.13

Table 1. Comparison with $\tilde{O}(p^{1/2})$ Sage 9.0 implementation [2] for superelliptic curves $y^m = f(x)$ where $f \in \mathbb{Z}[x]$ is irreducible of degree d. Times are millisecond averages per prime $p \leq N$ for a single thread running on a 2.8GHz Cascade Lake Intel CPU. The sage column lists the average time to execute `CyclicCover(m,f.change_ring(GF(p))).frobenius_matrix(1)` in Sage 9.0, the matrix column lists the average time to compute the Cartier–Manin matrix modulo p using algorithm ComputeCartierManinMatrices, and the trace column is the average time to compute the trace of Frobenius via Remark 18.

			$N = 2^{20}$			$N = 2^{24}$			$N = 2^{28}$		
g	m	d	sage	matrix	trace	sage	matrix	trace	sage	matrix	trace
1	2	3	28	0.01	0.01	73	0.04	0.04	230	0.09	0.08
1	2	4	43	0.04	0.05	119	0.12	0.12	456	0.28	0.27
1	3	3	45	0.01	0.01	131	0.02	0.02	500	0.05	0.05
2	2	5	53	0.11	0.12	151	0.31	0.30	583	0.72	0.72
2	2	6	84	0.26	0.28	267	0.66	0.64	1071	1.40	1.40
3	2	7	116	0.55	0.54	311	1.22	1.20	1219	2.58	2.59
3	2	8	164	0.94	0.92	532	2.06	2.04	2094	4.19	4.23
3	3	4	62	0.14	0.07	184	0.41	0.20	701	0.96	0.47
3	4	3	55	0.03	0.03	157	0.08	0.08	605	0.20	0.20
3	4	4	103	0.08	0.09	334	0.23	0.23	1286	0.55	0.54
4	2	9	199	1.50	1.47	586	3.48	3.41	2232	7.10	7.12
4	2	10	295	2.30	2.29	942	5.37	5.24	3816	10.53	10.37
4	3	5	92	0.38	0.19	283	1.06	0.51	1111	2.40	1.21
4	3	6	153	0.79	0.41	529	1.85	0.91	2098	3.96	1.99
4	5	3	68	0.05	0.02	202	0.16	0.05	780	0.39	0.13
4	6	3	95	0.03	0.03	301	0.09	0.09	1186	0.22	0.21
5	2	11	354	3.45	3.46	977	7.85	7.87	3682	15.94	15.85
5	2	12	530	5.11	5.12	1543	11.30	11.17	5857	22.61	22.62
6	3	7	192	1.47	0.72	605	3.57	1.78	2361	7.67	3.79
6	4	5	136	0.32	0.25	416	0.94	0.61	1660	2.17	1.43
6	5	4	108	0.30	0.10	348	1.00	0.32	1369	2.43	0.81
6	5	5	196	0.52	0.15	710	1.48	0.48	2755	3.49	1.16
6	7	3	96	0.06	0.02	296	0.23	0.06	1146	0.63	0.15
7	3	8	276	3.05	1.54	836	7.04	3.49	3234	15.09	7.64
7	3	9	427	4.09	2.16	1409	9.28	4.74	5551	21.20	10.35
7	4	6	227	0.98	0.65	774	2.30	1.48	3143	5.26	3.33
7	6	4	155	0.23	0.17	525	0.66	0.44	2108	1.53	1.04
7	8	3	111	0.06	0.04	343	0.20	0.12	1333	0.51	0.30
7	9	3	141	0.08	0.03	476	0.28	0.09	1876	0.76	0.23
9	4	7	289	1.85	1.23	917	4.56	2.88	3555	10.28	6.23
9	7	4	156	0.61	0.10	509	1.78	0.35	2007	4.47	0.88
9	8	4	211	0.33	0.18	752	1.05	0.50	2946	2.64	1.23
9	10	3	139	0.08	0.04	430	0.26	0.12	1694	0.66	0.31
10	5	6	253	2.08	0.52	825	5.96	1.49	3265	13.97	3.37
10	6	5	213	0.68	0.42	676	1.61	1.06	2693	3.83	2.43
10	6	6	365	1.23	0.86	1276	2.94	2.00	5195	6.46	4.34
10	11	3	154	0.14	0.02	477	0.52	0.08	1878	1.48	0.21
10	12	3	189	0.08	0.07	640	0.26	0.17	2552	0.63	0.42
12	7	5	242	1.22	0.24	879	3.99	0.77	3227	9.89	1.93
12	9	4	204	0.60	0.17	672	1.66	0.52	2663	4.30	1.26
12	13	3	175	0.19	0.02	569	0.71	0.09	2245	2.06	0.25
13	10	4	267	0.64	0.22	942	1.69	0.65	3779	4.32	1.56
13	14	3	191	0.14	0.05	617	0.47	0.15	2429	1.23	0.37
13	15	3	240	0.14	0.04	806	0.50	0.14	3246	1.35	0.36
15	11	4	251	1.15	0.14	836	3.92	0.49	3314	9.89	1.26
15	16	3	218	0.15	0.06	728	0.52	0.19	2797	1.37	0.48

Table 2. Timings for superelliptic curves $X : y^m = f(x)$ when $f \in \mathbb{Z}[x]$ splits into d distinct linear factors. Times are millisecond averages per prime $p \le N$ for a single thread running on a 2.8GHz Cascade Lake Intel CPU. The sage column lists the average time to execute `CyclicCover(m,f.change_ring(GF(p)).frobenius_matrix(1)` in Sage 9.0, the matrix column lists the average time to compute the Cartier–Manin matrix modulo p using algorithm ComputeCartierManinMatrices, and the trace column is the average time to compute the trace of Frobenius via Remark 18.

References

[1] Jeffrey D. Achter and Everett W. Howe, *Hasse–Witt and Cartier–Manin matrices: a warning and a request*, Arithmetic geometry: computation and applications, Contemp. Math., no. 722, Amer. Math. Soc., Providence, RI, 2019, pp. 1–18. MR 3896846

[2] Vishal Arul, Alex J. Best, Edgar Costa, Richard Magner, and Nicholas Triantafillou, *Computing zeta functions of cyclic covers in large characteristic*, Proceedings of the Thirteenth Algorithmic Number Theory Symposium (Berkeley, CA), Open Book Ser., no. 2, Math. Sci. Publ., 2019, pp. 37–53. MR 3952003

[3] Andrew R. Booker, Jeroen Sijsling, Andrew V. Sutherland, John Voight, and Dan Yasaki, *A database of genus-2 curves over the rational numbers*, LMS J. Comput. Math., **19A** (2016), 235–254. MR 3540958

[4] Alin Bostan, Pierrick Gaudry, and Éric Schost, *Linear recurrences with polynomial coefficients and application to integer factorization and Cartier–Manin operator*, SIAM J. Comput. **36** (2007), no. 6, 1777–1806. MR 2299425

[5] Irene I. Bouw, *The p-rank of ramified covers of curves*, Compositio Math. **126** (2001), no. 3, 295–322. MR 1834740

[6] Irene I. Bouw and Stefan Wewers, *Computing L-functions and semistable reduction of superelliptic curves*, Glasg. Math. J. **59** (2017), no. 1, 77–108. MR 3576328

[7] Alina Bucur, Francesc Fité, and Kiran S. Kedlaya, *Effective Sato–Tate conjecture for abelian varieties and applications*, preprint, 2020. arXiv 2002.08807

[8] Claude Chevalley, *Introduction to the Theory of Algebraic Functions of One Variable*, Mathematical Surveys, no. 6, American Mathematical Society, New York, 1951. MR 0042164

[9] Edgar Costa, Francesc Fité, and Andrew V. Sutherland, *Arithmetic invariants from Sato–Tate moments*, C. R. Math. Acad. Sci. Paris **357** (2019), no. 11-12, 823–826. MR 4038255

[10] A. Eisinberg and G. Fedele, *On the inversion of the Vandermonde matrix*, Appl. Math. Comput. **174** (2006), no. 2, 1384–1397. MR 2220623

[11] Francesc Fité, Kiran S. Kedlaya, Víctor Rotger, and Andrew V. Sutherland, *Sato–Tate distributions and Galois endomorphism modules in genus 2*, Compos. Math. **148** (2012), no. 5, 1390–1442. MR 2982436

[12] Francesc Fité, Kiran S. Kedlaya, and Andrew V. Sutherland, *Sato–Tate groups of abelian threefolds: a preview of the classification*, preprint, 2019. arXiv 1911.02071

[13] Cécile Gonçalves, *A point counting algorithm for cyclic covers of the projective line*, Algorithmic arithmetic, geometry, and coding theory, Contemp. Math., no. 637, Amer. Math. Soc., Providence, RI, 2015, pp. 145–172. MR 3364447

[14] Daniel Gorenstein, *An arithmetic theory of adjoint plane curves*, Trans. Amer. Math. Soc. **72** (1952), 414–436. MR 49591

[15] David Harvey, Maike Massierer, and Andrew V. Sutherland, *Computing L-series of geometrically hyperelliptic curves of genus three*, LMS J. Comput. Math., **19A** (2016), 220–234. MR 3540957

[16] David Harvey and Andrew V. Sutherland, *Computing Hasse–Witt matrices of hyperelliptic curves in average polynomial time*, LMS J. Comput. Math., **17A** (2014), 257–273. MR 3240808

[17] David Harvey and Andrew V.Sutherland, *Computing Hasse–Witt matrices of hyperelliptic curves in average polynomial time, II*, Frobenius distributions: Lang–Trotter and Sato–Tate conjectures, Contemp. Math., no. 663, Amer. Math. Soc., Providence, RI, 2016, pp. 127–147. MR 3502941

[18] David Harvey and Joris van der Hoeven, *Integer multiplication in time $O(n \log n)$*, preprint, 2019.

[19] David Harvey and Joris van der Hoeven, *Polynomial multiplication over finite fields in time $O(n \log n)$*, preprint, 2019.

[20] Moritz Minzlaff, *Computing zeta functions of superelliptic curves in larger characteristic*, Math. Comput. Sci. **3** (2010), no. 2, 209–224. MR 2608297

[21] Pascal Molin and Christian Neurohr, *Computing period matrices and the Abel–Jacobi map of superelliptic curves*, Math. Comp. **88** (2019), no. 316, 847–888. MR 3882287

[22] Jean-Pierre Serre, *Lectures on $N_X(p)$*, Research Notes in Mathematics, no. 11, CRC Press, 2012. MR 2920749

[23] Henning Stichtenoth, *Algebraic function fields and codes*, 2nd ed., Graduate Texts in Mathematics, no. 254, Springer, 2009. MR 2464941

[24] Karl-Otto Stöhr and José Felipe Voloch, *A formula for the Cartier operator on plane algebraic curves*, J. Reine Angew. Math. **377** (1987), 49–64. MR 887399

[25] Andrew V. Sutherland, *A database of nonhyperelliptic genus-3 curves over* $\mathbb{Q}$, Proceedings of the Thirteenth Algorithmic Number Theory Symposium (Berkeley, CA) (Renate Scheidler and Jonathan Sorenson, eds.), Open Book Ser., no. 2, Math. Sci. Publ., 2019, pp. 443–459. MR 3952027

[26] Joachim von zur Gathen and Jürgen Gerhard, *Modern computer algebra*, 3rd ed., Cambridge University Press, 2013. MR 3087522

[27] Yuri G. Zarhin, *Endomorphism algebras of abelian varieties with special reference to superelliptic Jacobians*, Geometry, algebra, number theory, and their information technology applications, Springer Proc. Math. Stat., no. 251, Springer, 2018, pp. 477–528. MR 3880401

Received 28 Feb 2020. Revised 28 Feb 2020.

ANDREW V. SUTHERLAND: drew@math.mit.edu
Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA, United States

VOLUME EDITORS

Stephen D. Galbraith
Mathematics Department
University of Auckland
New Zealand
https://orcid.org/0000-0001-7114-8377

The cover image is based on an illustration from the article "Supersingular curves with small noninteger endomorphisms", by Jonathan Love and Dan Boneh (see p. 9).

Electronic copies can be obtained free of charge from http://msp.org/obs/4 and printed copies can be ordered from MSP (contact@msp.org).

The Open Book Series is a trademark of Mathematical Sciences Publishers.

ISSN: 2329-9061 (print), 2329-907X (electronic)

ISBN: 978-1-935107-07-1 (print), 978-1-935107-08-8 (electronic)

First published 2020.

MATHEMATICAL SCIENCES PUBLISHERS
798 Evans Hall #3840, c/o University of California, Berkeley CA 94720-3840
contact@msp.org **http://msp.org**

THE OPEN BOOK SERIES 4

Fourteenth Algorithmic Number Theory Symposium

The Algorithmic Number Theory Symposium (ANTS), held biennially since 1994, is the premier international forum for research in computational and algorithmic number theory. ANTS is devoted to algorithmic aspects of number theory, including elementary, algebraic, and analytic number theory, the geometry of numbers, arithmetic algebraic geometry, the theory of finite fields, and cryptography.

This volume is the proceedings of the fourteenth ANTS meeting, which took place 29 June to 4 July 2020 via video conference, the plans for holding it at the University of Auckland, New Zealand, having been disrupted by the COVID-19 pandemic. The volume contains revised and edited versions of 24 refereed papers and one invited paper presented at the conference.

TABLE OF CONTENTS