COUNTING POINTS ON SMOOTH PLANE QUARTICS

EDGAR COSTA, DAVID HARVEY, AND ANDREW V. SUTHERLAND

Abstract. We present efficient algorithms for counting points on a smooth plane quartic curve X modulo a prime p. We address both the case where X is defined over $\mathbb{F}_p$ and the case where X is defined over $\mathbb{Q}$ and p is a prime of good reduction. We consider two approaches for computing $\#X(\mathbb{F}_p)$, one which runs in $O(p \log p \log\log p)$ time using $O(\log p)$ space and one which runs in $O(p^{1/2}\log^2 p)$ time using $O(p^{1/2}\log p)$ space. Both approaches yield algorithms that are faster in practice than existing methods. We also present average polynomial-time algorithms for $X/\mathbb{Q}$ that compute $\#X(\mathbb{F}_p)$ for good primes $p \leqslant N$ in $O(N\log^3 N)$ time using $O(N)$ space. These are the first practical implementations of average polynomial-time algorithms for curves that are not cyclic covers of $\mathbb{P}^1$, which in combination with previous results addresses all curves of genus $g \leqslant 3$. Our algorithms also compute Cartier–Manin/Hasse–Witt matrices that may be of independent interest.

1. Introduction

Let $X/\mathbb{Q}$ be a smooth projective curve of genus g. The L-function $L(X,s) = \sum_{n\geqslant 1} a_n n^{-s}$ is a Dirichlet series that is defined by an Euler product $\prod_p L_p(p^{-s})^{-1}$, where $L_p(T)$ is an integer polynomial of degree at most $2g$. For primes p of good reduction for X the polynomial $L_p(T)$ is the numerator of the zeta function

$$Z_p(T) \coloneqq \exp\left(\sum_{r\geqslant 1} \#X(\mathbb{F}_{p^r})\frac{T^r}{r}\right) = \frac{L_p(T)}{(1-T)(1-pT)} \tag{1.1}$$

of the reduction of X modulo p. The L-function $L(X,s)$ and its coefficients a_n are the subject of many outstanding conjectures, including the connection to automorphic forms predicted by the Langlands program, generalizations of the Sato–Tate conjecture, the Lang–Trotter conjecture, and the conjecture of Birch and Swinnerton-Dyer, as well as conjectures about the zeros and special values of $L(X,s)$. To numerically investigate these conjectures one needs to compute the Dirichlet coefficients a_n for n up to some bound N that one would like to make as large as possible, and at a minimum, larger than the square root of the conductor of $L(X,s)$ by a significant constant factor.

Since $L(X,s)$ is defined by an Euler product, its coefficients a_n for $n \leqslant N$ are determined by the coefficients a_{p^e} for prime powers $p^e \leqslant N$, almost all of which are Frobenius traces

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

at primes p of good reduction for X. From a computational perspective, the problem of computing the integers a_n for $n \leqslant N$ is overwhelmingly dominated by the cost of computing Frobenius traces a_p for good primes $p \leqslant N$, equivalently, counting points on X modulo primes $p \leqslant N$ of good reduction, which is the problem we consider here.

The first and third authors were supported by Simons Foundation grant 550033.

The second author was supported by the Australian Research Council (grant FT160100219).

There are two existing algorithms that can compute a_p for good primes $p \leqslant N$ in time $\widetilde{O}(N)$, which is optimal up to logarithmic factors, since it is quasilinear in the size of the output. The first is Pila's generalization of Schoof's algorithm [Sch85, Pil90, AH01], which can compute each a_p in time $(\log p)^{O(1)}$, leading to a total time of $N(\log N)^{O(1)}$. The second is Harvey's average polynomial-time algorithm [Har15], which can compute a_p for good $p \leqslant N$ in time $O(N \log^3 N)$. Neither of these algorithms is meant to be practical for $g > 1$, but the second has the distinct advantage that the implicit constant (which increases with g) is not in the exponent of the complexity bound. For $g = 1$ both algorithms are practical, but the $\widetilde{O}(N^{5/4})$ generic group algorithm described in [KS08] is faster for all practical values of N.

The case $g = 2$ is efficiently addressed by the practical implementation of Harvey's algorithm for hyperelliptic curves given in [HS14] and improved in [HS16]. Prior work has addressed the case $g = 3$ in various special cases, including when X is hyperelliptic, either as a degree-2 cover of $\mathbb{P}^1$ [HS16] or as a degree-2 cover of a pointless conic [HMS16], and when X is superelliptic, including Picard curves and cyclic 4-covers of $\mathbb{P}^1$ [Sut20]. But the generic case of a smooth plane quartic is not efficiently addressed by any prior work we are aware of.

In this article we consider three practical average polynomial-time algorithms for computing the Frobenius traces a_p of a smooth plane quartic $X/\mathbb{Q}$ at good primes $p \leqslant N$. As with the average polynomial-time algorithms mentioned above, they all involve the computation of partial products of a sequence of $r \times r$ integer matrices $M_0, \ldots, M_{N-1}$ reduced modulo coprime integers $m_0, \ldots, m_{N-1}$ that include the primes $p \leqslant N$. This can be accomplished in $O(r^2 N \log^3 N)$ time using $O(r^2 N \log N)$ space via an accumulating remainder tree, and one can improve the constant factors in the time complexity and reduce the space complexity to $O(r^2 N)$ using the accumulating remainder forest described in [HS14, HS16]; see Theorem 5.21 for a precise statement. As with other average polynomial-time algorithms, one can alternatively use these matrices to count points modulo a particular prime p in two ways: one runs in $O(r^2 p \log p \log\log p)$ time using $O(r^2 \log p)$ space and the other runs in $O(r^2 p^{1/2} \log^2 p)$ time using $O(r^2 p^{1/2} \log p)$ space, assuming $r = O(\log p)$.

Our restriction to genus 3 curves effectively fixes r, so r^2 becomes a constant factor that is hidden in our complexity bounds. But r takes different values in each of the three algorithms we present, and this has a significant impact on their relative running times. Constant factors related to the size of the matrix coefficients size also play a role, but they are less significant; see §6 for a detailed discussion and a performance comparison of the three algorithms.

Our algorithms compute the trace of Frobenius a_p by computing the trace of the Cartier–Manin matrix $A_p \in \mathbb{F}_p^{3\times 3}$ of the smooth plane quartic $X_p\colon f(x_0, x_1, x_2) = 0$ over $\mathbb{F}_p$ given by reducing X modulo p. The precise definition of A_p is recalled in §2, but its entries consist of nine particular coefficients of f^{p-1} and its trace is congruent to a_p modulo p, which uniquely determines a_p for $p > 144$. The Cartier–Manin matrix provides additional information about X_p, including the p-rank of its Jacobian and the reduction of $L_p(T)$ modulo p, which constrains $L_p(T)$ to $O(p^{1/2})$ possibilities. These possibilities can be distinguished in $\widetilde{O}(p^{1/4})$ time using a probabilistic generic group algorithm working in the Jacobian of X; see [Sut07, KS08, Sut09] for details of the algorithm and see [FOR08] for efficient implementation of the group operation. This does not yield an average polynomial time for computing $L_p(T)$ for good $p \leqslant N$, it would have complexity $\widetilde{O}(N^{5/4})$, but for the practical range of N this approach is faster in practice than using the average polynomial-time algorithm in [Har15], which can compute $L_p(T)$ for good $p \leqslant N$ in $O(N \log^3 N)$ time.

The key differences among the three algorithms we consider lie in the relations that are used to define the matrices M_i and the sizes of these matrices; in particular the value of r may be 66, 28, or 16. The relations used in [Har15] are based on a deformation approach that in the case of a plane quartic curve $X := f(x_0, x_1, x_2) = 0$ introduces an auxiliary polynomial $g(x_0, x_1, x_2) = x_0^4 + x_1^4 + x_2^4$ and derives relations between the coefficients that appear in the terms of the binomial expansion of $(f + tg)^{p-1}$, where t is an auxiliary parameter. These relations yield 66×66 matrices M_i. Rather than using the general algorithm given in [Har15], which does not not require X to be smooth or even a curve (it can be any hypersurface), one can use these matrices to directly compute the coefficients of f^{p-1} that appear in the Cartier–Manin matrix A_p via [Har15, Thm. 4.1], as we explain in §5. With appropriate optimizations the resulting algorithm is quite practical and faster than previous approaches, as demonstrated by the timings in Table 1.

However, the main focus of this paper is deriving new relations that yield smaller matrices M_i. In contrast to [Har15], which uses relations that involve coefficients of mth-powers of the homogeneous polynomial F that defines X, where the parameter m may vary, here we fix m. This forces us to impose nondegeneracy conditions on F that are not required in [Har15], but it yields 28×28 matrices, and the resulting algorithms for computing Cartier–Manin matrices, either for a single prime p or all good $p \leqslant N$ are substantially faster in practice than those that use the 66×66 matrices based on [Har15]. The relations we obtain are not independent, and we develop tools that allow us to compress them. This yields 16×16 matrices of full rank with slightly larger coefficients that provides a further substantial improvement in practical running times; see Tables 1–3.

Our algorithms for smooth plane quartics are not as fast as those that have been developed for genus 3 curves of a special form, such as hyperelliptic or superelliptic curves; see Table 4 for a comparison. Nevertheless, for general genus 3 curve the algorithms we present substantially extend the practical range of N one may consider. This played a key role in [FKS21, FKS22] where a preliminary version of our algorithm was used to compute Sato–Tate distributions, and in computing the L-functions of the nonhyperelliptic genus 3 curves tabulated in [Sut19].

We conclude this introduction with an outline of the paper. After briefly recalling the definition of the Cartier–Manin matrix and some of its properties in Section 2, we devote Sections 3 and 4 to developing the recurrences that determine the matrices M_i used by our algorithms; the main result used to define the 28×28 matrices M_i appears in Lemma 4.4, and the result that allows us to compress them to 16×16 matrices appears in Lemma 3.13. The algorithms themselves are presented in Section 5, along with an analysis of their complexity, and Section 6 compares the performance of our algorithms to each other and to existing approaches for counting points on smooth plane quartics, as well as to previously developed average polynomial-time algorithms for hyperelliptic and superelliptic genus 3 curves.

2. The Cartier matrix of a smooth plane curve

In this section we recall the definition of the Cartier matrix of a smooth plane curve, following [Sut20]. Let k be a perfect field of characteristic $p > 0$, let K be a function field of transcendence degree one with field of constants k, and let Ω_K denote its module of differentials, which we identify with its module of Weil differentials via [Sti09, Def. 4.17] and [Sti09, Rm. 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 pth 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 $z_i \in K$. Each rational differential form $\omega = z dx$ can then be written uniquely as

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

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

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

It maps regular differentials to regular differentials and thus restricts to an operator on the space $\Omega_K(0) \coloneqq \{\omega \in \Omega_K : \omega = 0 \text{ or } \operatorname{div}(\omega) \geqslant 0\}$, which is a k-vector space whose dimension g is the genus of K. See [Sti09, Ex. 4.12-17] for these and other standard facts about the Cartier operator.

Definition 2.1. Let $\vec{\omega} \coloneqq (\omega_1, \ldots, \omega_g)$ 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 $\vec{\omega}$) is the matrix $A \coloneqq [a_{ij}] \in k^{g \times g}$.

If X/k is a smooth projective curve with function field 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 explained in [AH19], the matrices A and B are related by 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, hence have the same rank and characteristic polynomials. But we shall follow the warning/request of [AH19] and call A the Cartier–Manin matrix, although one can find examples in the literature where A is called the Hasse–Witt matrix (see [AH19] for a list).

Following Stöhr–Voloch [SV87] we 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

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

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 $\nabla\colon K \to K^p$. Writing F_y for $\frac{\partial}{\partial y} F \in k[x, y]$, for any $h \in K$ we have

$$\mathcal{C}\left(h \frac{dx}{F_y}\right) = \left(\nabla(F^{p-1} h)\right)^{1/p} \frac{dx}{F_y}, \tag{2.2}$$

by [SV87, Thm. 1.1]. If we choose a basis for $\Omega_X(0)$ using regular differentials of the form $h \frac{dx}{F_y}$, we can compute the action of the Cartier operator on this basis via (2.2). To construct such a basis, we use differentials of the form

$$\omega_{k\ell} \coloneqq x^{k-1} y^{\ell-1} \frac{dx}{F_y} \qquad (k, \ell \geqslant 1, \quad k + \ell \leqslant \deg(F) - 1). \tag{2.3}$$

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 $k,\ell \geqslant 1$ we have

$$\nabla\left(\sum_{i,j\geqslant 0} F_{ij}^{p-1} x^{i+k-1} y^{j+\ell-1}\right) = \sum_{i,j\geqslant 1} F_{ip-k,\,jp-\ell}^{p-1} x^{(i-1)p} y^{(j-1)p}. \tag{2.4}$$

Now $F_{ip-k,\,jp-\ell}^{p-1}$ is nonzero only when $(i+j)p-(k+\ell) \leqslant (p-1)\deg(F)$, and $k+\ell \leqslant \deg(F)-1$, so we can restrict the sum on the RHS to $i+j \leqslant \deg(F)-1$. From (2.2) and (2.4) we obtain

$$\mathcal{C}(\omega_{k\ell}) = \sum_{i,j\geqslant 1} \left(F_{ip-k,\,jp-\ell}^{p-1}\right)^{1/p} \omega_{ij}. \tag{2.5}$$

When X is a smooth plane curve the complete set of ω_{ij} defined in (2.3) is a basis for $\Omega_K(0)$ and we can read off the entries of the Cartier–Manin matrix A of X directly from (2.5). Following the convention in [Sut20], we order our basis $\boldsymbol{\omega} := (\omega_{ij})$ for $\Omega_k(0)$ in increasing order by j and then i, so that $\boldsymbol{\omega} = (\omega_{11}, \omega_{21}, \ldots, \omega_{12}, \ldots)$, and we view the Cartier–Manin matrix as acting on the column vector $\boldsymbol{\omega}^{\mathsf{T}}$, so that we may express (2.5) as $\mathcal{C}(\boldsymbol{\omega}^{\mathsf{T}}) = A\boldsymbol{\omega}^{\mathsf{T}}$.

If $X\colon f(x_0,x_1,x_2) = 0$ is a smooth plane quartic curve with $f(0,1,0) \neq 0$ (an assumption that will hold under non-degeneracy constraints we impose on X), then we may write its function field as $k(x)[y]/(F(x,y))$ with $x = x_0/x_2$ and $y = x_1/x_2$ so that its Cartier–Manin matrix with respect to the basis in (2.3) is

$$A = \begin{bmatrix} f_{p-1,\,p-1,\,2p-2}^{p-1} & f_{2p-1,\,p-1,\,p-2}^{p-1} & f_{p-1,\,2p-1,p-2}^{p-1} \\ f_{p-2,\,p-1,\,2p-1}^{p-1} & f_{2p-2,\,p-1,\,p-1}^{p-1} & f_{p-2,\,2p-1,p-1}^{p-1} \\ f_{p-1,\,p-2,\,2p-1}^{p-1} & f_{2p-1,\,p-2,\,p-1}^{p-1} & f_{p-1,\,2p-2,p-1}^{p-1} \end{bmatrix}, \tag{2.6}$$

where $f_{i,j,k}^{p-1}$ denotes the coefficient of the term $x_0^i x_1^j x_2^k$ in $f(x_0,x_1,x_2)^{p-1}$.

An essential property of the Cartier–Manin matrix is the identity

$$\det(I - TA) \equiv L_p(T) \bmod p, \tag{2.7}$$

where $L_p(T)$ is the numerator of the zeta function of X defined in (1.1); see [Katz73, Thm. 3.1] and [Man65, Thm. 1]. In particular, we have $\operatorname{tr} A \equiv a_p \bmod p$, where a_p is the trace of Frobenius. The Weil bounds imply $|a_p| \leqslant 2g\sqrt{p}$, which allows us to derive $\#X(\mathbb{F}_p) = p+1-a_p$ from $\operatorname{tr} A$ for all $p > 16g^2 = 144$ (for $g = 3$).

Remark 2.8. All of our algorithms compute $\#X(\mathbb{F}_p) = p+1-a_p$ by computing the Cartier–Manin matrix A and lifting $\operatorname{tr} A \in \mathbb{Z}/p\mathbb{Z}$ to the unique $a_p \in \mathbb{Z}$ with $|a_p| \leqslant 6\sqrt{p}$ when $p > 144$. For $p \leqslant 144$ we are happy to count points naïvely via (6.1).

3. Setup

Throughout this section, R denotes one of the rings $\mathbb{Z}$ or $\mathbb{F}_p$. Many of the results we use hold in greater generality, but we make no attempt to generalize them beyond the cases of interest to us here.

We write $R[x^{\pm}]$ for the Laurent polynomial ring $R[x_0, x_0^{-1}, \ldots, x_n, x_n^{-1}]$ in $n+1$ variables. We use multi-index notation: for $v := (v_0, \ldots, v_n) \in \mathbb{Z}^{n+1}$, we write x^v for the monomial $x_0^{v_0} \cdots x_n^{v_n}$. For $G \in R[x^{\pm}]$ we write G_v for the coefficient of G at the monomial x^v. We also define the degree of $v \in \mathbb{Z}^{n+1}$ to be $\deg v := \deg x^v = \sum_{i=0}^n v_i$.

For $\ell \in \mathbb{Z}$, we write $R[x^{\pm}]_\ell$ for the R-submodule of $R[x^{\pm}]$ generated by the monomials of degree ℓ. More generally, for any subset $S \subseteq \mathbb{Z}^{n+1}$, we define $R[x^{\pm}]_S$ to be the R-submodule of

Laurent polynomials supported on S, consisting of all $G \in R[x^{\pm}]$ such that $G_v = 0$ for $v \notin S$. We typically use this notation in the case that S corresponds to a finite set of monomials, all of the same degree. For $G \in R[x^{\pm}]$ we define $G|_S$, the restriction of G to S, to be the polynomial $\sum_{v \in S} G_v x^v \in R[x^{\pm}]_S$.

For any R-submodule $M \subseteq R[x^{\pm}]$, we put $M_\ell \coloneqq M \cap R[x^{\pm}]_\ell$. In particular, let $R[x]$ denote the subring $R[x_0, \dots, x_n]$; then $R[x]_\ell$ is the submodule of homogeneous polynomials of degree ℓ, or the zero submodule if $\ell < 0$. More generally, if I is a homogeneous ideal of $R[x]$, then I_ℓ is the R-submodule consisting of homogeneous polynomials of degree ℓ in I. The monomials generating $R[x]_\ell$ are indexed by the set $D_\ell \coloneqq \{v \in \mathbb{Z}_{\geqslant 0}^{n+1} : \deg v = \ell\}$ of cardinality $\#D_\ell = \dim_R R[x]_\ell = \binom{\ell+n}{n}$ for $\ell \geqslant 0$, with $D_\ell = \emptyset$ for $\ell < 0$.

We denote by K the fraction field of R, which is either $\mathbb{Q}$ or $\mathbb{F}_p$. All of the definitions for $R[x^{\pm}]$ above may be extended in the obvious way to $K[x^{\pm}]$. We write $\mathbb{P}^n_K = \operatorname{Proj} K[x]$ for projective n-space over K.

For the rest of the section we fix a homogeneous polynomial $F \in R[x]_d$ of degree $d \geqslant 2$. We always assume that $d \neq 0$ in R; in particular, if $R = \mathbb{F}_p$, then we require that $p \nmid d$. Our goal is to establish a framework for efficiently computing individual coefficients $F^m_u \coloneqq (F^m)_u$, for a prescribed integer $m \geqslant 0$, without computing the entire polynomial F^m. Our strategy will be to observe that F^m satisfies certain partial differential equations (see (3.7)), which imply various relations between nearby coefficients of F^m.

Definition 3.1. For $\ell \in \mathbb{Z}_{\geqslant 0}$ and $v \in \mathbb{Z}^{n+1}$ we define $D(v, \ell) \coloneqq \{v - w : w \in D_\ell\} \subseteq \mathbb{Z}^{n+1}$. The set $D(v, \ell)$ may be thought of as an inverted simplex of size ℓ centered at v.

We will study the vectors of coefficients of $F^m|_{D(v,\ell)}$, for certain small integers ℓ and $v \in \mathbb{Z}^{n+1}$ with $\deg v = dm + \ell$. As we will see, the differential equations lead naturally to relations among these vectors, for fixed m, as we vary v.

Remark 3.2. When $n = 2$ and F defines a smooth plane curve X in $\mathbb{P}^2_{\mathbb{F}_p}$ of genus $g = \binom{d-1}{2}$, the Cartier–Manin matrix of X consists of g^2 coefficients F^{p-1}_u with $u \in D(v, \ell)$ for g particular choices of v of degree $d(p-1) + \ell$ with $\ell = d - 3$. It turns out to be more convenient to use $m = p - 2$, as we will eventually want $d(m+1) \neq 0$ in $\mathbb{F}_p$, and to use v of degree $d(p-2) + \ell$ with $\ell = 2d - 2$. For smooth plane quartics we have $n = 2$, $d = 4$, and $\ell = nd - n = 6$, values the reader may find useful to keep in mind.

Let I_F be the homogeneous ideal $\langle \partial_0 F, \dots, \partial_n F \rangle$ in $K[x]$, where ∂_i is the degree-preserving differential operator $\partial_i \coloneqq x_i \frac{\partial}{\partial x_i}$. For $\ell \in \mathbb{Z}$, the K-vector space $K[x]_\ell/(I_F)_\ell$ is spanned by the monomials $\{x^\beta : \beta \in D_\ell\}$, so we may choose a subset $B_\ell \subseteq D_\ell$ such that $\{x^\beta : \beta \in B_\ell\}$ projects to a basis of $K[x]_\ell/(I_F)_\ell$. For the rest of the discussion, we assume a choice for B_ℓ has been fixed for each ℓ. Note that for $\ell < d$ we have $(I_F)_\ell = 0$, in which case $B_\ell = D_\ell$.

Definition 3.3. Let $b_\ell \coloneqq \dim_K K[x]_\ell/(I_F)_\ell = \#B_\ell \leqslant \#D_\ell$. For $v \in \mathbb{Z}^{n+1}$ we define the set $B(v, \ell) \coloneqq \{v - \beta : \beta \in B_\ell\} \subseteq D(v, \ell) \subseteq \mathbb{Z}^{n+1}$. We also define the K-vector spaces $\mathcal{D}_{v,\ell} \coloneqq K[x^{\pm}]_{D(v,\ell)}$ and $\mathcal{B}_{v,\ell} \coloneqq K[x^{\pm}]_{B(v,\ell)} \subseteq \mathcal{D}_{v,\ell}$.

We recall the following Hilbert series computation due to Macaulay [Ma1916].

Lemma 3.4. *Let $h_0, \dots, h_n$ be homogeneous polynomials in $K[x]$, of positive degree with no common zeros in $\mathbb{P}^n_K$. For $\ell \geqslant 0$, let*

$$\delta_\ell \coloneqq \dim_K K[x]_\ell/\langle h_0, \dots, h_n \rangle_\ell.$$

Then, in $\mathbb{Z}[t]$ we have the identity

$$\sum_{\ell\geqslant 0}\delta_\ell t^\ell = \prod_{i=0}^{n}(1+t+\cdots+t^{\deg h_i-1}).$$

Proof. See Theorem 58 in [Ma1916, pp. 64–66]. □

Recall that the discriminant $\Delta_d(F)$ of $F\in R[x]_d$ is determined up to sign by the formula

$$\Delta_d(F) = \pm d^{((-1)^{n+1}-(d-1)^{n+1})/d}\operatorname{Res}_{d-1}\Big(\frac{\partial F}{\partial x_0},\dots,\frac{\partial F}{\partial x_n}\Big),$$

where $\operatorname{Res}_e(h_0,\dots,h_n)$ is the resultant, the irreducible integer polynomial in the $(n+1)\binom{e+n}{n}$ coefficients of $h_0,\dots,h_n\in R[x]_e$ that vanishes if and only if $h_0,\dots,h_n$ have a common zero in $\mathbb{P}^n_K$ and satisfies $\operatorname{Res}_e(x_0^e,\dots,x_n^e)=1$; see [GKZ94, pp. 433–435] for details.

The hypersurface defined by $F\in R[x]_d$ is smooth if and only if $\partial F/\partial x_0,\dots,\partial F/\partial x_n$ have no common zeros in $\mathbb{P}^n_K$, that is, if and only if $\Delta_d(F)\neq 0$. (Note that any common zero of the $\partial F/\partial x_i$ is automatically a zero of F by Euler's identity $d\cdot F=\sum_i \partial_i F$, since $d\neq 0$ in R.) We say that F is nondegenerate if $\partial_0 F,\dots,\partial_n F$ have no common zeros in $\mathbb{P}^n_K$. Nondegeneracy of F is equivalent to requiring that the intersection of the hypersurface defined by F with every set of coordinate hyperplanes is smooth (see [Bat93, Prop. 4.6], [CV09, Prop. 1.2]); this implies that the hypersurface defined by F is smooth, but it is a stronger condition. If we let $D_d(S) \coloneqq \{v\in D_d : v_i=0 \text{ for } i\in S\}$ and define

$$\Delta^*_d(F) \coloneqq \prod_{S\subsetneq\{0,\dots,n\}} \Delta_d\big(F|_{D_d(S)}\big), \tag{3.5}$$

where the discriminants on the right are taken with respect to the variables not in S, then we see that F is nondegenerate if and only if $\Delta^*_d(F)\neq 0$.

For $n=1$ we have $\Delta^*_d(F) = \pm F_{0,d}F_{d,0}\Delta_d(F) = \pm F_{0,d}F_{d,0}\operatorname{disc}F(t,1)$, where disc denotes the usual discriminant of a univariate polynomial in $R[t]$. For $n=2$ we have

$$\Delta^*_d(F) = \pm F_{0,0,d}F_{0,d,0}F_{d,0,0}\operatorname{disc}F(t,1,0)\operatorname{disc}F(t,0,1)\operatorname{disc}F(0,t,1)\Delta_d(F).$$

Let $H_F(t)\coloneqq\sum_{\ell\geqslant 0} b_\ell t^\ell\in\mathbb{Z}[t]$ denote the Hilbert series of the quotient ring $K[x]/I_F$.

Corollary 3.6. *If $F\in R[x]_d$ is nondegenerate then*

$$H_F(t)\coloneqq\sum_{\ell\geqslant 0} b_\ell t^\ell = \big(1+t+\cdots+t^{d-1}\big)^{n+1},$$

and we have $\sum_{\ell\equiv k \bmod d} b_\ell = d^n$ for any integer k.

Proof. The first claim follows from Lemma 3.4. For the second, fix $k\in\mathbb{Z}$ and let ζ be a primitive dth root of unity. We have

$$\sum_{i=0}^{d-1} H_F(\zeta^i)\zeta^{-ki} = \sum_{i=0}^{d-1}\sum_{\ell\geqslant 0} b_\ell\zeta^{(\ell-k)i} = d\sum_{\ell\equiv k \bmod d} b_\ell,$$

and also

$$\sum_{i=0}^{d-1} H_F(\zeta^i)\zeta^{-ki} = \sum_{i=0}^{d-1}(1+\zeta^i+\cdots+(\zeta^i)^{d-1})^{n+1}\zeta^{-ki} = d^{n+1}.$$

Comparing these two expressions yields the desired result. □

Let $m \geqslant 0$ and consider the system of differential equations for $G \in K[x^{\pm}]_{dm}$ given by

$$\partial_i(FG) = (m+1)(\partial_i F)G, \qquad i = 0, \ldots, n. \tag{3.7}$$

The scalar multiples of F^m are solutions to (3.7). Note that the Euler identity

$$\sum_{i=0}^{n} \partial_i(FG) = d(m+1)FG = (m+1)\sum_{i=0}^{n}(\partial_i F)G \tag{3.8}$$

implies that one of these $n+1$ equations is redundant, so for many purposes we may treat it as a system of only n equations.

We now show that (3.7) defines a system of linear equations on the coefficients of G. For any $w \in \mathbb{Z}^{n+1}$ of degree $dm+d$, equating coefficients in (3.7) for the monomial x^w gives rise to the system of linear equations

$$w_i \sum_{t \in D_d} F_t G_{w-t} = (m+1)\sum_{t \in D_d} t_i F_t G_{w-t}, \qquad i = 0, \ldots, n. \tag{3.9}$$

Via (3.8) we may view this as a system of n equations in $\#D_d$ unknowns G_u for $u \in D(w,d)$.

More generally, for any $\ell \geqslant d$ and $v \in \mathbb{Z}^{n+1}$ of degree $dm+\ell$ we may consider the system of linear equations involving the coefficients G_u for $u \in D(v,\ell)$, obtained by including the equations (3.9) for each $w \in D(v, \ell-d)$. Here we are using the fact that $D(v,\ell)$ is the union of the sets $D(w,d)$ as w ranges over $D(v,\ell-d)$. Explicitly, these equations are given by

$$(v_i - s_i)\sum_{t \in D_d} F_t G_{v-s-t} = (m+1)\sum_{t \in D_d} t_i F_t G_{v-s-t}, \qquad s \in D_{\ell-d},\ i = 0, \ldots, n. \tag{3.10}$$

Via (3.8) we view this as a system of $n\#D_{\ell-d}$ equations in $\#D_\ell$ unknowns G_u for $u \in D(v,\ell)$.

Definition 3.11. Let $\mathcal{E}_{v,\ell}$ denote the K-vector subspace of $\mathcal{D}_{v,\ell} = K[x^{\pm}]_{D(v,\ell)}$ consisting of those Laurent polynomials $G \in \mathcal{D}_{v,\ell}$ satisfying the system (3.10).

Note that $\mathcal{E}_{v,\ell}$ is only defined when $\deg v$ is of the form $dm+\ell$ for some $m \geqslant 0$. The value of m is implicitly defined by v and ℓ: we always have $m = (\deg v - \ell)/d$, so a choice of v and ℓ determines a choice of m.

Since F^m satisfies the original differential equations (3.7), we see immediately that

$$F^m|_{D(v,\ell)} \in \mathcal{E}_{v,\ell}.$$

We also have the following basic result concerning inclusions of sets of the form $D(v,\ell)$.

Lemma 3.12. *Let $\ell, \ell' \geqslant d$ and let $v, v' \in \mathbb{Z}^{n+1}$ have degrees $dm+\ell$ and $dm+\ell'$ respectively. Assume that $D(v,\ell) \subseteq D(v',\ell')$. Then the restriction map $\mathcal{D}_{v',\ell'} \twoheadrightarrow \mathcal{D}_{v,\ell}$, $G \mapsto G|_{D(v,\ell)}$, maps $\mathcal{E}_{v',\ell'}$ into $\mathcal{E}_{v,\ell}$.*

Proof. The equations defining $\mathcal{E}_{v,\ell}$ are a subset of those defining $\mathcal{E}_{v',\ell'}$. □

In the remainder of this section we develop further properties of the vector spaces $\mathcal{E}_{v,\ell}$. In particular, we compute their dimension and give explicit bases for certain cases of interest.

Lemma 3.13. *Let $\ell \geqslant d$, and let $v \in \mathbb{Z}^{n+1}$ be of degree $dm+\ell$. Consider the K-linear map*

$$\begin{aligned} \pi_{v,\ell} \colon \mathcal{D}_{v,\ell} &\longrightarrow \mathcal{B}_{v,\ell} \oplus \mathcal{D}_{v,\ell-d}, \\ G &\longmapsto G|_{B(v,\ell)} + \big(FG\big)|_{D(v,\ell-d)}. \end{aligned}$$

The map $\pi_{v,\ell}$ may be represented by a matrix whose entries lie in R and are independent of v.

Moreover, there exists a nonzero constant $\lambda_\ell \in R$ and a K-linear map

$$\psi_{v,\ell} \colon \mathcal{B}_{v,\ell} \oplus \mathcal{D}_{v,\ell-d} \to \mathcal{D}_{v,\ell}$$

such that the composition

$$\psi_{v,\ell} \circ \pi_{v,\ell} \colon \mathcal{D}_{v,\ell} \to \mathcal{D}_{v,\ell}$$

restricts to scalar multiplication by $(m+1)\lambda_\ell$ on $\mathcal{E}_{v,\ell}$. The map $\psi_{v,\ell}$ may be represented by a matrix whose entries are R-linear combinations of $1, v_0, \dots, v_n$ and m, which we may view as polynomials in $R[v,m] = R[v_0, \dots, v_n, m]$ of degree at most 1.

Note that when using matrices to represent maps such as $\pi_{v,\ell}$ and $\psi_{v,\ell}$, we always work with respect to the obvious monomial bases. For example, the columns of $\pi_{v,\ell}$ are indexed by D_ℓ, and its rows are indexed by the concatenation of B_ℓ and $D_{\ell-d}$. For this purpose we assume that some ordering of the monomials of each degree has been chosen, such as the lexicographical ordering.

Remark 3.14. One may think of $\pi_{v,\ell}$ as "compressing" a vector of length $\#D_\ell$ into a vector of length $\#B_\ell + \#D_{\ell-d}$. If the input vector lies in the subspace $\mathcal{E}_{v,\ell}$, i.e., satisfies the appropriate differential equations, then no information is lost in the compression, and $\psi_{v,\ell}$ "decompresses" the result to recover the original vector (multiplied by a certain scalar).

Proof. We observe that $\pi_{v,\ell}$ may be represented by a matrix in which the rows corresponding to $\mathcal{B}_{v,\ell}$ have entries in $\{0,1\}$, and the entries of the rows corresponding to $\mathcal{D}_{v,\ell-d}$ are either zero or of the form F_u for some $u \in D_d$ with $(FG)_{v-w} = \sum_{u \in D_d} F_u G_{v-w-u}$ for $w \in D(v, \ell-d)$. This matrix is the same for every $v \in \mathbb{Z}^{n+1}$ of degree $dm+\ell$.

We now explain how to construct $\psi_{v,\ell}$. Our task is to construct a formula that recovers a polynomial $G \in \mathcal{E}_{v,\ell}$ from knowledge of $G|_{B(v,\ell)}$ and $(FG)|_{D(v,\ell-d)}$.

First, it follows from the definition of B_ℓ that for any $u \in D_\ell$ we may write

$$\lambda_\ell x^u = \sum_{i=0}^{n} h_{u,i} \partial_i F + \sum_{\beta \in B_\ell} c_{u,\beta} x^\beta, \tag{3.15}$$

for some $\lambda_\ell, c_{u,\beta} \in R$ ($\lambda_\ell \neq 0$) and $h_{u,i} \in R[x]_{\ell-d}$. (For $u \in B_\ell \subseteq D_\ell$ we may take $h_{u,i} = 0$, $c_{u,u} = \lambda_\ell$, and $c_{u,\beta} = 0$ for $\beta \neq u$.)

Now suppose that $G \in \mathcal{E}_{v,\ell}$. Multiplying both sides of (3.15) by $(m+1)G$ and equating coefficients of x^v yields

$$(m+1)\lambda_\ell G_{v-u} = \sum_{i=0}^{n} \sum_{s \in D_{\ell-d}} (m+1)(h_{u,i})_s \big((\partial_i F)G\big)_{v-s} + (m+1) \sum_{\beta \in B_\ell} c_{u,\beta} G_{v-\beta}$$

for each $u \in D_\ell$. By assumption G satisfies (3.10), so

$$(m+1)((\partial_i F)G)_{v-s} = (\partial_i (FG))_{v-s} = (v_i - s_i)(FG)_{v-s} \tag{3.16}$$

for all $s \in D_{\ell-d}$ and $i = 0, \dots, n$. Therefore, for each $u \in D_\ell$,

$$(m+1)\lambda_\ell G_{v-u} = \sum_{i=0}^{n} \sum_{s \in D_{\ell-d}} (v_i - s_i)(h_{u,i})_s (FG)_{v-s} + (m+1) \sum_{\beta \in B_\ell} c_{u,\beta} G_{v-\beta}. \tag{3.17}$$

The right hand side of (3.17) involves the coefficients of FG on $D(v, \ell - d)$ and the coefficients of G on $B(v, \ell)$, so we may use this expression to define $\psi_{v,\ell}$. Explicitly, for $H \in \mathcal{B}_{v,\ell}$ and $J \in \mathcal{D}_{v,\ell-d}$ we define $\psi_{v,\ell}(H + J) \in \mathcal{D}_{v,\ell}$ via

$$\psi_{v,\ell}(H+J)_{v-u} \coloneqq \sum_{i=0}^{n} \sum_{s \in D_{\ell-d}} (v_i - s_i)(h_{u,i})_s J_{v-s} + (m+1) \sum_{\beta \in B_\ell} c_{u,\beta} H_{v-\beta}. \tag{3.18}$$

It is clear that the entries of the corresponding matrix are polynomials of degree at most 1 in $v_0, \dots, v_n, m$ with coefficients in R. By construction, if $G \in \mathcal{E}_{v,\ell}$, then (3.17) implies that

$$\begin{aligned}
\psi_{v,\ell}(\pi_{v,\ell}(G))_{v-u} &= \psi_{v,\ell}\Big(G|_{B(v,\ell)} + \big(FG\big)|_{D(v,\ell-d)}\Big)_{v-u} \\
&= \sum_{i=0}^{n} \sum_{s \in D_{\ell-d}} (v_i - s_i)(h_{u,i})_s (FG)_{v-s} + (m+1) \sum_{\beta \in B_\ell} c_{u,\beta} G_{v-\beta} \\
&= (m+1)\lambda_\ell G_{v-u}
\end{aligned}$$

for $u \in D_\ell$. Thus $\psi_{v,\ell} \circ \pi_{v,\ell}$ restricts to scalar multiplication by $(m+1)\lambda_\ell$ on $\mathcal{E}_{v,\ell}$. □

Definition 3.19. We define $\mathcal{W}_{v,\ell} \coloneqq \mathcal{B}_{v,\ell} \oplus \mathcal{B}_{v,\ell-d}$. For $\ell < 2d$ this is the codomain of $\pi_{v,\ell}$ and the domain of $\psi_{v,\ell}$, since $B(v, \ell - d) = D(v, \ell - d)$ for $\ell - d < d$.

Corollary 3.20. *Let $d \leqslant \ell < 2d$ and $v \in \mathbb{Z}^{n+1}$ of degree $dm + \ell$. Assume that $m \neq -1$ in R. Then*

$$\dim_K \mathcal{E}_{v,\ell} \leqslant \dim_K \mathcal{W}_{v,\ell} = b_\ell + b_{\ell-d}, \tag{3.21}$$

and if F is nondegenerate then we have $\dim_K \mathcal{E}_{v,\ell} \leqslant d^n$.

When equality holds in (3.21) we may restrict the domain of $\pi_{v,\ell}$ and the codomain of $\psi_{v,\ell}$ to obtain K-linear isomorphisms

$$\pi^{\mathcal{E}}_{v,\ell} \colon \mathcal{E}_{v,\ell} \to \mathcal{W}_{v,\ell}, \qquad \psi^{\mathcal{E}}_{v,\ell} \colon \mathcal{W}_{v,\ell} \to \mathcal{E}_{v,\ell}.$$

Proof. As noted above, the hypothesis $\ell < 2d$ ensures that the codomain of $\pi_{v,\ell}$ and domain of $\psi_{v,\ell}$ are both equal to $\mathcal{W}_{v,\ell}$. Let λ_ℓ be as in Lemma 3.13. Since $(m+1)\lambda_\ell \neq 0$ in R, Lemma 3.13 implies that the map $\pi_{v,\ell}$ is injective when restricted to $\mathcal{E}_{v,\ell}$ (since scalar multiplication by $(m+1)\lambda_\ell$ is injective), and the first inequality follows. The equality in (3.21) is simply the observation that $\dim_K \mathcal{W}_{v,\ell} = \#B(v,\ell) + \#B(v,\ell-d) = \#B_\ell + \#B_{\ell-d} = b_\ell + b_{\ell-d}$. If F is nondegenerate, then by Corollary 3.6 we have $b_\ell + b_{\ell-d} \leqslant \sum_{\ell' \equiv \ell \bmod d} b_{\ell'} = d^n$.

Suppose now that equality holds in (3.21), so $\dim_K \mathcal{E}_{v,\ell} = \dim_K \mathcal{W}_{v,\ell}$. Let $\pi^{\mathcal{E}}_{v,\ell} \colon \mathcal{E}_{v,\ell} \to \mathcal{W}_{v,\ell}$ be the restriction of $\pi_{v,\ell}$ to $\mathcal{E}_{v,\ell}$. As shown in the previous paragraph, $\pi^{\mathcal{E}}_{v,\ell}$ is injective, and by comparing dimensions we see that it is an isomorphism onto $\mathcal{W}_{v,\ell}$. Then, since $\psi_{v,\ell} \circ \pi^{\mathcal{E}}_{v,\ell} \colon \mathcal{E}_{v,\ell} \to \mathcal{D}_{v,\ell}$ is injective (by Lemma 3.13) it follows that $\psi_{v,\ell}$ is injective. The image of $\psi_{v,\ell}$ contains $\mathcal{E}_{v,\ell}$ (again by Lemma 3.13), and by comparing dimensions we find that its image is equal to $\mathcal{E}_{v,\ell}$. Restricting the codomain of $\psi_{v,\ell}$ then yields the desired isomorphism $\psi^{\mathcal{E}}_{v,\ell} \colon \mathcal{W}_{v,\ell} \to \mathcal{E}_{v,\ell}$. □

Corollary 3.22. *Let $n = 2$, $\ell \in \{2d-2, 2d-1\}$, and $v \in \mathbb{Z}^{n+1}$ of degree $dm + \ell$. Then $\dim_K \mathcal{E}_{v,\ell} \geqslant d^2$, and if F is nondegenerate and $m \neq -1$ in R, then $\dim_K \mathcal{E}_{v,\ell} = b_\ell + b_{\ell-d} = d^2$.*

Proof. Recall that $\mathcal{E}_{v,\ell}$ is defined by a system of $n\#D_{\ell-d}$ equations in $\#D_\ell$ unknowns. Its dimension is therefore at least $\#D_\ell - n\#D_{\ell-d} = \binom{\ell+n}{n} - n\binom{\ell-d+n}{n}$, which is precisely d^2 for

$n = 2$ and $\ell \in \{2d-2, 2d-1\}$, in which case $\dim_K \mathcal{E}_{v,\ell} \geqslant d^2$. If additionally F is nondegenerate and $m \neq -1$ in R, then Corollary 3.20 and Corollary 3.6 imply that $\dim_K \mathcal{E}_{v,\ell} \leqslant b_\ell + b_{\ell-d} \leqslant d^2$, so we conclude that $\dim_K \mathcal{E}_{v,\ell} = b_\ell + b_{\ell-d} = d^2$. □

Remark 3.23. Corollaries 3.20 and 3.22 explain why we use $m = p-2$ rather than $m = p-1$ when computing Cartier–Manin matrices: we want $(m+1)\lambda_\ell$ to be nonzero in characteristic p.

Remark 3.24. We expect that generalizations of Corollary 3.22 for $n > 2$ also hold, that is, $\dim \mathcal{E}_{v,\ell} = d^n$ for F nondegenerate and ℓ large enough. However, a simple dimension count no longer shows that $\pi_{v,\ell}$ is surjective, more is needed.

4. Shifting coefficients

To simplify the exposition we now specialize to the case $n = 2$. As in the previous section, R is $\mathbb{Z}$ or $\mathbb{F}_p$, K is its fraction field, $R[x^{\pm}]$ is the Laurent polynomial ring in $n+1 = 3$ variables x_0, x_1, x_2, $R[x]$ is the subring $R[x_0, x_1, x_2]$, and we work with a fixed homogeneous polynomial $F \in R[x]_d$ of degree $d > 1$ and a positive integer m such that $d(m+1) \neq 0$ in R (we will take $m = p-2$ when $R = \mathbb{F}_p$). We assume throughout that F is nondegenerate, i.e., that $\Delta_d^*(F) \neq 0$ (see (3.5) for the definition of $\Delta_d^*(F)$).

Let e_0, e_1, e_2 be the standard basis for $\mathbb{Z}^3$. In this section we consider how to shift a solution to (3.10) from $D(v, \ell)$ to $D(v + e_i - e_j, \ell)$, for $\ell = 2d-2$ and $v \in \mathbb{Z}^3$ of degree $dm + \ell$, where $i, j \in \{0, 1, 2\}$ with $i \neq j$. Our goal is to construct a "shift" map

$$\tau_{v,i,j} \colon \mathcal{D}_{v,\ell} \to \mathcal{D}_{v+e_i-e_j,\ell},$$

illustrated in the top row of Figure 1, with two key properties:

(1) For any $G \in \mathcal{D}_{v,\ell}$, the coefficients of G and $\tau_{v,i,j}(G)$ should agree on the intersection $D(v,\ell) \cap D(v+e_i-e_j,\ell) = D(v-e_j,\ell-1)$, up to multiplication by a known nonzero scalar. The region $D(v-e_j,\ell-1)$ is indicated by the dotted lines in Figure 1.

(2) $\tau_{v,i,j}$ should restrict to a map

$$\tau^{\mathcal{E}}_{v,i,j} \colon \mathcal{E}_{v,\ell} \to \mathcal{E}_{v+e_i-e_j,\ell},$$

i.e., if $G \in \mathcal{D}_{v,\ell}$ satisfies the differential equations on $D(v,\ell)$, then the shifted polynomial $\tau_{v,i,j}(G)$ satisfies the equations on $D(v+e_i-e_j,\ell)$.

It will be convenient to reformulate the first condition as follows. For any $\ell' \geqslant 1$, $w \in \mathbb{Z}^3$ and $k \in \{0,1,2\}$ let

$$P_{w,\ell',k} \colon \mathcal{D}_{w,\ell'} \twoheadrightarrow \mathcal{D}_{w-e_k,\ell'-1}$$

denote the restriction map $G \mapsto G|_{D(w-e_k,\ell'-1)}$ induced by the inclusion $D(w-e_k,\ell'-1) \subseteq D(w,\ell')$. Then condition (1) is equivalent to requiring $P_{v+e_i-e_j,\ell,i} \circ \tau_{v,i,j} \colon \mathcal{D}_{v,\ell} \to \mathcal{D}_{v-e_j,\ell-1}$ to be a nonzero scalar multiple of $P_{v,\ell,j} \colon \mathcal{D}_{v,\ell} \to \mathcal{D}_{v-e_j,\ell-1}$.

Remark 4.1. Later we will apply this framework to $G = F^m|_{D(v,\ell)}$. It is easy to compute $F^m|_{D(v,\ell)}$ when v is near dme_k, i.e., at the corners of the simplex. By repeatedly applying the $\tau_{v,i,j}$ maps, we may shift this solution to obtain $F^m|_{D(v,\ell)}$ for a given target value of v. For certain carefully chosen v, the components of these vectors will in turn yield the entries of the Cartier–Manin matrix of the smooth plane quartic defined by F, when $d = 4$, $\ell = 6$ and $m = p-2$. These shifts are illustrated in Figure 2.

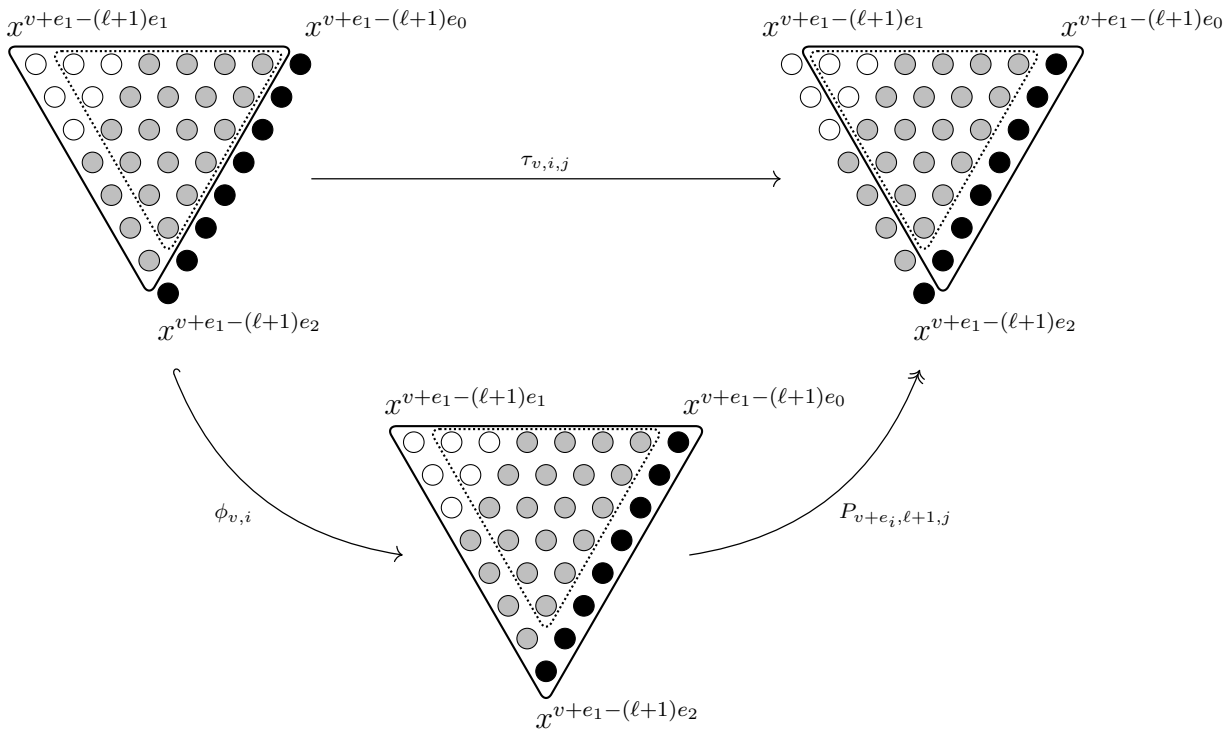

FIGURE 1. Illustration of the maps $\phi_{v,i}$ and $\tau_{v,i,j}$ for $d = 4$, $\ell = 6$, $i = 1$, $j = 0$. The common domain $\mathcal{D}(v, \ell)$ of $\tau_{v,i,j}$ and $\phi_{v,i}$ is represented by the white and gray dots enclosed in the upper left triangle (the dots represent a monomial basis). The codomain $\mathcal{D}(v + e_i - e_j, \ell)$ of $\tau_{v,i,j}$ is represented by the subset of white, gray, and black dots enclosed in the upper right triangle, and the codomain $\mathcal{D}(v + e_i, \ell + 1)$ of $\phi_{v,i}$ is the entire bottom triangle, which contains both $\mathcal{D}(v, \ell)$ and $\mathcal{D}(v + e_i - e_j, \ell)$. As shown in the proof of Lemma 4.4, the coordinates in the codomain of $\phi_{v,i}$ represented by the black dots are determined by the coordinates represented by the gray dots.

By composing $\phi_{v,i}$ with the projection $P_{v+e_i,\ell+1,j} \colon \mathcal{D}_{v+e_i,\ell+1} \twoheadrightarrow \mathcal{D}_{v+e_i-e_j,\ell}$ we obtain the desired map $\tau_{v,i,j}$, as shown in the following commutative diagram:

$$\begin{array}{ccc} \mathcal{D}_{v,\ell} & \xrightarrow{\tau_{v,i,j}} & \mathcal{D}_{v+e_i-e_j,\ell}\,. \\ & {\scriptstyle \phi_{v,i}} \searrow \quad \nearrow {\scriptstyle P_{v+e_i,\ell+1,j}} & \\ & \mathcal{D}_{v+e_i,\ell+1} & \end{array} \tag{4.2}$$

See Figure 1 for an illustration of this diagram in the case $d = 4$.

$$\begin{array}{ccc} \mathcal{E}_{v,\ell} & \xrightarrow{\tau^{\mathcal{E}}_{v,i,j}} & \mathcal{E}_{v+e_i-e_j,\ell}\,. \\ & {\scriptstyle \phi^{\mathcal{E}}_{v,i}} \searrow \quad \nearrow {\scriptstyle P^{\mathcal{E}}_{v+e_i,j}} & \\ & \mathcal{E}_{v+e_i,\ell+1} & \end{array} \tag{4.3}$$

The first step in defining $\tau_{v,i,j}$ is to construct an "extension" map $\phi_{v,i}\colon \mathcal{D}_{v,\ell} \to \mathcal{D}_{v+e_i,\ell+1}$ that extends G from $D(v,\ell)$ to the larger set $D(v+e_i,\ell+1)$. This is carried out in Lemma 4.4 below. The idea is to explicitly solve the system (3.10) for the unknown coefficients of $\phi_{v,i}(G)$, i.e., for the monomials in $D(v+e_i,\ell+1) \setminus D(v,\ell)$. These are shown as the black dots in Figure 1.

We remind the reader that $n = 2$, $d > 1$, $\ell = 2d-2$, $d(m+1)$ is nonzero in R, and $\Delta_d^*(F) \neq 0$. In particular, $\Delta_d^*(F_{x_i=0}) \neq 0$, since the latter is a factor of $\Delta_d^*(F)$; see (3.5).

Lemma 4.4. *Let $v \in \mathbb{Z}^3$ be of degree $dm+\ell$, let $i \in \{0,1,2\}$, and let $\theta_i \coloneqq \pm\Delta_d^*(F_{x_i=0}) \neq 0$. There exists a K-linear map*

$$\phi_{v,i}\colon \mathcal{D}_{v,\ell} \to \mathcal{D}_{v+e_i,\ell+1}$$

such that $P_{v+e_i,\ell+1,i} \circ \phi_{v,i} = (v_i+1)\theta_i \cdot \mathrm{id}_{\mathcal{D}_{v,\ell}}$, and such that if $v_i+1 \neq 0$ in R then $\phi_{v,i}(\mathcal{E}_{v,\ell}) \subseteq \mathcal{E}_{v+e_i,\ell+1}$.

The map $\phi_{v,i}$ may be represented by a $\binom{2d+1}{2} \times \binom{2d}{2}$ matrix whose entries are R-linear combinations of $1, v_0, v_1, v_2$ and m, which may be viewed as linear polynomials in $R[v,m] = R[v_0,v_1,v_2,m]$.

Remark 4.5. The sign of θ_i is not canonically determined; it depends on choices made during the following proof (such as the choice of j and k). An explicit formula for θ_i, as the determinant of a certain Sylvester matrix, is given in (4.11).

Proof. We are given as input $G \in \mathcal{D}_{v,\ell}$, and we wish to extend it to some $\widetilde{G} \in \mathcal{D}_{v+e_i,\ell+1}$. We first set $\widetilde{G}_w \coloneqq G_w$ for $w \in D(v,\ell)$. Let

$$S \coloneqq D(v+e_i,\ell+1) \setminus D(v,\ell).$$

Our task is to show how to define the missing coefficients $\widetilde{G}_w$ for $w \in S$ in such a way that $\widetilde{G} \in \mathcal{E}_{v+e_i,\ell+1}$ whenever $G \in \mathcal{E}_{v,\ell}$. These $2d$ coefficients are indicated by the black dots in Figure 1. We can alternatively write S as

$$S = \big\{(v+e_i) - (ce_j + (2d-1-c)e_k) : 0 \leqslant c \leqslant 2d-1\big\}$$

where j and k are chosen so that $\{j,k\} = \{0,1,2\} \setminus \{i\}$.

According to (3.10), $\widetilde{G}$ lies in $\mathcal{E}_{v+e_i,\ell+1}$ if and only if

$$((v+e_i)_h - s_h) \sum_{t \in D_d} F_t \widetilde{G}_{v+e_i-s-t} = (m+1) \sum_{t \in D_d} t_h F_t \widetilde{G}_{v+e_i-s-t} \tag{4.6}$$

for all $s \in D_{\ell+1-d}$ and $h = 0,1,2$. Consider the subset of equations in (4.6) corresponding to those s with $s_i \geqslant 1$, i.e., for those $s = s' + e_i$ with $s' \in D_{\ell-d}$:

$$(v_h - s'_h) \sum_{t \in D_d} F_t \widetilde{G}_{v-s'-t} = (m+1) \sum_{t \in D_d} t_h F_t \widetilde{G}_{v-s'-t}, \qquad s' \in D_{\ell-d},\ h = 0,1,2.$$

These equations only involve $\widetilde{G}_w$ for $w \in D(v,\ell)$, and in fact are exactly the equations defining $\mathcal{E}_{v,\ell}$. If $G \in \mathcal{E}_{v,\ell}$, then $\widetilde{G}$ automatically satisfies these equations, since we already arranged that $\widetilde{G}_w = G_w$ for $w \in D(v,\ell)$. The remaining equations correspond to those $s \in D_{\ell+1-d} = D_{d-1}$ for which $s_i = 0$, i.e., to $s \in E$ where

$$E \coloneqq \{ae_j + (d-1-a)e_k : 0 \leqslant a \leqslant d-1\}.$$

Consequently, for $\widetilde{G}$ to lie in $\mathcal{E}_{v+e_i,\ell+1}$, it suffices to choose $\widetilde{G}_w$ for $w \in S$ so that (4.6) holds for all $s \in E$ and $h = 0, 1, 2$. Moreover, we recall that one value of h is redundant, thanks to the Euler identity (3.8). Taking $h = i$ and $h = j$, this system of $2|E| = 2d$ equations is given explicitly by

$$\begin{aligned}(v_i+1)\sum_{t\in D_d} F_t\widetilde{G}_{v+e_i-s-t} &= (m+1)\sum_{t\in D_d} t_iF_t\widetilde{G}_{v+e_i-s-t}, \qquad s\in E,\\ (v_j-s_j)\sum_{t\in D_d} F_t\widetilde{G}_{v+e_i-s-t} &= (m+1)\sum_{t\in D_d} t_jF_t\widetilde{G}_{v+e_i-s-t}, \qquad s\in E.\end{aligned} \tag{4.7}$$

Let us manipulate these equations to put them into a more useful form. For each s, multiply the second equation by $v_i + 1$, subtract $v_j - s_j$ times the first equation, and divide by $m + 1 \neq 0$, to obtain the system

$$\begin{aligned}(v_i+1)\sum_{t\in D_d} F_t\widetilde{G}_{v+e_i-s-t} &= (m+1)\sum_{t\in D_d} t_iF_t\widetilde{G}_{v+e_i-s-t}, \quad s\in E,\\ \sum_{t\in D_d}\big((v_i+1)t_j-(v_j-s_j)t_i\big)F_t\widetilde{G}_{v+e_i-s-t} &= 0, \qquad\qquad\qquad\qquad s\in E.\end{aligned} \tag{4.8}$$

The system (4.8) is equivalent to (4.7), provided that $v_i + 1 \neq 0$. Now we rearrange so that the terms with $t_i = 0$ appear on the left hand side:

$$\begin{aligned}(v_i+1)\sum_{\substack{t\in D_d\\ t_i=0}} F_t\widetilde{G}_{v+e_i-s-t} &= \sum_{\substack{t\in D_d\\ t_i\neq 0}}\big((m+1)t_i-(v_i+1)\big)F_t\widetilde{G}_{v+e_i-s-t}, \qquad s\in E,\\ (v_i+1)\sum_{\substack{t\in D_d\\ t_i=0}} t_jF_t\widetilde{G}_{v+e_i-s-t} &= \sum_{\substack{t\in D_d\\ t_i\neq 0}}\big((v_j-s_j)t_i-(v_i+1)t_j\big)F_t\widetilde{G}_{v+e_i-s-t}, \quad s\in E.\end{aligned} \tag{4.9}$$

We may rewrite the system (4.9) in matrix form as follows.

- The coefficients $\widetilde{G}_w$ on the left hand side are exactly the unknowns of interest: writing $t = be_j + (d-b)e_k$ for $0 \leqslant b \leqslant d$ and $s = ae_j + (d-1-a)e_k$ for $0 \leqslant a \leqslant d-1$, we see that $w = v + e_i - s - t = (v + e_i) - ce_j - (2d-1-c)e_k \in S$ for $c = a + b$. Let $y \in K^{2d}$ represent this vector of unknowns, with $y_c = \widetilde{G}_{v+e_i-ce_j-(2d-1-c)e_k}$ for $0 \leqslant c \leqslant 2d-1$.
- The coefficients $\widetilde{G}_w$ on the right hand side are shown as the gray dots in Figure 1. These coefficients are already known, i.e., all such w lie in $D(v, \ell)$, so that $\widetilde{G}_w = G_w$. Indeed, if $t = t' + e_i$ for $t' \in D_{d-1}$, then $w = v + e_i - s - t = v - s - t' \in D(v, (d-1)+(d-1)) = D(v, \ell)$. Let $z \in K^{\binom{2d}{2}}$ be the vector consisting of all G_w for $w \in D(v, \ell)$, for some convenient ordering of $D(v, \ell)$.
- Let $\bar{F}_b \coloneqq F_{be_j+(d-b)e_k}$ for $0 \leqslant b \leqslant d$; these are the coefficients F_t appearing on the left hand side of (4.9). Let A be the $2d \times 2d$ matrix (over R) given by

$$A = \begin{pmatrix} \bar{F}_0 & \bar{F}_1 & \bar{F}_2 & \cdots & \cdots & \bar{F}_d & & \\ & \ddots & & & & & \ddots & \\ & & \bar{F}_0 & \bar{F}_1 & \bar{F}_2 & \cdots & \cdots & \bar{F}_d \\ 0 & \bar{F}_1 & 2\bar{F}_2 & \cdots & \cdots & d\bar{F}_d & & \\ & \ddots & & & & & \ddots & \\ & & 0 & \bar{F}_1 & 2\bar{F}_2 & \cdots & \cdots & d\bar{F}_d \end{pmatrix}.$$

The columns correspond to the unknowns y_c for $0 \leqslant c \leqslant 2d-1$. The first group of d rows corresponds to the first equation in (4.9), and the second group to the second equation. The rows in each group are indexed by $a = 0, \ldots, d-1$, corresponding to the values of $s \in E$ via $s = ae_j + (d-1-a)e_k$.

- Let $M_{v,m}$ be the $2d \times \binom{2d}{2}$ matrix encoding the linear combinations on the right hand side of (4.9). The columns of $M_{v,m}$ correspond to the known values G_w for $w \in D(v,\ell)$, and the rows to the $2d$ equations. More explicitly, in the first d rows, indexed by $a = 0, \ldots, d-1$, we place the value $(m+1)t_i - (v_i+1)$ in the column corresponding to $v + e_i - s - t$ for each $t = t' + e_i$, $t' \in D_{d-1}$. Similarly, in the last d rows, we place the values $(v_j - s_j)t_i - (v_i+1)t_j$ in suitable positions. The entries of $M_{v,m}$ may be regarded as linear polynomials in $R[v,m]$.

With these definitions, the system (4.9) may be expressed compactly as

$$(v_i+1)Ay = M_{v,m}z. \tag{4.10}$$

The matrix A is the Sylvester matrix of $F_{x_i=0,x_k=1}$ and $(\partial_j F)_{x_i=0,x_k=1}$ as degree d polynomials in x_j. By Proposition 1.8 in [GKZ94, p. 435] we have

$$\det A = \pm F_{de_j} F_{de_k} \operatorname{disc}_{x_j} F_{x_i=0,x_k=1} = \pm \Delta_d^*(F_{x_i=0}) \neq 0.$$

We may therefore solve the system explicitly as follows. Define

$$\theta_i := \det A, \tag{4.11}$$

and let $\operatorname{adj}(A) \in R^{2d \times 2d}$ denote the matrix satisfying $\operatorname{adj}(A)A = (\det A)I$. Multiplying (4.10) by $\operatorname{adj}(A)$ on the left yields the solution

$$(v_i+1)\theta_i y = \operatorname{adj}(A) M_{v,m} z.$$

Note that the columns of $\operatorname{adj}(A)M_{v,m}$ correspond to monomials $u \in D(v,\ell)$, and the rows correspond to monomials $w \in S \subseteq D(v+e_i, \ell+1)$, i.e., the c-th row corresponds to $w = v + e_i - ce_j - (2d-1-c)e_k$ for $0 \leqslant c \leqslant 2d-1$.

Finally we show how to define the matrix for the desired map $\phi_{v,i} \colon \mathcal{D}_{v,\ell} \to \mathcal{D}_{v+e_i,\ell+1}$. For $w \in D(v+e_i, \ell+1)$ and $u \in D(v,\ell)$, the matrix entry $(\phi_{v,i})_{w,u}$ is given by

$$(\phi_{v,i})_{w,u} = \begin{cases} (v_i+1)\theta_i \delta_{w,u}, & \text{if } w \in D(v,\ell), \\ (\operatorname{adj}(A) M_{m,v})_{w,u}, & \text{if } w \notin D(v,\ell), \end{cases} \tag{4.12}$$

where $\delta_{w,u}$ if $w = u$ and 0 otherwise. □

Remark 4.13. One may attempt to apply the construction in the proof of Lemma 4.4 for values of ℓ other than $2d-2$. This leads to a system of $2(\ell - d + 2)$ equations in $\ell + 2$ unknowns. Ultimately, the reason we work with $\ell = 2d-2$ is that this is the smallest value of ℓ for which there are at least as many equations as unknowns.

Remark 4.14. As observed in Lemma 3.12 the equations defining $\mathcal{E}_{v,\ell}$ are a subset of the equations defining $\mathcal{E}_{v+e_i,\ell+1}$. In the setup of Lemma 4.4 this difference of equations has size $2d$.

The condition $\Delta_d(F_{x_i=0}) \neq 0$ ensures that these $2d$ equations are linearly independent. Furthermore, if $v_i + 1 \neq 0$, then given $G \in \mathcal{E}_{v,\ell}$ there is a unique $\widetilde{G} \in \mathcal{E}_{v+e_i,\ell+1}$ such that $\widetilde{G}|_{D(v,\ell)} = (v_i+1)\theta_i G$. Thus when $v_i + 1 \neq 0$, we have $\phi_{v,i}(\mathcal{E}_{v,\ell}) = \mathcal{E}_{v+e_1,\ell+1}$.

For the remainder of this section we fix distinct $i, j \in \{0, 1, 2\}$. By composing the map $\phi_{v,i} \colon \mathcal{D}_{v,\ell} \to \mathcal{D}_{v+e_i,\ell+1}$ with the projection $P_{v+e_i,j} \colon \mathcal{D}_{v+e_i,\ell+1} \twoheadrightarrow \mathcal{D}_{v+e_i-e_j,\ell}$ we obtain the map

$$\tau_{v,i,j} \coloneqq P_{v+e_i,j} \circ \phi_{v,i} \colon \mathcal{D}_{v,\ell} \to \mathcal{D}_{v+e_i-e_j,\ell}, \tag{4.15}$$

and the diagram (4.2) as desired. We now check that $\tau_{v,i,j}$ has the desired properties. In particular, if $G \in \mathcal{E}_{v,\ell}$, then $\tau_{v,i,j}(G) \in \mathcal{E}_{v+e_i-e_j,\ell}$, meaning that $\tau_{v,i,j}(G)$ satisfies the equations on a shifted set of monomials.

Corollary 4.16. *We have $\tau_{v,i,j}(\mathcal{E}_{v,\ell}) \subseteq \mathcal{E}_{v+e_i-e_j,\ell}$ and the composition*

$$\mathcal{D}_{v-e_j,\ell-1} \hookrightarrow \mathcal{D}_{v,\ell} \xrightarrow{\tau_{v,i,j}} \mathcal{D}_{v+e_i-e_j,\ell} \xrightarrow{P_{v+e_i,j}} \mathcal{D}_{v-e_j,\ell-1}$$

is scalar multiplication by $(v_i + 1)\theta_i$, and $\tau_{v,i,j}$ is invertible when $v_i + 1 \neq 0$ in R.

The map $\tau_{v,i,j}$ may be represented by a $\binom{2d}{2} \times \binom{2d}{2}$ matrix whose entries are R-linear combinations of $1, v_0, v_1, v_2$ corresponding to linear polynomials in $R[v]$.

Proof. The first part follows by the definition of $\tau_{v,i,j}$ combined with Lemmas 4.4 and 3.12. The last part also follows from Lemma 4.4, where we note that $\#D(v, \ell) = \#D(v+e_i-e_j, \ell) = \#D_\ell = \binom{\ell+n}{n} = \binom{2d}{2}$ for $n = 2$ and $\ell = 2d - 2$. □

Let $\phi^{\mathcal{E}}_{v,i} \colon \mathcal{E}_{v,\ell} \to \mathcal{E}_{v+e_i,\ell+1}$ be the restriction of $\phi_{v,i} \colon \mathcal{D}_{v,\ell} \to \mathcal{D}_{v+e_i,\ell+1}$ and similarly define $\tau^{\mathcal{E}}_{v,i,j}$ and $P^{\mathcal{E}}_{v,i}$. Because we have assumed that F is nondegenerate and $m + 1 \neq 0$ in R, applying Corollary 3.22 with $\ell = 2d - 2$ and $\ell + 1 = 2d - 1$ yields

$$\dim_K \mathcal{W}_\ell = \dim_K \mathcal{E}_{v,\ell} = \dim_K \mathcal{E}_{v+e_i,\ell+1} = \dim_K \mathcal{E}_{v+e_i-e_j,\ell} = d^2. \tag{4.17}$$

Since $\dim_K \mathcal{E}_{v,\ell} = \dim_K \mathcal{W}_\ell$, by (4.17), Corollary 3.20 gives us bijections

$$\pi^{\mathcal{E}}_{v,\ell} \colon \mathcal{E}_{v,\ell} \to \mathcal{W}_{v,\ell}, \qquad \psi^{\mathcal{E}}_{v,\ell} \colon \mathcal{W}_{v,\ell} \to \mathcal{E}_{v,\ell}, \tag{4.18}$$

which are the restrictions of $\pi_{v,\ell}$ and $\psi_{v,\ell}$, respectively. We now consider the map

$$T_{v,i,j} \coloneqq \pi^{\mathcal{E}}_{v+e_i-e_j,\ell} \circ \tau^{\mathcal{E}}_{v,i,j} \circ \psi^{\mathcal{E}}_{v,\ell} \colon \mathcal{W}_{v,\ell} \longrightarrow \mathcal{W}_{v+e_i-e_j,\ell}. \tag{4.19}$$

In other words, the map $T_{v,i,j}$ re-expresses the shifting map $\tau^{\mathcal{E}}_{v,i,j}$ in terms of a basis for $\mathcal{W}_{v,\ell}$. We are interested in applying chains of such maps $T_{v+\bullet,i,j}$, thus for any $s > 0$ we define

$$T^s_{v,i,j} \coloneqq \prod_{s > k \geqslant 0} T_{v+k(e_i-e_j),i,j} = T_{v+(s-1)e_i-(s-1)e_j,i,j} \circ \cdots \circ T_{v+e_i-e_j,i,j} \circ T_{v,i,j}, \tag{4.20}$$

where the product is taken over decreasing values of k starting from $s - 1$; note that the symbol s in $T^s_{v,i,j}$ is a superscript, not an exponent.

Corollary 4.21. *Let s be a positive integer. We have*

$$T^s_{v,i,j} = (m+1)^{s-1} \lambda_\ell^{s-1} \pi^{\mathcal{E}}_{v+se_i-se_j,\ell} \circ \left(\prod_{s > k \geqslant 0} \tau^{\mathcal{E}}_{v+k(e_i-e_j),i,j} \right) \circ \psi^{\mathcal{E}}_{v,\ell}.$$

Furthermore, $\pi^{\mathcal{E}}_{v+se_i-se_j,\ell} \circ \left(\prod_{s>k\geqslant 0} \tau^{\mathcal{E}}_{v+k(e_i-e_j),i,j} \right) \circ \psi^{\mathcal{E}}_{v,\ell}$ may be represented by $d^2 \times d^2$ matrix whose entries are polynomials in $R[v, m] = R[v_0, v_1, v_2, m]$ of degree at most $s + 1$.

Proof. Lemma 3.13 implies $\psi^{\mathcal{E}}_{v+k(e_i-e_j),\ell} \circ \pi^{\mathcal{E}}_{v+k(e_i-e_j),\ell} = (m+1)\lambda_\ell \mathrm{id}_{\mathcal{E}_{v+k(e_i-e_j),\ell}}$ for $0 \leqslant k < s$. Applying this repeatedly yields

$$\begin{aligned} T^s_{v,i,j} &:= \prod_{s>k\geqslant 0} T_{v+k(e_i-e_j),i,j} \\ &= \prod_{s>k\geqslant 0} \pi^{\mathcal{E}}_{v+(k+1)(e_i-e_j),\ell} \circ \tau^{\mathcal{E}}_{v+k(e_i-e_j),i,j} \circ \psi^{\mathcal{E}}_{v+k(e_i-e_j),\ell} \\ &= \pi^{\mathcal{E}}_{v+s(e_i-e_j),\ell} \circ \left(\prod_{s>k>0} \tau^{\mathcal{E}}_{v+k(e_i-e_j),i,j} \circ \psi^{\mathcal{E}}_{v+k(e_i-e_j),\ell} \circ \pi^{\mathcal{E}}_{v+k(e_i-e_j),\ell} \right) \circ \tau^{\mathcal{E}}_{v,i,j} \circ \psi^{\mathcal{E}}_{v,\ell} \\ &= (m+1)^{s-1} \lambda_\ell^{s-1} \pi^{\mathcal{E}}_{v+se_i-se_j,\ell} \circ \left(\prod_{s>k\geqslant 0} \tau^{\mathcal{E}}_{v+k(e_i-e_j),i,j} \right) \circ \psi^{\mathcal{E}}_{v,\ell}. \end{aligned} \tag{4.22}$$

Lemma 3.13, Corollary 3.20, and Corollary 4.16 imply that the RHS can be represented as the product of a scalar, a $d^2 \times \binom{2d}{2}$ matrix, $s-1$ different $\binom{2d}{2} \times \binom{2d}{2}$ matrices, and a $\binom{2d}{2} \times d^2$ matrix, all of whose entries are linear polynomials in $R[v,m]$. The corollary follows. □

Corollary 4.21 combined with Lemma 3.13 yields the following corollary.

Corollary 4.23. *Let $s \in \mathbb{Z}_{\geqslant 0}$ and let $G \in R[x]_{dm}$ satisfy equation (3.7). Then,*

$$T^s_{v,i,j} \circ \pi^{\mathcal{E}}_{v,\ell} \left(G|_{D(v,\ell)} \right) = \theta_i^s \lambda_\ell^s (m+1)^s \left(\prod_{k=1}^{s} (v_i + k) \right) \pi^{\mathcal{E}}_{v+s(e_i-e_j),\ell} \left(G|_{D(v+s(e_i-e_j),\ell)} \right).$$

Before stating the final result of this section, we remind the reader of our running assumptions:

- $i, j \in \{0,1,2\}$ distinct;
- $R = \mathbb{Z}$ or $\mathbb{F}_p$, $n = 2$, $d > 1$, $\ell = 2d-2$, $m > 0$, and $d(m+1) \neq 0$ in R;
- $F \in R[x]_d$ is nondegenerate, meaning $\Delta^*_d(F) \neq 0$ (see (3.5) for the definition of Δ^*_d).

Theorem 4.24. *Let p be a prime that is equal to the characteristic of R when $R = \mathbb{F}_p$ and does not divide $\Delta^*_d(F)d(m+1)$ when $R = \mathbb{Z}$. Let s be a positive integer, and let $G \in R[x]_{dm}$ satisfy equation (3.7). The following hold:*

(a) *If $w \in \mathbb{Z}^{n+1}$ of degree $dm+\ell$ and $v \equiv w \bmod p$ then the matrices representing $T^s_{v,i,j}$ and $T^s_{w,i,j}$ agree modulo p.*

(b) *If $v_i \equiv 0 \bmod p$ and $s = p-1$, then $(m+1)^s \lambda_\ell^s \theta_i^s \prod_{k=1}^{s} (v_i+k) \equiv -1 \bmod p$ and*

$$T^{p-1}_{v,i,j} \circ \pi^{\mathcal{E}}_{v,\ell} \left(G|_{D(v,\ell)} \right) \equiv -\pi^{\mathcal{E}}_{v+(p-1)(e_i-e_j),\ell} \left(G|_{D(v+(p-1)(e_i-e_j),\ell)} \right) \bmod p.$$

When $v_j \equiv -1 \bmod p$ also holds, the matrix $T^{p-1}_{v,i,j}$ is invertible modulo p and its inverse is $T^{p-1}_{v+(p-1)(e_i-e_j),j,i}$.

Proof. For (a) note that $T^s_{v,i,j}$ is representable as a matrix with entries in $R[v]$. For (b) we apply Fermat's little theorem and Wilson's theorem to obtain $\prod_{k=1}^{p-1}(v_i+k) \equiv (p-1)! \equiv -1 \bmod p$, which together with Corollary 4.23 implies the first claim. For the second claim in (b), we

apply $T^{p-1}_{v+(p-1)(e_i-e_j),j,i}$ to both sides of the first claim to obtain

$$\begin{aligned} T^{p-1}_{v+(p-1)(e_i-e_j),j,i} \circ\ T^{p-1}_{v,i,j} \circ \pi^{\mathcal{E}}_{v,\ell}\Big(G|_{D(v,\ell)}\Big) \\ &\equiv -T^{p-1}_{v+(p-1)(e_i-e_j),j,i} \circ \pi^{\mathcal{E}}_{v+(p-1)(e_i-e_j),\ell}\Big(G|_{D(v+(p-1)(e_i-e_j),\ell)}\Big) \bmod p \\ &\equiv \pi^{\mathcal{E}}_{v,\ell}\left(G|_{D(v,\ell)}\right) \bmod p, \end{aligned}$$

where the last equivalence follows from the first claim in (b), since $v_j \equiv -1 \bmod p$ implies $(v+(p-1)(e_i-e_j))_j \equiv 0 \bmod p$, allowing us to apply the first claim to $T^{p-1}_{v+(p-1)(e_i-e_j),j,i}$. □

5. Computing Cartier–Manin matrices of a smooth plane quartic

Let $X\colon f(x_0,x_1,x_2)=0$ be a smooth plane quartic defined by a nondegenerate homogeneous quartic $f \in R[x_0,x_1,x_2]_4$. In this section we give algorithms to compute the Cartier–Manin matrix A_p of X when $R=\mathbb{F}_p$, or the Cartier–Manin matrices A_p of the reductions of X modulo primes $p \leqslant N$ of good reduction up to a given bound N when $R=\mathbb{Z}$.

We first consider the case $R=\mathbb{F}_p$, where p is an odd prime, noting that for $p=2$ the Cartier–Manin matrix can be extracted directly from the coefficients of $f=f^{p-1}$ via (2.6). We will apply the infrastructure developed in §4 with $F=f$ and $m=p-2$. In particular, we work with $\ell=6=2d-2$ and $dm+\ell=4(p-2)+6=4p-2$ throughout.

Let us first sketch our algorithm by working backwards from our goal. The coefficients of f^{p-1} that appear in the ith column of the matrix A_p in (2.6) lie in $f^{p-1}|_{D(v^{(i)},2)}$ for

$$v^{(1)} \coloneqq (p-1,p,2p-1), \quad v^{(2)} \coloneqq (2p,p-1,p-1), \quad v^{(3)} \coloneqq (p-1,2p,p-1); \tag{5.1}$$

note that the $v^{(i)}$ are not symmetric because the indices in the columns of (2.6) are not. Now $D_{v,2}=B_{v,2}$, since $2<4=d$, so $\pi_{v,6}$ has codomain $\mathcal{W}_{v,6}$ and it suffices to compute

$$\pi_{v,6}\big(f^{p-2}|_{D(v,6)}\big) = f^{p-2}|_{B(v,6)} + f^{p-1}|_{B(v,2)} \in \mathcal{W}_{v,6} \tag{5.2}$$

for $v=v^{(1)},v^{(2)},v^{(3)}$. We now define

$$w^{(1)} \coloneqq (0,2p-1,2p-1), \quad w^{(2)} \coloneqq (3p-1,0,p-1), \quad w^{(3)} \coloneqq (0,3p-1,p-1), \tag{5.3}$$

with $w^{(1)}=v^{(1)}+(p-1)(e_1-e_0)$, $w^{(2)}=v^{(2)}+(p-1)(e_0-e_1)$, and $w^{(3)}=v^{(3)}+(p-1)(e_1-e_0)$. Let $C_p \in \mathbb{F}_p^{16\times 16}$ denote the matrix representing the linear operator

$$T^{p-1}_{w^{(1)},0,1}\colon \mathcal{W}_{w^{(1)},6} \to \mathcal{W}_{v^{(1)},6}, \tag{5.4}$$

determined by the nondegenerate polynomial $f \in \mathbb{F}_p[x_0,x_1,x_2]_4$. By Theorem 4.24 (a), the matrix C_p also represents

$$T^{p-1}_{w^{(3)},0,1}\colon \mathcal{W}_{w^{(3)},6} \to \mathcal{W}_{v^{(3)},6}, \tag{5.5}$$

since $v^{(1)} \equiv v^{(3)} \bmod p$ and $w^{(1)} \equiv w^{(3)} \bmod p$, and by Theorem 4.24 (b), C_p^{-1} represents

$$\left(T^{p-1}_{v^{(2)},0,1}\right)^{-1} \equiv T^{p-1}_{w^{(2)},1,0}\colon \mathcal{W}_{w^{(2)},6} \to \mathcal{W}_{v^{(2)},6} \tag{5.6}$$

since $v^{(2)}_0 \equiv 0 \bmod p$ and $v^{(2)}_1 \equiv -1 \bmod p$ and $w^{(2)}=v^{(2)}+(p-1)(e_0-e_1)$. We can thus use the matrix C_p and its inverse to traverse the three paths from the intermediate points w depicted as blue dots on the exterior of triangle in Figure 2 to the target interior points v.

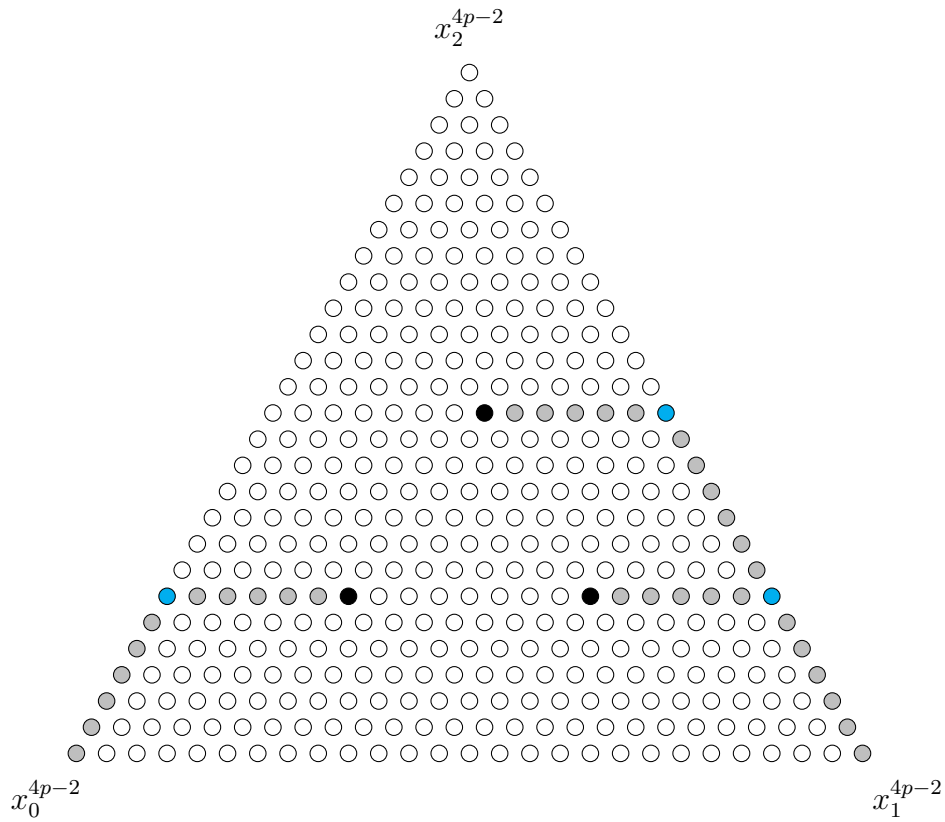

FIGURE 2. Illustration for $p = 7$. The target points v in the interior are shown in black with $v^{(1)}$ at the top center, $v^{(2)}$ at the lower left, and $v^{(3)}$ at the lower right. The intermediate points w are in blue, and the paths used to reach the target points v are shown in gray.

To obtain the coefficients of $f^{p-2}|_{D(w,6)}$ for $w = w^{(1)}, w^{(2)}, w^{(3)}$ we could apply a variation of the method of §4 for $n = 1$ (each w has a zero entry we can ignore), but we prefer to use a simpler approach that we illustrate for $w = w^{(3)}$. Let $h(t) \coloneqq f(0, 1, t)$. Then

$$h^{p-2}(t) \equiv h(t^p)h^{-2}(t). \tag{5.7}$$

If we put $g(t) \coloneqq h(t)^2 = \sum_{i=0}^{8} a_i t^i$ and let

$$a_0/g(t) = \sum_{i \geqslant 0} c_i t^i \in \mathbb{F}_p[[t]], \tag{5.8}$$

then we can compute $(c_s, c_{s-1}, \cdots, c_{s-7})$ as the first column of Q_g^s, where

$$a_0 Q_g \coloneqq \begin{bmatrix} -a_1 & -a_2 & -a_3 & \cdots & -a_8 \\ a_0 & 0 & 0 & \cdots & 0 \\ 0 & a_0 & 0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & \cdots & 0 & a_0 & 0 \end{bmatrix}, \tag{5.9}$$

Computing Q_g^s with $s = p-1$ allows us to derive the $\binom{6+1}{1} = 7$ coefficients of $f^{p-2}|_{D(w,6)}$ we need using $c_s, \dots, c_{s-6}$; the other $\binom{6+2}{2} - \binom{6+1}{1} = 21$ coefficients correspond to monomials in $\mathbb{F}_p[x^{\pm}]$ that contain a negative exponent and are necessarily zero because f^{p-2} is a polynomial. In terms of Figure 2, the computation we have just described corresponds to walking $p-1$ steps along the gray path from the lower right corner of the triangle to the first blue dot on the right edge (the 21 zero coefficients correspond to monomials outside the triangle).

The cases $w = w^{(1)}, w^{(2)}$ are treated similarly using suitable $g(t)$ and s.

Algorithm 5.10. Given a nondegenerate $f \in \mathbb{F}_p[x_0, x_1, x_2]_4$ and the corresponding matrix $C_p \in \mathbb{F}_p^{16\times 16}$, compute the Cartier–Manin matrix of $X\colon f(x_0, x_1, x_2) = 0$ as follows:

(1) Compute $f^{p-2}|_{D(w,6)}$ for $w = w^{(1)}, w^{(2)}, w^{(3)}$ (the blue dots in Figure 2) using suitably chosen $g \in \mathbb{F}_p[t]$ and $Q_g^s \in \mathbb{F}_p^{8\times 8}$ as described above:

(a) Compute the edge coefficients of $f^{p-2}|_{D(w^{(1)},6)}$ using $g(t) \coloneqq f(0,1,t)^2$:

$$\left(f^{p-2}_{w^{(1)}-je_2-(6-j)e_1}\right)_{0\leqslant j\leqslant 7} = \left(f_{(0,3,1)}f_{(0,4,0)}{}^{-2}Q_g^{p-1} + f_{(0,4,0)}{}^{-1}Q_g^{2p-1}\right)\cdot(1,0,\dots,0)^T.$$

(b) Compute the edge coefficients of $f^{p-2}|_{D(w^{(2)},6)}$ using $g(t) \coloneqq f(1,0,t)^2$:

$$\left(f^{p-2}_{w^{(2)}-je_2-(6-j)e_0)}\right)_{0\leqslant j\leqslant 7} = f_{(4,0,0)}{}^{-1}Q_g^{p-1}\cdot(1,0,\dots,0)^T.$$

(c) Compute the edge coefficients of $f^{p-2}|_{D(w^{(3)},6)}$ using $g(t) \coloneqq f(0,1,t)^2$:

$$\left(f^{p-2}_{w^{(3)}-je_2-(6-j)e_1}\right)_{0\leqslant j\leqslant 7} = f_{(0,4,0)}{}^{-1}Q_g^{p-1}\cdot(1,0,\dots,0)^T.$$

(2) Compute $\pi_{v,6}(f^{p-2}|_{D(v,6)})$ for $v = v^{(1)}, v^{(2)}, v^{(3)}$ (the black dots in Figure 2) using Theorem 4.24 and Equation (5.2) as follows:

(a) Compute the first column of A_p using $v^{(1)} = (p-1, p, 2p-1)$:

$$\left(f^{p-1}_{(p-1,p,2p-3)}, \boldsymbol{f^{p-1}_{(p-1,p-1,2p-2)}}, \boldsymbol{f^{p-1}_{(p-1,p-2,2p-1)}}, f^{p-1}_{(p-2,p,2p-2)}, \boldsymbol{f^{p-1}_{(p-2,p-1,2p-1)}}, f^{p-1}_{(p-3,p,2p-2)}\right)$$
$$= \left(\pi_{v^{(1)},6}\left(f^{p-2}|_{D(v^{(1)},6)}\right)\right)|_{B(v^{(1)},2)} = -C_p\circ\pi_{w^{(1)},6}\left(f^{p-2}|_{D(w^{(1)},6)}\right).$$

(b) Compute the second column of A_p using $v^{(2)} = (2p, p-1, p-1)$:

$$\left(f^{p-1}_{(2p,p-1,p-3)}, f^{p-1}_{(2p,p-2,p-2)}, f^{p-1}_{(2p,p-3,p-1)}, \boldsymbol{f^{p-1}_{(2p-1,p-1,p-2)}}, \boldsymbol{f^{p-1}_{(2p-1,p-2,p-1)}}, \boldsymbol{f^{p-1}_{(2p-2,p-1,p-1)}}\right)$$
$$= \left(\pi_{v^{(2)},6}\left(f^{p-2}|_{D(v^{(2)},6)}\right)\right)|_{B(v^{(2)},2)} = -C_p{}^{-1}\circ\pi_{w^{(2)},6}\left(f^{p-2}|_{D(w^{(2)},6)}\right).$$

(c) Compute the third column of A_p using $v^{(3)} = (p-1, 2p, p-1)$:

$$\left(f^{p-1}_{(p-1,2p,p-3)}, \boldsymbol{f^{p-1}_{(p-1,2p-1,p-2)}}, \boldsymbol{f^{p-1}_{(p-1,2p-2,p-1)}}, f^{p-1}_{(p-2,2p,p-2)}, \boldsymbol{f^{p-1}_{(p-2,2p-1,p-1)}}, f^{p-1}_{(p-3,2p,p-1)}\right)$$
$$= \left(\pi_{v^{(3)},6}\left(f^{p-2}|_{D(v^{(3)},6)}\right)\right)|_{B(v^{(3)},2)} = -C_p\circ\pi_{w^{(3)},6}\left(f^{p-2}|_{D(w^{(3)},6)}\right).$$

(3) Output the matrix $A_p \in \mathbb{F}_p^{3\times 3}$ defined in (2.6) using the coefficients of f^{p-1} that are shown in bold above.

Remark 5.11. The matrix Q_g^{p-1} in step (1c) is the same as in step (1a) and need not be recomputed. The matrices that represent $\pi_{w,6}$ for $w = w^{(1)}, w^{(2)}, w^{(3)}$ in step (2) are all the same, since $\pi_{w,6}$ does not depend on w, by Lemma 3.13. Indeed, if $\iota(t) \in \{1, \dots \#D_\ell\}$ is the index of $t \in D_\ell$ in its lexicographic ordering, the matrix $W \in R^{16\times 28}$ with nonzero entries $W_{\iota(u),\iota(t+u)} \coloneqq F_t$ for $u \in D_2$ and $t \in D_4$ and $W_{6+j,18+j} \coloneqq 1$ for $1 \leqslant j \leqslant 10$ represents $\pi_{w,6}$.

Remark 5.12. If we instead use the "uncompressed" matrix $U_p \in \mathbb{F}_p^{28\times 28}$ representing the linear operator $\prod_{p-1>k\geqslant 0} \tau^{\mathcal{E}}_{w^{(1)}+k(e_0-e_1),0,1}$, which by (4.21) satisfies

$$U_p = -\lambda_6^{-1} \psi^{\mathcal{E}}_{v^{(1)},6} \circ C_p \circ \pi^{\mathcal{E}}_{w^{(1)},6},$$

we can consider an "uncompressed" version of Algorithm 5.10. We replace $C_p \circ \pi_{w^{(1)},6}$ and $C_p \circ \pi_{w^{(3)},6}$ with $\pi_{v^{(1)}} \circ U_p$ and $\pi_{v^{(3)}} \circ U_p$, respectively, to obtain the desired vectors in (2a) and (2c), and for (2b) we replace C_p^{-1} with $-\lambda_6(\pi^{\mathcal{E}}_{v^{(1)},6} \circ U_p \circ \psi^{\mathcal{E}}_{w^{(1)},6})^{-1}$.

Lemma 5.13. *Algorithm 5.10 runs in $O(\log^2 p \log\log p)$ time using $O(\log p)$ space.*

Proof. The algorithm uses $O(\log p)$ ring operations for the matrix exponentiations and $O(1)$ field inversions in step (1), and $O(1)$ field operations in step (2). Each ring operation in $\mathbb{F}_p$ can be achieved using $O(1)$ ring operations in $\mathbb{Z}$ on integers with $O(\log p)$ bits (using Newton iteration to perform fast Euclidean division, see [GG13, Thm. 9.8]), which yields a bit complexity of $O(\mathsf{M}(\log p)) = O(\log p \log\log p)$ per ring operation via [HvdH21]. We can perform field inversions in $O(\mathsf{M}(\log p)\log\log p) = O(\log p(\log\log p)^2)$ time using a fast GCD algorithm [GG13, Cor. 11.13], which is dominated by the cost of $O(\log p)$ ring operations; the time bound follows and the space bound is immediate. □

We now give our algorithms to compute the Cartier–Manin matrix of a smooth plane quartic. Let us define the matrix

$$M(t) \coloneqq T_{w^{(1)}+t(e_0-e_1),0,1} \in R[t]^{16\times 16}, \tag{5.14}$$

whose entries are polynomials in t of degree at most 2, by Corollary 4.21. From (4.19), we see that $M(t)$ can be computed as the product of matrices in $R[t]^{16\times 28}, R[t]^{28\times 28}, R[t]^{28\times 16}$ representing the maps $\pi^{\mathcal{E}}_{v(t)+e_0-e_1,6}$, $\tau^{\mathcal{E}}_{v(t),0,1}$, $\psi^{\mathcal{E}}_{v(t),6}$, respectively, where $v(t) = w^{(1)} + t(e_0 - e_1)$. The matrices representing $\pi^{\mathcal{E}}_{v(t)+e_0-e_1,6}$ and $\psi^{\mathcal{E}}_{v(t),6}$ are computed as in the proof of Lemma 3.13: the matrix representing $\pi^{\mathcal{E}}_{v(t)+e_0-e_1,6}$ is independent of $v(t)$, its entries are coefficients of f or elements of $\{0,1\}$, while the entries in $\psi^{\mathcal{E}}_{v(t),6}$ are determined by (3.18). The matrix representing $\tau^{\mathcal{E}}_{v(t),0,1} = P^{\mathcal{E}}_{v(t)+e_0,j} \circ \phi^{\mathcal{E}}_{v(t),i}$ is computed by composing the matrix in $\{0,1\}^{28\times 36}$ representing the projection $P^{\mathcal{E}}_{v(t)+e_0,1}$ with the matrix in $R[t]^{36\times 28}$ representing $\phi^{\mathcal{E}}_{v,i}$ whose entries are given by (4.12). From equation (4.20) we then have

$$C_p \coloneqq T^{p-1}_{w^{(1)},0,1} = \prod_{p-1>j\geqslant 0} M(j) \bmod p \in \mathbb{F}_p^{16\times 16}. \tag{5.15}$$

Algorithm 5.16. Given a nondegenerate $f \in \mathbb{F}_p[x_0, x_1, x_2]_4$, compute the Cartier–Manin matrix A_p of the smooth plane quartic $X\colon f(x_0, x_1, x_2) = 0$ as follows:

(1) Compute the matrix $M(t) \in \mathbb{F}_p[t]^{16\times 16}$ corresponding to f as described above.

(2) Compute the matrix $C_p = T^{p-1}_{w_1,0,1} = \prod_{p-1>j\geqslant 0} M(j) \in \mathbb{F}_p^{16\times 16}$.

(3) Use Algorithm 5.10 with inputs C_p and f to compute the Cartier–Manin matrix A_p.

Remark 5.17. We may also consider an uncompressed version of Algorithm 5.10 that uses $M(t) \coloneqq \tau^{\mathcal{E}}_{w^1+t(e_0-e_1),0,1} \in R[t]^{28\times 28}$ to compute the matrix U_p defined in Remark 5.10 rather than using the matrices $M(t)$ defined in (5.14) to compute C_p. Note that in the former case the entries of $M(t)$ have degree at most 1 rather than 2.

Theorem 5.18. *Algorithm 5.16 can be implemented to use $O(p\log p\log\log p)$ time and $O(\log p)$ space, and also to use $O(p^{1/2}\log^2 p)$ time and $O(p^{1/2}\log p)$ space.*

Proof. The first complexity bound is achieved by iteratively instantiating the entries of $M(t)$ at $t = k$ and accumulating the matrix product in C_p. This involves $O(p)$ ring operations in $\mathbb{F}_p$, which takes $O(p\log p\log\log p)$ time using $O(\log p)$ space. The second complexity bounds is achieved by using the interpolation/evaluation algorithm of Bostan–Gaudry–Schost [BGS07] to compute $\prod_{p-1>j\geqslant 0} M(j)$, which uses $\mathsf{M}(p^{1/2}\log p) = O(p^{1/2}\log^2 p)$ time and $O(p^{1/2}\log p)$ space. The cost of invoking Algorithm 5.10 in step (2) is negligible in both cases. □

Remark 5.19. In our $O(p\log p\log\log p)$ implementation, rather than computing C_p as the product of $p-1$ matrices $M(j)$, we instead iteratively multiply the vectors $\pi_{w^{(i)},6}(f^{p-2}|_{D(w^{(i)},6)})$ that appear in steps (2a) and (2c) of Algorithm 5.10 by each matrices $M(j)$ as it is computed. We then repeat this process using the curve defined by $f(x_1,x_0,x_2)$ to obtain the vector computed in step (2b); note that in steps (1c) and (2c) of Algorithm 5.10 are identical to steps (1b) and (2b) except the roles of x_0 and x_1 are reversed. This effectively replaces each matrix multiplication with 3 matrix-vector multiplications and is practically faster in the range of p we consider. The matrices $M(j)$ for $j = 0,\dots,p-1$ can be efficiently enumerated using finite differences (the entries of $M(t)$ are polynomials of degree at most 2).

We now turn to the case $R = \mathbb{Z}$, where our strategy is to use an average polynomial-time approach to simultaneously compute the matrices C_p at suitable primes $p \leqslant N$ using a single matrix $M(t) \in \mathbb{Z}[t]^{16\times 16}$. A nondegenerate polynomial $f \in \mathbb{Z}[x_0,x_1,x_2]_4$ will have nondegenerate reduction modulo all primes p that do not divide $\Delta_4^*(f)$, but in order to obtain a valid matrix C_p to use as input for Algorithm 5.10 computed via (5.15) with $M(t) \in \mathbb{Z}[t]^{16\times 16}$ we also need to ensure that the scalar $(m+1)\lambda_6$ arising in Lemma 3.13 and the degree $d = 4$ are both nonzero modulo p.

Now $m+1 = p-1$ is never divisible by p, so it suffices to restrict our attention to odd primes that do not divide λ_6. We thus define $D \coloneqq 2\lambda_6\Delta_4^*(f)$ and treat all primes $p \leqslant N$ that do not divide D using an average polynomial-time approach and handle good primes $p \mid D$ as special cases via Remark 5.20 below. The primes $p \mid D$ are bounded by a constant that does not depend on N, thus the time spent handling the good $p \mid D$ has no impact on the complexity of our algorithm as a function of N (and it is completely negligible in practice).

Remark 5.20. For primes $p \mid D$ where f has good reduction we can compute the Cartier–Manin matrix directly from its definition, but we can more efficiently treat $p \nmid \Delta_4^*(f)$ by simply applying Algorithm 5.16 to the nondegenerate reduction of f modulo p. In our implementation we do the same for good primes $p \mid \Delta_4^*(f)$ greater than 3 by applying a random linear transformation to the reduction of f modulo p until we obtain a nondegenerate polynomial $\widetilde{f} \in \mathbb{F}_p[x_0,x_1,x_2]$ that defines an isomorphic curve. In practice this appears to always work for $p > 3$ but we make no attempt to prove this here. Note that we have assumed $f(x_0,x_1,x_2) = 0$ is a model for X that is smooth a p, but if not, replace f modulo p with the reduction of a model for X that is smooth at p.

Before describing our average polynomial-time algorithms to compute A_p for $p \leqslant N$ coprime to D, we briefly recall some background material on remainder trees and forests. Given a sequence of integer matrices $M_0, \dots, M_{N-1}$ and a sequence of coprime integers $m_0, \dots, m_{N-1}$ we wish to compute the following sequence of reduced partial products for $0 \leqslant k < N$:

$$P_k \coloneqq M_0 \cdots M_k \bmod m_k.$$

Let $M_{-1} \coloneqq M_N \coloneqq m_N \coloneqq 1$, and for $0 \leqslant k < N/2$ let $M'_k \coloneqq M_{2k-1}M_{2k}$ and $m'_k \coloneqq m_{2k}m_{2k+1}$. If we now recursively compute $P'_k \coloneqq M'_0 \cdots M'_k \bmod m'_k = M_0 \cdots M_{2k} \bmod m_{2k}m_{2k+1}$ for $0 \leqslant k < N/2$, we can then compute

$$P_{2k} = P'_k \bmod m_{2k} \qquad \text{and} \qquad P_{2k+1} = P'_k M_{2k+1} \bmod m_{2k+1}.$$

Unwinding this recursion yields the RemainderTree algorithm described in [HS14].

The RemainderForest algorithm in [HS16] reduces the time and (especially) the space needed by splitting the remainder tree into 2^κ-subtrees, for a suitable choice of κ. In [HS14, HS16, Sut20] the RemainderForest algorithm is used to compute the sequence of vectors $V_k \coloneqq V_0 M_0 \cdots M_k \bmod m_k$ using vector-matrix multiplications to carry results from one subtree to the next, but it can also be used to compute $P_k = IM_0 \cdots M_k \bmod m_k$ using the same approach. Below we record a special case of [HS16, Theorem 3.3], in which $\|M_k\|$ denotes the logarithm of the largest absolute value appearing in the nonzero matrix M_k.

Theorem 5.21. *Fix a constant $c > 0$. Let N be a positive integer, let $m_0, \dots, m_{N-1}$ be positive coprime integers with $\log \prod_{k=0}^{n} m_k \leqslant cn$ for $2 \leqslant n < N$, let $M_0, \dots, M_{N-1} \in \mathbb{Z}^{r\times r}$ be nonzero integer matrices with $r < c \log N$ and $\|M_i\| \leqslant c \log N$. We can compute the matrices*

$$P_k \coloneqq \prod_{i=0}^{k} M_i \bmod m_k$$

for $0 \leqslant k < N$ in $O(r^2 N \log^3 N)$ time using $O(r^2 N)$ space.

Proof. We apply [HS16, Thm. 3.3] with $\kappa \coloneqq \lfloor 2 \log_2 \log_2 N \rfloor$, $B = cN$, $B' = 1$, $H = c \log N$. We use $\mathsf{M}(n) = O(n \log n)$ from [HvdH21] and note that replacing $\mathsf{M}(n)$ with $n \log n$ in the statement of [HvdH18, Lem. 4] allows us to omit the last step of the proof where the hypothesis that $\mathsf{M}(n)/(n \log n)$ is increasing is used and remove that hypothesis.

Provided $\log r = O(\log B)$, the complexity of multiplying $r \times r$ matrices with B-bit entries is $O(r^2 B \log B + r^\omega B \log\log B)$, where $\omega < 3$ is the exponent of matrix multiplication. We have $r = O(\log B)$, so this is $O(r^2 B \log B) = O(r^2 N \log N)$, which we may substitute for [HS16, Lem. 3.1] in the proof of [HS16, Thm 3.3]. The cost of replacing vector-matrix multiplications with matrix multiplications as we transition from one subtree to the next is asymptotically negligible: we may reduce modulo $m \coloneqq \prod_{k=0}^{N-1} m_k$ throughout and perform $O(2^\kappa) = O(\log^2 N)$ matrix multiplications with $O(N)$-bit entries, each involving $O(r^2 N \log N)$ bit operations. □

Algorithm 5.22. Given $f \in \mathbb{Z}[x_0, x_1, x_2]_4$ with $\Delta_4^*(f) \neq 0$ and a positive integer N, compute the Cartier–Manin matrices A_p of the reductions of the smooth plane quartic $X\colon f(x_0, x_1, x_2) = 0$ modulo primes $p \leqslant N$ of good reduction for X as follows:

(1) Use the RemainderForest algorithm to compute $C_p = \prod_{p-1>j\geqslant 0} M(j) \bmod p$ for primes $p \leqslant N$ with $p \nmid D$ using the matrices $M_i \coloneqq M(-2-i) \in \mathbb{Z}^{16\times 16}$ and moduli $m_i \coloneqq i+2$ when $i+2$ is a prime $p \nmid D$ and with $m_i \coloneqq 1$ otherwise, for $0 \leqslant i < N-1$. The matrices M_i and moduli m_i should be dynamically computed as needed.

(2) For each C_p computed in (1) apply Algorithm 5.10 with input $f \bmod p$ and C_p to compute A_p. This step should be interleaved with step (1), computing the relevant A_p in batches as the RemainderForest algorithm completes each subtree.

(3) For $p \leqslant N$ of good reduction dividing D compute A_p via Remark 5.20.

Note that for primes $p \leqslant N$ that do not divide D we have

$$\begin{aligned} P_{p-2} &= \prod_{i=0}^{p-2} M_i \bmod m_{p-2} = \prod_{i=0}^{p-2} M(-2-i) \bmod p \\ &\equiv \prod_{i=0}^{p-2} M(p-2-i) \equiv \prod_{p-1>j\geqslant 0} M(j) \equiv C_p \bmod p, \end{aligned} \tag{5.23}$$

thus step (1) of Algorithm 5.22 computes exactly the matrices C_p that are needed in step (2).

Remark 5.24. Lemma 3.13 and Corollary 4.21 imply that each integer matrix product $M_i M_{i+1}$ is divisible by λ_6. In our implementation of Algorithm 5.22 we precompute λ_6 and remove it from each matrix product computed during the RemainderForest computation in step (1). This changes the output $P_{p-2} \bmod p$ by a factor of λ_6^{p-2}, and we divide once more by λ_6 to obtain the desired matrix C_p, since $\lambda_6^{p-1} \equiv 1 \bmod p$ (note that $\lambda_6 \,|\, D$ so $p \nmid \lambda_6$). This does not change the complexity of the algorithm, but it reduces the sizes of the matrix coefficients in every layer of the product tree above the leaves by roughly a factor of 2, which yields a significant constant factor speedup (more than a factor of 2 in our tests).

Remark 5.25. As in Remark 5.17, we may also consider an uncompressed version of Algorithm 5.22 that instead computes 28×28 matrices $U_p \bmod p$ and uses Remark 5.12 to compute the Cartier–Manin matrices A_p. In this uncompressed version we are not able to apply the optimization noted in Remark 5.24.

Remark 5.26. Algorithms 5.16 and 5.22 can be modified to more efficiently handle smooth plane quartics of the form $f(x_0, x_1, x_2) = x_0^4 + h(x_1, x_2)x_0^2 + g(x_1, x_2)$. In this case $f_v^{p-1} = 0$ whenever v_0 is odd, and for $p > 2$ this implies that the Cartier–Manin matrix $A_p \in \mathbb{F}_p^{3\times 3}$ has at most five nonzero entries: the four corners and the center. The center corresponds to the 1×1 Cartier–Manin matrix of the genus 1 curve $x_0^2 = h(x_1, x_2)^2 - 4g(x_1, x_2)$ which can be computed via [HS16] using 4×4 matrices. Restricting the domain and codomain of

$$\tau^{\mathcal{E}}_{w^{(1)}+(2t+1)(e_0-e_1),0,1} \circ \tau^{\mathcal{E}}_{w^{(1)}+2t(e_0-e_1),0,1}$$

to the subspaces spanned by monomials with even degree in x_0 yields a matrix $M \in R[t]^{16\times 16}$. One finds that M can be compressed via a coordinate projection to a 10×10 matrix M', and we compute $W_p := \prod_{\frac{p-3}{2}\geqslant k\geqslant 0} M(k) \bmod p$ as the product of $M(\frac{p-3}{2})$ and the zero extension of $\prod_{\frac{p-3}{2}>k\geqslant 0} M'(t) \bmod p$. The matrix W_p can then be zero extended to $U_p \in \mathbb{F}_p^{28\times 28}$ and used to compute the four corner entries of A_p via Remark 5.17.

Theorem 5.27. *Algorithm 5.22 runs in $O(N \log^3 N)$ time using $O(N)$ space.*

Proof. Theorem 5.21 implies that the complexity of step (1) is within the desired bounds. Step (2) calls Algorithm 5.10 $O(N/\log N)$ times, which takes $O(N \log N \log\log N)$ time using $O(\log N)$ space. The complexity of step (3) is asymptotically negligible, since D is fixed as a function of N, and the theorem follows. □

To help assess the benefits of our new recurrences, we also implemented an algorithm that uses the recurrences derived in [Har15] to compute the Cartier–Manin matrix A_p of a smooth plane quartic $X : f(x_0, x_1, x_2) = 0$ (or its reduction modulo p when $R = \mathbb{Z}$). If one applies [Har15, Thm. 4.1] with $n = 2$, $d = 4$, $s = 1$, $h = (d-1)(n+1)+1 = 10$, $k_0 = p-1$, and $w = v + z$ with $z = (0, 0, 6) \in D_{h-d}$, one obtains a matrix $Q \in R[k, l]^{66\times 66}$ that can be used to compute $f^{p-1}|_{D(pv+z,10)}$ for any $v \in D_4$ via

$$f^{p-1}|_{D(pv+z,10)} = \frac{1}{d^{p-1}(p-1)!} Q(p-1, p-2)Q(p-1, p-3)\cdots Q(p-1, 0)g^{p-1}|_{D(pv+z,10)}, \quad (5.28)$$

where $g(x_0, x_1, x_2) = x_0^4 + x_1^4 + x_2^4$. The algorithm in [Har15] uses the matrix Q to compute a matrix M_s which is then used to compute the matrix $A_{F^s}^{ar}$ that appears in the trace formula [Har15, Thm. 3.1], but the Cartier–Manin matrix A_p can be computed directly from (5.28), and it suffices to compute the product $M(p-2)M(p-3)\cdots M(0) \bmod p$, where $M(j) \coloneqq Q(-1, j)$; the algorithm in [Har15] works modulo p^2 when $s = 1$, but that is not necessary here. This product does not depend on $v \in D_4$, so it suffices to compute a single matrix product and then apply (5.28) using $v = (1, 1, 2), (2, 1, 1), (1, 2, 1)$; this yields three vectors in $\mathbb{F}_p^{66}$, each of which contains three entries that correspond to a column of A_p.

Having reduced the problem to computing $\prod_{p-1>j\geqslant 0} M(j) \bmod p$ we immediately obtain algorithms to compute A_p with the complexities given in Theorem 5.18 for $R = \mathbb{F}_p$, and for $R = \mathbb{Z}$ we obtain an average polynomial-time algorithm with the complexities given in Theorem 5.27 using a remainder forest. The difference in the size of the matrices (66 versus 28 or 16) only impacts the constant factors, which we consider in the next section.

Remark 5.29. There is an additional optimization that we exploit in our implementation of the average polynomial-time algorithm based on [Har15, Thm. 4.1]. In the remainder forest algorithm, rather than computing the 66×66 matrix $P_k = M_0 \cdots M_k \bmod m_k$ we instead compute the 3×66 matrix $P_k = V_0 M_0 \cdots M_k \bmod m_k$, where V_0 is a 3×66 matrix with entries in $\{0, 1\}$ and zeros in all but one entry of each row. This optimization is possible because we only need 3 rows of the matrix product to compute A_p. This optimization is not applicable in the context of Algorithm 5.22 because we need to invert the reduced matrix products in order to compute the middle column of A_p via Algorithm 5.10.

A demonstration version of the $\widetilde{O}(p)$ and average polynomial-time versions of all three approaches (compressed, uncompressed, and the algorithm based on [Har15, Thm. 4.1]) written in the SageMath computer algebra system [Sage] is available at [CHS22]. The optimized C implementation whose practical performance is analyzed in the next section will be part of the next release of the open source `smalljac` software library [KS08].

6. Performance comparisons

In this section we compare the practical performance of our new algorithms to each other, and to existing implementations, both for computing the Cartier–Manin matrix of a smooth plane quartic over $\mathbb{F}_p$ (see Table 1), and for computing the Cartier–Manin matrices of the reductions of a smooth plane quartic over $\mathbb{Q}$ at good primes $p \leqslant N$ for some bound N. Table 2 compares the new average polynomial-time algorithms to each other and Table 4 compares them to average polynomial-time algorithms for other types of genus 3 curves.

We first consider $\widetilde{O}(p)$ and $\widetilde{O}(p^{1/2})$ implementations of the compressed and uncompressed versions of Algorithm 5.16 (denoted Algorithm 5.16c and Algorithm 5.16u below) as well as $\widetilde{O}(p)$ and $\widetilde{O}(p^{1/2})$ implementations of the approach based on [Har15, Thm. 4.1] described at the end of the previous section (denoted [Har15] (optimized) below). We compared the performance of these six algorithms to each other, and to the following existing algorithms:

- In [Cos15] Costa gives an $\widetilde{O}(p)$-time p-adic algorithm for computing the matrix of Frobenius to a specified p-adic precision, which can be used to compute the Cartier–Manin matrix of a smooth plane quartic. This algorithm is available at [Cos15a].
- The `smalljac` software library [KS08] includes a naïve point-counting algorithm for plane projective curves $X\colon f(x_0,x_1,x_2)=0$ that computes

$$\#X(\mathbb{F}_p) = 0^{f(1,0,0)} + \#\{t\in\mathbb{F}_p : f(t,0,1)=0\} + \sum_{a\in\mathbb{F}_p} \#\{t\in\mathbb{F}_p : f(t,1,a)=0\} \tag{6.1}$$

  via the identity $\#\{t\in\mathbb{F}_p : g(t)=0\} = \deg\gcd(g(t),t^p-t)$ (valid for $g\neq 0$), in $O(p\log^2 p\log\log p)$ time using $O(\log p)$ space.
- For smooth plane curves the `RationalPoints` function in Magma [Magma] uses an $O(p\log^2 p\log\log p)$-time algorithm to enumerate rational points over $\mathbb{F}_p$.

The last two algorithms only compute $\#X(\mathbb{F}_p)$, they do not compute the Cartier–Manin matrix A_p, which provides additional information about X, including the reduction of its zeta function modulo p and the p-rank of its Jacobian. Magma includes an implementation of Tuitman's algorithm [Tui16] that computes the entire zeta function in $\widetilde{O}(p)$ time, but the constant factors make it more than 100 times slower than the three $\widetilde{O}(p)$ algorithms listed above in the ranges we tested, so we chose not to include it in our comparison.

We ran each of these 9 algorithm on smooth plane quartics defined by dense polynomials $f\in\mathbb{F}_p[x_0,x_1,x_2]_4$, taking p to be the first prime larger than 2^n for $n=10,11,\dots,30$. The running times for each algorithm can be found in Table 1, in which the complexity bounds in the column headings ignore $O(\log\log p)$ factors.

Each of the three $\widetilde{O}(p^{1/2})$ algorithms is substantially faster than the existing approaches, as one would expect given the asymptotic advantage. For $p\approx 2^{30}$ Algorithm 5.16c appears to be faster than Algorithm 5.16u by factor of about 3, which in turn appears to be faster than [Har15] (optimized) by a factor of almost 8. The factor of $3\approx(28/16)^2$ is as expected, while the factor of $8>5.6\approx(66/28)^2$ is larger than one might expect; this is likely due to the fact that p is not large enough for the $O(r^\omega p^{1/2}\log p\log\log p)$ term in the complexity bound from [BGS07] to become completely negligible. All three implementations use the `smalljac` library [KS08], which includes an implementation of the algorithm in [BGS07] built on the `zn_poly` library [Har10], which is used for fast cache-friendly multiplication in $\mathbb{F}_p[x]$.

The relative performance of the $\widetilde{O}(p)$ implementations of Algorithm 5.16 is perhaps more surprising: Algorithm 5.16u outperforms Algorithm 5.16c by a wide margin. This is explained by the fact that in our $\widetilde{O}(p)$ implementation of Algorithm 5.16u we exploit the shape of the 28×28 matrices $M(t)$ defined in Remark 5.17: as can be seen from (4.12), it has only $7\cdot 22+21=165<256=16^2$ nonzero entries. As noted in Remark 5.19, in our $\widetilde{O}(p)$ implementation we iteratively compute matrix-vector products, which lets us exploit the

sparsity of the uncompressed $M(t)$ (the compressed matrices are not sparse). Additionally, the uncompressed $M(t)$ have degree 1 rather than 2, which provides a further speedup.

We also analyzed the performance of the three average polynomial-time algorithms introduced in this paper: the compressed and uncompressed versions of Algorithm 5.22 and the algorithm based on [Har15, Thm. 4.1]. Table 2 lists the total time and space, and average time per prime, to compute the Cartier–Manin matrices of the reductions modulo p of a fixed smooth plane quartic curve over $\mathbb{Q}$ for good primes $p \leqslant N = 2^n$ for $n = 10, 11, \ldots, 23$. We used a dense polynomial $f \in \mathbb{Z}[x_0, x_1, x_2]_4$ with small (single digit) coefficients as input to all three algorithms. The parameter κ that determines the number 2^κ of trees in the remainder forest was chosen to optimize the running time; for $N = 2^{18}, \ldots, 2^{23}$ this led us to use $\kappa = 6$ for both versions of Algorithm 5.22 and $\kappa = 7$ for the algorithm based on [Har15, Thm. 4.1], which is close to the asymptotic value $\kappa = \lfloor 2 \log_2 \log_2 N \rfloor$ used in Theorem 5.21.

Remark 6.2. For the algorithm based on [Har15, Thm. 4.1], at small values of N the optimal value of κ is actually $\log_2 N$, meaning that each "tree" in the forest consists of a single matrix. This choice of κ leads to an $\widetilde{O}(N^2)$ time complexity but is advantageous for small values of N because it allows the algorithm to avoid full matrix multiplications via Remark 5.29. This explains the rapid growth in the running times for this algorithm for $N \leqslant 2^{17}$.

In addition to κ, the memory used by our algorithms is influenced by the matrix dimensions and the size of the matrix coefficients. To get a better understanding of these parameters, we analyzed the computation of a single product tree in the middle of a remainder forest with $N = 2^{24}$ and $\kappa = 6$ for all three algorithms. The results are shown in Table 3, in which one can see the growth in the size of the matrix coefficients at each level of the product tree in the "KB/entry" columns, the total size of all the matrices in each level in the "MB" columns, and the total time per level. The decrease in the total size of the matrices in the first few layers of the product tree for Algorithm 5.22c is explained by Remark 5.24.

Remark 6.3. In our implementation we use the algorithm for integer matrix multiplication described in [HvdH18]. As explained in the proof of Theorem 5.27, this algorithm computes the product of $r \times r$ matrices with b-bit entries in time $O(r^2 b \log b + r^\omega b \log\log b)$, provided that $\log r = O(\log b)$. This becomes $O(r^2 b \log b)$ when b is large relative to r, as in the context of Theorem 5.21 where we have $r = O(\log B)$, and in Theorem 5.27 where $r = O(1)$. But for the small values of b that arise in the lower levels of the product tree the constant factors make this approach less efficient than naïve matrix multiplication, so we use the algorithm of [HvdH18] only once it becomes faster to do so. These crossover points are indicated by thin horizontal lines in Table 3. Given that r is fixed in all the algorithms we consider, we made no attempt to achieve the optimal value of ω in our implementation; doing so might have improved the relative performance of the algorithm with $r = 66$ in the range we tested.

In Table 3 one can see that the matrix coefficient sizes roughly double in each level while the number of matrix products is cut in half, and the total size of the products in each level is essentially constant in the top half of each tree. Asymptotically, the time to build each layer of the product tree is quasilinear in the total size, so for sufficiently large $N/2^\kappa$ one would expect the relative running times of the three algorithms in the top half of the tree to approach the ratios of these total sizes, which are roughly $1 : 2.6 : 7.0$ for the algorithms with $r = 16, 28, 66$, respectively. The ratios of the actual times to build these trees for $N = 2^{24}$ are

	Cartier–Manin matrix							point counting	
	Algorithm 5.16c		Algorithm 5.16u		[Har15] (optimized)		[Cos15]	`smalljac`	`magma`
p	$p^{1/2}\log^2 p$	$p\log p$	$p^{1/2}\log^2 p$	$p\log p$	$p^{1/2}\log^2 p$	$p\log p$	$p\log p$	$p\log^2 p$	$p\log^2 p$
$2^{10}+7$	0.003	0.001	0.002	0.000	0.022	0.001	0.014	0.000	0.000
$2^{11}+5$	0.003	0.001	0.003	0.000	0.029	0.003	0.017	0.001	0.010
$2^{12}+3$	0.004	0.002	0.004	0.000	0.041	0.006	0.023	0.001	0.020
$2^{13}+17$	0.004	0.004	0.006	0.001	0.056	0.011	0.035	0.002	0.040
$2^{14}+27$	0.005	0.009	0.008	0.002	0.081	0.023	0.058	0.004	0.070
$2^{15}+3$	0.006	0.017	0.012	0.003	0.113	0.047	0.112	0.008	0.140
$2^{16}+1$	0.008	0.033	0.018	0.006	0.175	0.089	0.192	0.023	0.300
$2^{17}+29$	0.011	0.066	0.028	0.012	0.255	0.184	0.372	0.039	0.620
$2^{18}+3$	0.017	0.130	0.047	0.024	0.402	0.368	0.718	0.078	1.23
$2^{19}+21$	0.025	0.263	0.072	0.047	0.598	0.735	1.43	0.158	2.62
$2^{20}+7$	0.039	0.527	0.119	0.092	0.956	1.41	2.84	0.324	5.50
$2^{21}+17$	0.060	1.05	0.186	0.188	1.47	2.84	5.65	0.740	11.4
$2^{22}+15$	0.100	2.11	0.318	0.370	2.41	5.65	11.3	1.47	23.9
$2^{23}+9$	0.154	4.15	0.488	0.736	3.69	11.8	22.6	2.93	48.3
$2^{24}+43$	0.269	8.43	0.858	1.46	6.26	23.4	44.9	6.44	99.3
$2^{25}+35$	0.421	16.6	1.35	2.93	9.73	45.2	89.9	13.6	201
$2^{26}+15$	0.735	33.7	2.36	5.83	16.8	90.4	180	26.9	723
$2^{27}+29$	1.16	66.4	3.68	11.7	27.4	188	360	54.5	1530
$2^{28}+3$	1.95	135	6.14	23.4	44.5	361	719	114	3080
$2^{29}+11$	2.90	265	9.04	46.7	68.5	750	1440	230	6430
$2^{30}+3$	4.89	539	15.1	93.1	119	1480	3130	465	13600

TABLE 1. Algorithms for smooth plane quartics over $\mathbb{F}_p$. Times in 5.2GHz Intel i9-12900K core-seconds. Complexities ignore $O(\log\log p)$ factors. The point counting computations only determine the trace of the Cartier–Manin matrix.

	Algorithm 5.22c			Algorithm 5.22u			[Har15] (optimized)		
N	seconds	ms/p	GB	seconds	ms/p	GB	seconds	ms/p	GB
2^{10}	0.060	0.355	0.042	0.151	0.903	0.033	0.092	0.550	0.034
2^{11}	0.135	0.444	0.043	0.395	1.30	0.035	0.219	0.719	0.034
2^{12}	0.280	0.500	0.044	1.12	2.01	0.035	0.592	1.06	0.034
2^{13}	0.648	0.633	0.047	3.60	3.51	0.036	1.84	1.80	0.035
2^{14}	1.47	0.774	0.053	7.00	3.69	0.077	6.66	3.34	0.035
2^{15}	3.62	1.03	0.067	15.9	4.54	0.123	24.2	6.89	0.037
2^{16}	8.08	1.24	0.088	36.9	5.65	0.217	74.4	11.4	0.040
2^{17}	19.2	1.57	0.131	85.2	6.96	0.410	252	20.5	0.071
2^{18}	44.8	1.95	0.223	192	8.37	0.805	676	29.4	0.910
2^{19}	106	2.44	0.413	437	10.1	1.63	1680	38.6	2.38
2^{20}	241	2.94	0.790	991	12.1	3.29	4100	50.0	4.91
2^{21}	543	3.49	1.57	2230	14.3	6.73	10800	69.3	10.1
2^{22}	1260	4.26	3.20	5040	17.0	13.8	29900	101	20.9
2^{23}	2950	5.23	6.57	11400	20.3	28.4	88200	156	43.2

TABLE 2. Average polynomial-time algorithms for smooth plane quartics over $\mathbb{Q}$ with small coefficients. Times in 5.2GHz Intel i9-12900K core-seconds.

	Algorithm 5.22c ($r = 16$)			Algorithm 5.22u ($r = 28$)			[Har15] (optimized) ($r = 66$)		
products	KB/entry	MB	seconds	KB/entry	MB	seconds	KB/entry	MB	seconds
2^{17}	0.014	457	2.91	0.005	469	6.62	0.003	1890	87.2
2^{16}	0.029	470	2.95	0.015	776	6.21	0.009	2508	70.7
2^{15}	0.055	449	2.28	0.039	989	7.37	0.019	2624	53.5
2^{14}	0.103	420	2.44	0.079	996	7.07	0.038	2679	36.3
2^{13}	0.198	406	2.62	0.159	999	8.68	0.078	2708	31.8
2^{12}	0.389	399	3.58	0.319	1001	13.2	0.156	2723	46.0
2^{11}	0.772	395	3.71	0.639	1002	14.4	0.313	2730	73.6
2^{10}	1.54	393	3.44	1.28	1003	13.6	0.628	2734	79.6
2^{9}	3.07	392	3.39	2.56	1003	14.0	1.26	2736	77.6
2^{8}	6.13	392	3.43	5.12	1003	14.1	2.51	2737	76.8
2^{7}	12.2	392	3.51	10.2	1003	14.4	5.03	2737	76.5
2^{6}	24.5	392	3.81	20.5	1003	15.0	10.1	2737	77.9
2^{5}	49.0	392	3.90	40.9	1003	15.2	20.1	2738	80.0
2^{4}	97.9	392	4.05	81.9	1003	15.5	40.2	2738	80.8
2^{3}	196	392	4.18	164	1003	16.0	80.4	2738	82.0
2^{2}	392	392	4.37	328	1003	16.5	161	2738	84.1
2	783	392	4.52	655	1003	17.1	322	2738	85.7
1	1570	392	5.80	1310	1003	21.0	644	2738	96.4

TABLE 3. Computation of a product tree in the middle of a remainder forest with $N = 2^{24}$ and $\kappa = 6$ involving the product of $N/2^{\kappa} = 2^{18}$ $r \times r$ matrices. The "MB" columns list the total size of the products in megabytes. Horizontal lines indicate matrix multiplication algorithm crossovers. Times in 5.2GHz Intel i9-12900K core-seconds.

approximately $1 : 3.6 : 20.0$, a discrepancy that is likely explained by lower order complexity terms involving r^{ω} and the greater frequency of cache misses for larger total bit sizes.

Remark 6.4. Table 3 only captures the cost of building a product tree in the remainder forest, which is less than half the total running time (for the time-optimal value of κ). The other phases of the algorithm (transferring information between product trees and computing remainders down the trees) involve computations on matrices that one can assume have been reduced modulo m, where m is either the product of all remaining moduli, or the product of the moduli in some subtree. The values of m will be the same in all three algorithms, so one would asymptotically expect the relative costs of these phases to converge to the relative ratios of $2r^2$ for $r = 16, 28$ and $3r + r^2$ for $r = 66$ (via Remark 5.29), which are $1 : 3.1 : 8.9$.

Remark 6.5. As in Table 2, the data in Table 3 reflects a curve with small coefficients, which is the case we expect to most often arise in practice (as in [Sut19], for example). To assess the performance of our algorithms on curves with larger coefficients we also tested random curves with 10 and 100 digit coefficients with $N = 2^{24}$ using $\kappa = 8$ and $\kappa = 10$. As in Table 3, the total size of the matrix products at each level stabilizes in the top half of the product tree, as do the relative running times. For 10-digit coefficients the relative size ratios are $1 : 2.8 : 2.7$ and the time ratios are $1 : 3.5 : 6.0$ (for the algorithms with $r = 16, 28, 66$,

respectively), and for 100-digit coefficients the relative size ratios are $1 : 2.7 : 1.8$ and the time ratios are $1 : 2.4 : 2.7$ (as noted above, these ratios are relevant only to the build phase).

Finally, we compared the performance of Algorithm 5.22c to average polynomial-time algorithms that are applicable to various types of genus 3 curves over $\mathbb{Q}$, including:

- The algorithm in [HMS16] for computing Cartier–Manin matrices of reductions of a geometrically hyperelliptic curve of genus 3 defined over $\mathbb{Q}$ with a model of the form $g(x, y, z) = 0, w^2 = f(x, y, z)$, where g is a pointless conic and $\deg f = 4$.
- The algorithm in [HS16] for computing Cartier–Manin matrices of reductions of a hyperelliptic curve over $\mathbb{Q}$, applied to a genus 3 curve $y^2 = f(x)$ with $\deg f = 8$, which is a 2-cover of $\mathbb{P}^1$.
- The algorithm in [Sut20] for computing the Cartier–Manin matrices of reductions of superelliptic curves $y^m = f(x)$ over $\mathbb{Q}$ applied to genus 3 curves of the form $y^3 = f(x)$ and $y^4 = f(x)$ with $\deg f = 4$ (the case $y^3 = f(x)$ is a Picard curve).
- The algorithm for smooth plane quartics of the form $x^4 + h(y, z)x^2 = f(y, z)$ (these are degree 2 covers of genus 1 curves) described in Remark 5.26.

The results appear in Table 4, which reflects curves defined by dense polynomials with random single digit coefficients. All of these implementations use the RemainderForest algorithm and the same libraries for multiplying polynomials and matrices over $\mathbb{F}_p$ and $\mathbb{Z}$, based on [Har10] and [HvdH18]. None of these computations required more than 64GB memory, but the computations for smooth plane quartics were the most memory intensive.

N	plane quartic	geometrically hyperelliptic	rationally hyperelliptic	2-cover of a genus 1 curve	3-cover of $\mathbb{P}^1$	4-cover of $\mathbb{P}^1$
2^{10}	0.058	0.053	0.007	0.021	0.006	0.006
2^{11}	0.158	0.069	0.008	0.035	0.007	0.007
2^{12}	0.281	0.126	0.011	0.070	0.008	0.008
2^{13}	0.638	0.294	0.022	0.139	0.013	0.012
2^{14}	1.49	0.724	0.065	0.326	0.030	0.028
2^{15}	3.43	2.12	0.222	0.742	0.086	0.089
2^{16}	8.00	5.42	0.829	1.77	0.333	0.285
2^{17}	19.1	12.4	3.25	4.24	0.882	0.760
2^{18}	44.6	29.6	10.0	10.1	2.38	2.15
2^{19}	105	69.5	24.4	24.2	6.67	5.48
2^{20}	241	168	55.6	57.2	15.3	12.2
2^{21}	543	388	133	133	36.1	29.6
2^{22}	1260	921	320	315	87.6	72.0
2^{23}	2950	2160	746	748	214	173
2^{24}	6840	4860	1760	1750	514	410
2^{25}	15600	11200	4120	4050	1220	975
2^{26}	35600	26000	9560	9370	2880	2350

Table 4. Average polynomial-time algorithms for various genus 3 curves over $\mathbb{Q}$ with small coefficients. Times in 5.2GHz Intel i9-12900K core-seconds.

References

[AH19] Jeffrey D. Achter and Everett W. Howe, *Hasse–Witt and Cartier–Manin matrices: A warning and a request*, Arithmetic Geometry: Computations and Applications, Contemporary Mathematics **722** (2019), 1–18, American Mathematical Society. (MathSciNet: MR3896846, arXiv: 1710.10726v5) 4

[AH01] Leonard M. Adleman and Ming-Deh Huang, *Counting points on curves and abelian varieties over finite fields*, International Algorithmic Number Theory Symposium (ANTS I), LNCS **1122** (1996), 1–16, Springer. (MathSciNet: MR1446493) 2

[Bat93] Victor V. Batyrev, *Variations of the mixed Hodge structure of affine hypersurfaces in algebraic tori*, Duke Math J. **69** (1993). (MathSciNet: MR1203231) 7

[Magma] Wieb Bosma, John Cannon, and Catherine Playoust, *The Magma algebra system. I. The user language*, J. Symbolic Comput. **24** (1997), 235–265. (MathSciNet: MR1484478) 26

[BGS07] Alan 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), 1777–1806. (MathSciNet: MR2299425, HAL-Inria: 00103401) 22, 26

[CV09] Wouter Castryck and John Voight, *On nondegeneracy of curves*, Algebra Number Theory **3** (2009), 255–281. (MathSciNet: MR2525551) 7

[Cos15] Edgar Costa, *Effective computations of Hasse-Weil zeta functions*, Ph.D. thesis, New York University, 2015 (MathSciNet: MR3419250) 26, 28

[Cos15a] Edgar Costa, `PycontrolledReduction`, GitHub repository, `https://github.com/edgarcosta/controlledreduction` (retrieved March 2021) 26

[CHS22] Edgar Costa, David Harvey, and Andrew V. Sutherland, `SPQPointcounting`, Jupyter notebook, `https://cocalc.com/AndrewVSutherland/SPQPointCounting/ToyImplementation` (2022). 25

[FKS21] Francesc Fité, Kiran S. Kedlaya, and Andrew V. Sutherland, *Sato–Tate groups of abelian threefolds: a preview of the classification*, in Arithmetic Geometry, Cryptography, and Coding Theory, Contemp. Math. **770** (2021), 103–129. (MathSciNet: MR4280389, arXiv: 1911.02071) 3

[FKS22] Francesc Fité, Kiran S. Kedlaya, and Andrew V. Sutherland, *Sato–Tate groups of abelian threefolds*, preprint. arXiv: 2106.13759 3

[FOR08] Stéphane Flon, Roger Oyono, and Christophe Ritzenthaler, *Fast addition on non-hyperelliptic genus 3 curves*, in Algebraic Geometry and its Applications, Ser. Number Theory Appl. **5** (2008) 1–28, World Sci. Publ. (MathSciNet: MR2484046, IACR: 2004/118) 2

[GG13] Joachim von zur Gathen and Jürgen Gerhard, *Modern computer algebra*, third edition, Cambridge University Press, 2013. (MathSciNet: MR3087522) 21

[GKZ94] Israel M. Gelfand, Mikhail M. Kapranov, Andrei V. Zelevinsky, *Discriminants, resultants, and multidimensional determinants*, Birkhäuser, 1994. (MathSciNet: MR2394437) 7, 15

[Har10] David Harvey, *A cache-friendly truncated FFT*, Theoret. Comput. Sci. **410** (2009), 2649–2658. (MathSciNet: MR2531107, arXiv: 0810.3203) 26, 30

[Har15] David Harvey, *Computing zeta functions of arithmetic schemes*, Proc. Lond. Math. Soc. **111** (2015), 1379–1401. (MathSciNet: MR3447797 arXiv: 1402.3439) 2, 3, 25, 26, 27, 28, 29

[HvdH18] David Harvey and Joris van der Hoeven, *On the complexity of integer multiplication*, J. Symbolic Comput. **89** (2018). (MathSciNet: MR3804803, HAL: 01071191) 23, 27, 30

[HvdH21] David Harvey and Joris van der Hoeven, *Integer multiplication in time $O(n \log n)$*, Annals of Math. **193** (2021), 563–617. (MathSciNet: MR4224716, HAL: 02070778) 21, 23

[HMS16] David Harvey, Maike Massierer, and Andrew V. Sutherland, *Computing L-series of geometrically hyperelliptic curves of genus three*, in Algorithmic Number Theory 12th International Symposium (ANTS XII), LMS J. Comput. Math. **19A** (2016), 220–234. (MathSciNet: MR3540957, arXiv: 1605.04708) 2, 30

[HS14] David Harvey and Andrew V. Sutherland, *Computing Hasse–Witt matrices of hyperelliptic curves in average polynomial time*, Algorithmic Number Theory 11th International Symposium (ANTS XI), LMS J. Comput. Math. **17A** (2014), 257–273. (MathSciNet: MR3240808, arXiv: 1402.3246) 2, 23

[HS16] David Harvey and Andrew V. Sutherland, *Computing Hasse–Witt matrices of hyperelliptic curves in average polynomial time, II*, in Frobenius Distributions: Lang–Trotter and Sato–Tate Conjectures,

Contemp. Math. **663** (2016), 127–147, American Mathematical Society. (MathSciNet: MR3502941 arXiv: 1410.5222) 2, 23, 24, 30

[Katz73] Nicholas M. Katz, *Une formule de congruence pour la function ζ*, in Groups de Monodromie en Géométrie Algébrique, Lecture Notes in Mathematics **340** (1973), 401–438, Springer. (MathSciNet: MR0354657) 5

[KS08] Kiran S. Kedlaya and Andrew V. Sutherland, *Computing L-series of hyperelliptic curves*, Algorithmic Number Theory 8th International Symposium (ANTS VIII), Lecture Notes in Computer Science **487** (2009), 119–162, Springer. (MathSciNet: MR2467855, arXiv: 0801.2778) 2, 25, 26

[Ma1916] F.S. Macaulay, *The algebraic theory of modular systems*, Cornell Hist. Math Monographs, 1916. (MathSciNet: MR1281612) 6, 7

[Man65] Ju. I. Manin, *The Hasse–Witt matrix of an algebraic curve*, in Fifteen Papers on Algebra, Amer. Math. Soc. Transl. **45** (1965) 245–264, translated by J.W.S. Cassels. (MathSciNet: MR0124324) 5

[Pil90] Jonathan Pila, *Frobenius maps of abelian varieties and fining roots of unity in finite fields*, Math. Comp. **55** (1990), 745–763 (MathSciNet: MR1035941) 2

[Sage] The Sage Developers, SageMath, the Sage Mathematics Software System Version 9.4, available at https://www.sagemath.org, 2021. 25

[Sti09] Henning Stichtenoth, *Algebraic function fields and codes*, Graduate Texts in Mathematics **254**, Springer, 2009. (MathSciNet: MR2464941) 3, 4

[SV87] 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. (MathSciNet: MR0887399) 4

[Sch85] René Schoof, *Elliptic curves over finite fields and the computation of square roots mod p*, Math. Comp. **44** (1985), 483–494. (MathSciNet: MR777280) 2

[Sut07] Andrew V. Sutherland, *Order computations in generic groups*, PhD Thesis, Massachusetts Institute of Technology, 2007. (MathSciNet: MR2717420) 2

[Sut09] Andrew V. Sutherland, *A generic approach to searching for Jacobians*, Math. Comp. **78** (2009), 485–507. (MathSciNet: MR2448717, arXiv: 0708.3168) 2

[Sut19] Andrew V. Sutherland, *A database of nonhyperelliptic curves over $\mathbb{Q}$*, Thirteenth Algorithmic Number Theory Symposium (ANTS XIII), Open Book Series **2** (2019), 443–459. (MathSciNet: MR3952027, arXiv: 1806.06289) 3, 29

[Sut20] Andrew V. Sutherland, *Counting points on superelliptic curves in average polynomial time*, Fourteenth Algorithmic Number Theory Symposium (ANTS XIV), Open Book Series **4** (2020), 403–422. (MathSciNet: MR4235126, arXiv: 2004.10189) 2, 3, 5, 23, 30

[Tui16] Jan Tuitman, *Counting points on curves using a map to $\mathbb{P}^1$*, Math. Comp. **85** (2016), 961–981. (MathSciNet: MR3434890, arXiv: 1402.6758) 26

Department of Mathematics 77 Massachusetts Ave. Cambridge, MA 02139, USA
Email address: edgarc@mit.edu
URL: https://edgarcosta.org/

School of Mathematics and Statistics, University of New South Wales, Sydney NSW 2052, Australia
Email address: d.harvey@unsw.edu.au
URL: http://web.maths.unsw.edu.au/~davidharvey/

Department of Mathematics 77 Massachusetts Ave. Cambridge, MA 02139, USA
Email address: drew@math.mit.edu
URL: https://math.mit.edu/~drew/