MATHEMATICS OF COMPUTATION
Volume 76, Number 260, October 2007, Pages 2213–2239
S 0025-5718(07)01989-8
Article electronically published on April 23, 2007

# ASYMPTOTICALLY FAST GROUP OPERATIONS ON JACOBIANS OF GENERAL CURVES

KAMAL KHURI-MAKDISI

ABSTRACT. Let $C$ be a curve of genus $g$ over a field $k$. We describe probabilistic algorithms for addition and inversion of the classes of rational divisors in the Jacobian of $C$. After a precomputation, which is done only once for the curve $C$, the algorithms use only linear algebra in vector spaces of dimension at most $O(g \log g)$, and so take $O(g^{3+\epsilon})$ field operations in $k$, using Gaussian elimination. Using fast algorithms for the linear algebra, one can improve this time to $O(g^{2.376})$. This represents a significant improvement over the previous record of $O(g^4)$ field operations (also after a precomputation) for general curves of genus $g$.

## 1. INTRODUCTION

Let $C$ be a smooth projective geometrically irreducible algebraic curve of genus $g$ over a field $k$. The Jacobian variety $J$ of $C$ is a $g$-dimensional algebraic group that parametrizes the degree zero divisors on $C$, up to linear equivalence. The Jacobian plays a crucial role both in the theory and in the applications of the curve $C$, including cryptography and computational number theory. For all but the smallest $g$, it appears impractical to implement the group $J(k)$ algorithmically using an embedding of $J$ into a projective space $\mathbf{P}^N$: if we embed $J$ using the complete linear series attached to $3\Theta$ or $4\Theta$ ($\Theta$ being the theta divisor), then the equations of $J$ can be described, but the dimension $N$ grows exponentially with $g$; on the other hand, if we use an incomplete linear series, then the equations defining $J$ become much more complicated. Instead, algorithms for $J(k)$ generally work directly with $k$-rational divisors on $C$, and keep track of their linear equivalence to reduce "complicated" divisors to simpler ones as needed. This gives a computational handle on the Picard group, $\mathrm{Pic}^0_k(C)$, which is a subgroup of $J(k)$ (the two groups agree if $C(k)$ is nonempty). We shall nevertheless frequently abuse terminology and refer to the Jacobian instead of to the Picard group.

In this article, we present what we believe are asymptotically the fastest algorithms to date that implement the group law on the Picard group of a *general* curve $C$, as the genus $g$ grows. This assumes that $C$ is given in one of two specific forms, which we call "Representation A" and "Representation B," with respect to which we can also represent divisors $D$ on $C$. If we start with equations for $C$, we need to do a single initial precomputation to bring $C$ into one of these two forms. For Representation A, this involves computing two Riemann-Roch spaces of the form $H^0\big(\mathcal{O}_C(D_1)\big)$ on $C$ and a setting up a "multiplication table" $\mu$ between them, once

Received by the editor July 3, 2006 and, in revised form, August 20, 2006.
2000 *Mathematics Subject Classification.* Primary 11Y16, 14Q05, 14H40, 11G20.

and for all. For Representation B, we also need to describe the values of a basis for a space $H^0\big(\mathcal{O}_C(D_1)\big)$ at sufficiently many points of $C$; this allows us to speed up the multiplication $\mu$ in a way analogous to representing polynomials by their values at many points instead of by their coefficients. After that, our algorithms boil down to linear algebra on certain matrices of size $O(g) \times O(g \log g) = O(g) \times O(g^{1+\epsilon})$, which arise from subspaces of the Riemann-Roch spaces above. For Representation A, our algorithms attain a complexity of $O(g^{3+\epsilon})$ field operations in $k$ per group operation (such as addition or negation) in the Jacobian, and this complexity holds whether we use Gaussian elimination or we use asymptotically faster algorithms for linear algebra. In the case of Representation B, the complexity is determined by the linear algebra. The current best algorithms [CW90] allow us to attain a complexity of $O(g^{2.376})$ using Representation B. Our algorithms are straightforward to implement and analyze — the author had an easy time programming the algorithms for the Jacobian group in GP/PARI [PARI], for the case of "Representation A," in a fairly short program file — but we naturally need more sophisticated techniques to prove that our algorithms give the correct answer.

Our algorithms are probabilistic, since they have to find certain intermediate data ("an IGS" of a divisor $D$, defined in Section 3) for the computation by random search; the above complexity actually describes the expected number of field operations needed by our algorithms. Each trial to find an IGS for $D$ has a probability of success greater or equal to $1/2$, and we can recognize an IGS once we have found it, so our algorithms are guaranteed to terminate with a correct result. Thus our probabilistic algorithms are of Las Vegas type. We have measured complexity by counting field operations in $k$ instead of, say, bit operations, due to potential "coefficient explosion" in $k$. This is not an issue if $k$ is finite, but is unavoidable if $k = \mathbf{Q}$ (more generally, for number fields), since adding points on the Jacobian tends to increase their arithmetic height. This growth of coefficients will occur even if we carry out our linear algebra over $\mathbf{Q}$ in the best possible way, for example, by incorporating LLL reduction throughout our algorithms.

Prior to the results of this article, the best algorithms for Jacobians of general curves had a complexity of $O(g^4)$ after the initial precomputations, and were deterministic. The complexity $O(g^4)$ was attained both in the 1999 Ph.D. thesis of F. Hess [Hes99] (see also [Hes02]), and in a 2001 preprint of the author (published as [KM04a]), whose methods we adapt and extend for this article. The methods of Hess, and of several predecessors of whom we cite only [Can87], can be called "arithmetic": they begin with a degree $n$ map $\varphi : C \to \mathbf{P}^1$, and view the function field $k(C)$ as a degree $n$ extension of $k(x)$. Then $J(k)$ is essentially an ideal class group attached to $k(C)$, and we compute with ideals of the integral closure of $k[x]$ in $k(C)$ by representing them as lattices (i.e., free modules) over $k[x]$; one has also to consider the points of $C$ lying over $\infty \in \mathbf{P}^1$, and the implementation is somewhat involved. The methods of Hess and his predecessors work best if the minimum gonality $n = \deg \varphi$ remains bounded as $g$ grows (for example, [Can87] applies only to hyperelliptic curves, for which $n = 2$); in that case, their algorithms generally have complexity $O(g^2)$. However, their methods are sensitive to $n$, and

if $n$ grows linearly with $g$, as is the case[1] for general curves of genus $g$, then the complexity of Hess' algorithms rises to $O(g^4)$, as mentioned above.

In contrast, the methods in [KM04a] and in the present article can be called "geometric," in that we work with an embedding $\iota : C \to \mathbf{P}^n$. We choose $n$ moderately large, but still $O(g)$, and the two Riemann-Roch spaces that we need to compute are the restriction of linear and quadratic functions from the projective space to $C$. The "multiplication map" $\mu$ then multiplies two linear functions to produce a quadratic function. Once this is in place, the rest is linear algebra (the reader may wish to compare our approach with another use of linear algebra to study Jacobians in [And02], where linear algebra on Riemann-Roch spaces and invariant theory are used to describe explicit equations for the Jacobian). In contrast to our methods, earlier "geometric" algorithms for Jacobians ([HI94] and [Vol94]) preferred to work with $n$ as small as possible, preferably $n = 2$, even if this meant using a singular plane curve birational to $C$. Their algorithms involved fairly elaborate computations with polynomials of degree $O(g)$, to say nothing about the problems with singularities. The resulting complexity of those algorithms was $O(g^7)$ after precomputations, and so those methods were superseded by the algorithms of Hess. The author hopes that this article and its predecessor [KM04a] will revive interest in the geometric approach to algorithms for curves.

We also hope that this article will support a point of view explained in the introduction of [KM04a], namely, that it is profitable to do computational algebraic geometry with varieties embedded in Grassmannians. Here we represent points on Grassmannians as subspaces of a fixed vector space $V$, and use linear algebra throughout; we do *not* embed the Grassmannian variety into projective space, as the ambient projective space would be too large. In our setting, we represent a divisor $D$ of degree $d$ as a codimension $d$ subspace $W_D$ of $V$, which we can interpret as mapping the symmetric power variety $\mathrm{Sym}^d(C)$ into a Grassmannian. We take $d \geq 2g$, instead of the more usual approach $d = g$, because this simplifies our algorithms (essentially since the fibre in $\mathrm{Sym}^d(C)$ over a point of the Jacobian always has the same structure, a point used notably in Chow's projective construction of the Jacobian [Cho54]). We of course include an algorithm that determines whether two elements of $\mathrm{Sym}^d(C)$ represent the same point on the Jacobian. For all of this and more background, the reader is encouraged to consult [KM04a] alongside this article.

The speedup in our new algorithms comes partially from the speedup of multiplication in Representation B; however, the most significant improvement is due to our using an IGS for $D$ instead of the whole space $W_D$ at some strategic moments. This allows us to scale down the size of the matrices on which we need to do linear algebra, from $O(g) \times O(g^2)$ in [KM04a] to $O(g) \times O(g^{1+\epsilon})$ in this article. It turns out that the larger matrices of [KM04a] contain redundant data, but it is still not clear if one can remove the redundant data by a fast deterministic algorithm. This is why our algorithms are probabilistic.

[1] By [GH94], page 261, a general curve of genus $g$ over $\mathbf{C}$, or more generally over an algebraically closed field, has gonality $\lfloor (g+1)/2 \rfloor + 1$; over $k$, the gonality can be higher. We also note the result of [Abr96] that the gonality of a modular curve such as $X_0(N)$ also grows linearly with the genus, at least over $\mathbf{C}$; interestingly, our algorithms are particularly suited for modular curves, since it is easier to describe them using Representation A or Representation B than by finding nice equations.

**Acknowledgments.** The author gratefully thanks the following institutes for their support during the periods when the main results of this article and its predecessor were obtained: the Clay Mathematics Institute, for funding research visits to the U.S. in the summers of 2000 and 2003; the William and Flora Hewlett Foundation, for supporting a semester of paid research leave at the American University of Beirut in fall 2002; and the TEMPUS program of the European Union, for funding a research visit to France and Austria in the summer of 2003. The author also thanks Université Paris XIII and Princeton University for their hospitality during the initial stages of preparing the manuscript. The author gratefully thanks G. Frey for an invitation to Essen to lecture on the results of this article in the summer of 2005, and for many stimulating conversations on that occasion with him and with C. Diem and F. Hess, to whom the author is also grateful. Finally, the author thanks the referees of this article, who carefully read a rather lengthy earlier draft [KM04b], and made extensive comments and suggestions that helped to streamline the presentation and to produce a much improved and more compact version of this article. This version omits a few details and auxiliary results, particularly in Section 5, and the reader who wishes more explanations of those points may wish to read selected portions of the earlier manuscript.

*Remark* 1.1. We have slightly changed notation between this paper and [KM04a]. We now use "multiplicative" notation to refer to line bundles on the curve $C$, instead of the "additive" notation that was used in most of the previous paper (actually, the previous paper occasionally used multiplicative notation as well). Here is a small table of old vs. new notation.

| Old Notation: | New Notation: |
|---|---|
| $\mathcal{L}_1 + \mathcal{L}_2$ | $\mathcal{L}_1 \otimes \mathcal{L}_2$ |
| $H^0(D_1 - D_2)$ | $H^0\big(\mathcal{O}_C(D_1 - D_2)\big)$ |
| $H^0(2\mathcal{L} - D - E)$ | $H^0(\mathcal{L}^{\otimes 2}(-D - E))$ |

*Remark* 1.2. Throughout this article, we will view an $m \times n$ matrix $M$ (always with entries in $k$) as a linear transformation from $k^n$ to $k^m$, viewed as column vectors. Thus $v \in k^n$ is mapped to $Mv \in k^m$, and the notations $\ker M$ and $\operatorname{image} M$ should be interpreted accordingly. We shall need to refer to the complexity of the linear algebra steps in our algorithms, which include computing a kernel or an image of $M$ and/or reduced row and column echelon forms, as well as multiplying matrices. (See Chapter 16 of [BCS97] or Chapter 6 of [AHU75] for a reduction of general linear algebra to matrix multiplication.) We denote by $\omega$ the smallest exponent such that linear algebra on square $n \times n$ matrices has complexity $O(n^{\omega+\epsilon})$, measured in field operations in $k$. The current record [CW90] is $\omega < 2.376$, and it is conjectured that $\omega = 2$. Gaussian elimination gives $\omega \leq 3$ elementarily, and in fact the complexity of Gaussian elimination on a rectangular $m \times n$ matrix is $O(mn \min(m, n))$.

*Remark* 1.3. We use in this article the notation $\lceil a \rceil$ for the ceiling of $a \in \mathbf{R}$, i.e.,

$$\lceil a \rceil = \min\{n \in \mathbf{Z} \mid n \geq a\}. \tag{1.1}$$

## 2. Representing the curve. Basic linear algebra operations

In this section, we describe how we represent the curve $C$ and how we implement the basic building blocks of our algorithms via linear algebra. We shall assume that

the curve $C$ comes equipped with a line bundle $\mathcal{L}$ of moderately large, but not too large, degree:

$$\deg \mathcal{L} \geq 2g+2, \qquad \text{but nonetheless} \qquad \deg \mathcal{L} = O(g). \tag{2.1}$$

(We will typically take $\deg \mathcal{L} = 6g$ in applications.) We define $k$-vector spaces $V$ and $V'$ by

$$V = H^0(C, \mathcal{L}) = H^0(\mathcal{L}), \qquad V' = H^0(C, \mathcal{L}^{\otimes 2}) = H^0(\mathcal{L}^{\otimes 2}). \tag{2.2}$$

We also introduce the notation for dimensions and degrees

$$\begin{aligned} \Delta &= \deg \mathcal{L}, & \Delta' &= \deg \mathcal{L}^{\otimes 2} = 2\Delta, \\ \delta &= \dim V = \Delta + 1 - g, & \delta' &= \dim V' = 2\Delta + 1 - g. \end{aligned} \tag{2.3}$$

Note that all the above quantities are $O(g)$. All our algorithms work over the field $k$, but for some of our proofs we need to consider points of $C$ and elements of $V$ defined over the algebraic closure $\overline{k}$.

The most important ingredient in our description of $C$ is then the multiplication map $\mu$ on global sections,

$$\mu : V \otimes V = H^0(\mathcal{L}) \otimes H^0(\mathcal{L}) \to V' = H^0(\mathcal{L}^{\otimes 2}). \tag{2.4}$$

We will use the shorthand notation

$$s \cdot t = \mu(s \otimes t) \in V' \qquad \text{for } s, t \in V. \tag{2.5}$$

Before specifying the precise form in which we represent $\mu$ algorithmically, we note the following.

**Proposition 2.1.** *We can determine $C$ and $\mathcal{L}$ up to isomorphism from a knowledge of the multiplication map $\mu$. Moreover, given vector spaces $V$, $V'$, and a map $\mu$, it is possible to determine whether they come from a pair $(C, \mathcal{L})$ as above.*

*Proof.* For the first statement, assume that we are assured of the existence of some pair $(C, \mathcal{L})$, but that we only know the map $\mu$. We claim that the kernel of $\mu$ encodes equations for $C$. Indeed, consider the embedding $\iota : C \to \mathbf{P}(V) = \mathbf{P}^{\delta-1}$ given by $\mathcal{L}$. Since $\Delta \geq 2g+2$, this embedding is projectively normal (in particular, $\mu$ is surjective), and the homogeneous ideal $I_C \subset \operatorname{Sym}^* V$ defining $\iota(C)$ is generated by quadrics (see for example [Laz89]). Concretely, we can identify $\operatorname{Sym}^* V$ with $k[T_1, \ldots, T_\delta]$, upon choosing a basis $\{T_1, \ldots, T_\delta\}$ of $V$. The kernel of $\mu$ trivially contains all "commutators" $T_i \otimes T_j - T_j \otimes T_i$. After we quotient out by these commutators, the image of $\ker \mu$ inside the symmetric square $\operatorname{Sym}^2 V$ then corresponds to the degree 2 elements of $I_C$. Since these generate $I_C$, we can hence recover $C$; we also obtain $\mathcal{L}$ as the pullback of $\mathcal{O}_{\mathbf{P}(V)}(1)$ to $C$.

For the second statement, we first check that $\mu$ is surjective and symmetric (i.e., $T_i \cdot T_j = T_j \cdot T_i$ for all $i, j$). The kernel of $\mu$, when projected to $\operatorname{Sym}^2 V$, then corresponds to a space of degree 2 polynomials in $k[T_1, \ldots, T_\delta]$, and we let $I$ be the ideal generated by (a basis for) this space of degree 2 polynomials. We then check that the ideal $I$ is saturated and that it defines a smooth projective curve $C$ (e.g., using Gröbner bases). We then determine the degree $\Delta$ and genus $g$ of this curve $C$ from the Hilbert series of $I$; this again gives $\mathcal{L}$ as the pullback of $\mathcal{O}_{\mathbf{P}(V)}(1)$, with $V \subset H^0(\mathcal{L})$. We finally verify that $\dim V = \Delta + 1 - g$ to ensure that $V = H^0(\mathcal{L})$, i.e., that our embedding of $C$ comes from the *complete* linear series. □

Note that in light of the above proposition, we can view $V$ as the space of linear "functions" and $V'$ as the space of quadratic functions *on the curve* $C$ with respect to the projective embedding $\iota$.

For algorithmic purposes, we represent our knowledge of $V$, $V'$, and $\mu$ in either of two ways, Representation A and Representation B, with the former more straightforward, and the latter asymptotically faster. We also single out a simple special case Representation $B_0$ of Representation B, both for reasons of exposition and of ease of implementation. See Example 2.9 at the end of this section for an example of Representation A and Representation B.

(1) **Representation A**: This method works over all fields. We choose bases $\{T_1, \dots, T_\delta\}$ for $V$ and $\{U_1, \dots, U_{\delta'}\}$ for $V'$, thereby identifying $V$ and $V'$ with the spaces of column vectors $k^\delta$ and $k^{\delta'}$. Knowledge of $\mu$ is then encoded as a multiplication table, i.e., by storing the coefficients $c_{ijk}$ in each identity

$$T_i \cdot T_j = \mu(T_i \otimes T_j) = \sum_k c_{ijk} U_k. \tag{2.6}$$

It is convenient to store this information as a collection $\{M_1, \dots, M_\delta\}$ of matrices, each of size $\delta' \times \delta$, such that $M_i$ describes the linear transformation "multiplication by $T_i$" from $V$ to $V'$:

$$M_i = (c_{ijk})_{k,j} = \begin{pmatrix} c_{i11} & c_{i21} & \dots & c_{i\delta 1} \\ c_{i12} & c_{i22} & \dots & c_{i\delta 2} \\ \vdots & \vdots & \ddots & \vdots \\ c_{i1\delta'} & c_{i2\delta'} & \dots & c_{i\delta\delta'} \end{pmatrix}. \tag{2.7}$$

(2) **Representation $B_0$**: We take a divisor $D_1$ such that $\mathcal{L} = \mathcal{O}_C(D_1)$. We also assume that we can find $N = \Delta' + 1$ distinct points $P_1, \dots, P_N \in C(k)$ that are not in the support of $D_1$. This is a nontrivial assumption if $k$ is a number field, but is easy to arrange in cases of interest to cryptography, where $k$ is a finite field of large cardinality. We then represent $V$ and $V'$ as certain subspaces of $k^N$; namely, we have injections of vector spaces $V \to k^N$ and $V' \to k^N$ given by

$$s \mapsto (s(P_1), \dots, s(P_N)), \tag{2.8}$$

viewing $s \in V$ (respectively $V'$) as a meromorphic function on $C$ with poles at $D_1$ (respectively $2D_1$). Then the multiplication map $\mu$ is simply pointwise multiplication, since $s \cdot t$ corresponds to $(s(P_1)t(P_1), \dots, s(P_N)t(P_N))$. (Thus Representation $B_0$ is analogous to representing a polynomial $f(x) \in k[x]$ of bounded degree by its vector of values $(f(a_1), \dots, f(a_N))$ at sufficiently many points, in order to speed up the multiplication of polynomials.) In this setting, we represent $C$ by our knowledge of the subspaces of $k^N$ corresponding to $V$ and $V'$. It is most convenient to store an $N \times \delta$ matrix $A_V$ whose columns are a basis of $V$ (viewed as a subspace of $k^N$), as well as the equivalent data of an $(N - \delta) \times N$ matrix $K_V$ whose kernel is the subspace $V$. It turns out not to be necessary to store a basis for the subspace $V'$, but we can always recover it, if needed, from the fact that $\mu$ in (2.4) is surjective. Note that there is no need to store any information that describes the map $\mu$.

(3) **Representation B**: Even if we cannot find enough $k$-rational points on $C$, we can still work with the following generalization under the mild assumption of (2.10) (e.g., it is sufficient to assume that $k$ is perfect). We take a $k$-rational effective divisor $Z$ on $C$, of degree $N = \deg Z = O(g)$, such that $H^0(\mathcal{L}^{\otimes 2}(-Z)) = 0$. (We chose $N = \Delta' + 1$ and $Z = P_1 + \cdots + P_N$ for Representation $\mathrm{B}_0$.) We then wish to represent elements of $V$ and $V'$ by their "values" at the points of $Z$. Here the values of a global section $s \in V = H^0(\mathcal{L})$ at $Z$ are given by the image of $s$ in $H^0\big(\mathcal{L}_Z\big)$, where we define the sheaf $\mathcal{L}_Z = \mathcal{L}/\mathcal{L}(-Z)$. We similarly define $\mathcal{L}_Z^{\otimes 2} = \mathcal{L}^{\otimes 2}/\mathcal{L}^{\otimes 2}(-Z)$, and view the values of an element of $V'$ at $Z$ as belonging to $H^0\big(\mathcal{L}_Z^{\otimes 2}\big)$. By design, the natural $k$-linear map $V \to H^0\big(\mathcal{L}_Z\big)$ is injective, and similarly for $V'$. Moreover, one can find compatible isomorphisms of sheaves of $\mathcal{O}_C$-modules

$$\varphi : \mathcal{L}_Z \cong \mathcal{O}_Z, \qquad \varphi^{\otimes 2} : \mathcal{L}_Z^{\otimes 2} \cong \mathcal{O}_Z, \tag{2.9}$$

where $\mathcal{O}_Z = \mathcal{O}_C/\mathcal{O}_C(-Z)$. This identifies $V$ and $V'$ as $k$-subspaces of the $N$-dimensional $k$-algebra $\mathcal{A} := H^0\big(\mathcal{O}_Z\big)$, in a way such that the multiplication $\mu$ becomes multiplication in $\mathcal{A}$. Moreover, we need to assume the knowledge of an isomorphism of $k$-algebras:

$$\mathcal{A} \cong k[x]/(h_1(x)) \times \cdots \times k[x]/(h_r(x)). \tag{2.10}$$

We thus represent elements of $\mathcal{A}$ as tuples of polynomials $(f_1(x), \ldots, f_r(x))$ with $\deg f_i < \deg h_i$. The coefficients of the $f_i$ identify $\mathcal{A}$ with $k^N$ as a $k$-vector space; with respect to these coordinates, we can describe $V$ by matrices $A_V$ and $K_V$ as in the case of Representation $\mathrm{B}_0$. However, in this setting, we need to carry around the polynomials $h_1(x), \ldots, h_r(x)$ in order to know the multiplication map $\mu$. Note that multiplying two elements of $\mathcal{A}$ can be done in time $O(N^{1+\epsilon}) = O(g^{1+\epsilon})$ by FFT-based methods.

*Remark* 2.2. It is relatively straightforward to produce Representation A for a curve that is given in a more "classical" representation. For instance, we may be given polynomial equations that describe $C$ in some projective space (where the embedding need not be given by a complete linear series). Alternatively, we may start with a representation of the function field of $C$ as an extension $k(x)[y]$ of the rational function field $k(x)$, given by an equation $f(x, y) = 0$; this is tantamount to choosing a possibly singular plane curve birational to $C$. In either of these two cases, we choose a divisor $D_1$ of suitably large degree $\Delta$, and let $\mathcal{L} = \mathcal{O}_C(D_1)$. We then use standard algorithms ([HI94], [Vol94], [Hes02]) for calculating the Riemann-Roch spaces $V = H^0\big(\mathcal{O}_C(D_1)\big)$ and $V' = H^0\big(\mathcal{O}_C(2D_1)\big)$. The multiplication map $\mu$ is then immediate in terms of the representation of $V$ and $V'$ as subsets of the function field $k(C)$.

Another situation where we can produce Representation A is that of modular curves. If our curve $C$ is the completion of a quotient $\Gamma\backslash\mathcal{H}$ for some congruence subgroup $\Gamma$ acting on the upper half-plane $\mathcal{H}$, then we do not need to compute equations for $C$ directly; instead, we take a suitable weight $n$ (small values such as $n \in \{2, 3, 4\}$ usually suffice), and let $V = \mathcal{M}_n(\Gamma)$ and $V' = \mathcal{M}_{2n}(\Gamma)$ be the spaces of modular forms of weights $n$ and $2n$ with respect to $\Gamma$. The map $\mu$ is then the multiplication of modular forms; one way in which the modular forms can be represented is by their $q$-expansions up to a suitable order $O(q^N)$, where $N$ is large enough to

distinguish elements of $V'$. These $q$-expansions can be efficiently computed using modular symbols (see, e.g., [Ste04]). Note that working with $q$-expansions is essentially Representation B, where the divisor $Z$ is the $N$-fold multiple of the cusp at infinity. The author has also investigated Representation B for modular curves in the setting where one evaluates the form at several noncuspidal points.

*Remark* 2.3. Given a curve in Representation B, one can immediately convert the curve to Representation A. Conversely, given a curve in Representation A, we sketch in Section 5 how to convert this to Representation B, under some assumptions on the field $k$.

*Remark* 2.4. For uniformity of notation, we extend the definition of $N$ so that in the case of Representation A, we have $N = \delta$. Thus both in Representation A and in Representation B, we will identify $V$ with a subspace of $k^N$, viewed as column vectors:

(1) If we use Representation A, then $V = k^N$; in this case we can consider that $A_V$ is the $N \times N$ identity matrix.
(2) If we use Representation B, then $V = \operatorname{image} A_V = \ker K_V$.

Similarly, we define $N'$ by $N' = \delta'$ in the case of Representation A, and $N' = N$ in the case of Representation B, so that $V'$ is identified with a subspace of $k^{N'}$.

We will also need to represent ($k$-rational) subspaces $W \subset V$ and $W' \subset V'$. If $r = \dim W$, then we represent $W$ nonuniquely by an $N \times r$ matrix $A_W$, whose columns give a basis for $W$ (viewing the columns as elements of $V$). Thus we have an inclusion $\operatorname{image} A_W \subset \operatorname{image} A_V$ corresponding to the inclusion $W \subset V$. We similarly represent an $r'$-dimensional subspace $W' \subset V'$ by an $N' \times r'$ matrix $A_{W'}$ with $\operatorname{image} A_{W'} = W'$. Finally, note that the numbers $N$ and $N'$, as well as the smaller $r$ and $r'$, are all $O(g)$, regardless of whether we use Representation A or Representation B.

Our algorithms will represent divisors as certain subspaces of $V$ and of $V'$, and will all involve the following linear algebra techniques:

**Definition 2.5.** Given subspaces $W \subset V$ and $W' \subset V'$, and given elements $s, s_1, \dots, s_h \in V$, we define the following:

(1) The *simple multiplication* $s \cdot W$ is the subspace of $V'$ defined by

$$s \cdot W = \{s \cdot t \mid t \in W\}. \tag{2.11}$$

(2) The *sum of products* $s_1 \cdot W + \cdots + s_h \cdot W \subset V'$ is the usual sum of subspaces. (We can view this as a "full multiplication" between $S = \operatorname{span}\{s_1, \dots, s_h\}$ and $W$.)
(3) The *division* $W' \div \{s_1, \dots, s_h\}$ is the subspace of $V$ given by

$$W' \div \{s_1, \dots, s_h\} = \{t \in V \mid t \cdot s_i \in W', \text{ for all } 1 \le i \le h\}. \tag{2.12}$$

The above operations were used in the algorithms of [KM04a], with $h = O(g)$, but we shall only need the case $h = O(g^\epsilon)$ in this article. We can immediately describe the complexity of the above operations, measured as usual in the number of $k$-operations. The exponent $\omega$ in the complexity of linear algebra was mentioned in Remark 1.2.

**Proposition/Algorithm 2.6.** *Assume that $h = O(g^\epsilon)$. Using Representation A, we can:*

(1) *find one product $s \cdot t$ with complexity $O(g^3)$;*
(2) *compute a simple multiplication $s \cdot W$ with complexity $O(g^3)$;*
(3) *compute a sum of products $s_1 \cdot W + \cdots + s_h \cdot W$ with complexity $O(g^{3+\epsilon})$;*
(4) *compute a division $W' \div \{s_1, \ldots, s_h\}$ with complexity $O(g^{3+\epsilon})$.*

*Proof.* (1) Our representation of elements of $V$ as tuples in $k^N$, via the basis $\{T_i\}$ for $V$, means that we are given $s = \sum_i c_i T_i$ in the form of the column vector ${}^t(c_1, \ldots, c_N)$. It is useful to produce the $N' \times N$ matrix $M_s$ that describes the linear transformation "multiplication by $s$" from $V$ to $V'$:

$$M_s = \sum_i c_i M_i, \qquad M_i \text{ as in (2.7).} \tag{2.13}$$

Also viewing $t$ as a column vector in $k^N$, we then compute $s \cdot t = M_s t$. Here computing $M_s$ has complexity $O(g^3)$, and multiplying $M_s t$ has complexity $O(g^2)$. (Alternatively, we could have expanded $s \cdot t$ using the coefficients $c_{ijk}$ of the multiplication table (2.6), for the same complexity.)

(2) We are given the matrix $A_W$, as in Remark 2.4. Compute the matrix $M_s$ as above, with complexity $O(g^3)$; then form the matrix product $A_{s \cdot W} = M_s A_W$. We remain within complexity $O(g^3)$, even if we use fast matrix multiplication. Note that the naive method of multiplying $s$ by each column of $W$ would have had complexity $O(g^4)$.

(3) Compute the matrices $A_{s_1 \cdot W}, \ldots, A_{s_h \cdot W}$. So far, this requires a complexity of $O(g^3 h)$. Then our desired result is the image of the block matrix $A' = \begin{pmatrix} A_{s_1 \cdot W} & \cdots & A_{s_h \cdot W} \end{pmatrix}$, whose size is $O(g) \times O(gh)$. We then find a basis for image $A'$ by linear algebra, with complexity $O(g^3 h)$ if we use Gaussian elimination, and $O((gh)^{\omega+\epsilon})$ by fast methods. Our total complexity is then $O(g^{3+\epsilon})$.

(4) Let $r' = \dim W'$. From the $N' \times r'$ matrix $A_{W'}$, we use linear algebra to produce an $(N' - r') \times N'$ matrix $K_{W'}$ whose kernel is $W'$; the complexity of this is dominated by what comes next. Then our desired result is

$$A_{(W' \div \{s_1, \ldots, s_h\})} = \ker P, \qquad \text{where } P = \begin{pmatrix} K_{W'} M_{s_1} \\ \vdots \\ K_{W'} M_{s_h} \end{pmatrix}. \tag{2.14}$$

This takes complexity $O(g^3 h)$ to produce the $\{M_{s_i}\}$, then $O(g^{\omega+\epsilon} h)$ to obtain $P$. The matrix $P$ has size $\big((N' - r')h\big) \times N = O(gh) \times O(g)$, and finding its kernel has a complexity of $O((gh)^{\omega+\epsilon})$ (even if we use Gaussian elimination, the time is still dominated by finding the $M_{s_i}$). □

Note that for Representation A, there is no asymptotic advantage to using fast linear algebra; we can carry out the operations of Proposition/Algorithm 2.6 using Gaussian elimination with the same complexity, albeit with a higher implied constant in the $O(\cdot)$ notation. On the other hand, Representation B benefits significantly from fast linear algebra.

**Proposition/Algorithm 2.7.** *Assume that $h = O(g^\epsilon)$. Using Representation B, we can:*

(1) *find one product $s \cdot t$ with complexity $O(g^{1+\epsilon})$;*
(2) *compute a simple multiplication $s \cdot W$ with complexity $O(g^{2+\epsilon})$;*

(3) *compute a sum of products* $s_1 \cdot W + \cdots + s_h \cdot W$ *with complexity* $O(g^{\omega+\epsilon})$*;*
(4) *compute a division* $W' \div \{s_1, \ldots, s_h\}$ *with complexity* $O(g^{\omega+\epsilon})$.

*Proof.* This is largely the same as the previous result, except that the bottleneck caused by finding matrices of the form $M_s$ can be bypassed. We indicate the necessary modifications. Note that if we use Representation $\mathrm{B}_0$, then the first two statements hold without including $\epsilon$ in the exponents.

(1) Recall that we represent $s, t$ as elements of the algebra $\mathcal{A}$ (which is just $k \times \cdots \times k$ for Representation $\mathrm{B}_0$, in which case the result is even easier), and we can multiply two elements of $\mathcal{A}$ by FFT-techniques.
(2) Simply multiply $s$ by each column of $A_W$ separately. We note for later use the fact that the $N' \times N = N \times N$ matrix $M_s$ is block diagonal with a structure that allows fast multiplication by FFT — the matrix $M_s$ is furthermore genuinely diagonal in the case of Representation $\mathrm{B}_0$. Hence the multiplication $M_s A_W$ can be done with complexity $O(g^{2+\epsilon})$. If we want, we can actually produce $M_s$ by directly multiplying $s$ by each element in our basis for $\mathcal{A} \cong k^N$. This also has complexity $O(g^{2+\epsilon})$; it corresponds to replacing $A_W$ by the identity matrix.
(3) Here it only takes complexity $O(g^{2+\epsilon}h)$ to produce the matrix $A'$, so the result follows.
(4) First note that the matrix $P$ must be replaced by a slightly larger matrix $Q$ that includes an extra sub-block $K_V$ with $\ker K_V = V$ as mentioned in our descriptions of Representation $\mathrm{B}_0$ and Representation B:

$$A_{(W' \div \{s_1, \ldots, s_h\})} = \ker Q, \qquad \text{where } Q = \begin{pmatrix} K_V \\ K_{W'} M_{s_1} \\ \vdots \\ K_{W'} M_{s_h} \end{pmatrix}. \tag{2.15}$$

This ensures that elements of $\ker Q$ genuinely belong to $V$, which is a proper subspace of $k^N$. This does not affect the asymptotics of the linear algebra to find $\ker Q$, since $Q$ still has size $O(gh) \times O(g)$. As for finding $Q$ in the first place, note that the product matrices $\{K_{W'} M_{s_i}\}$ can be computed with complexity $O(g^{2+\epsilon})$. This is particularly clear for Representation $\mathrm{B}_0$, since $M_{s_i}$ is a diagonal matrix. The proof in general uses the transposition principle. Indeed, since the complexity using FFT-based algorithms of multiplying $M_{s_i} v$ for any column vector $v \in k^N$ is $O(g^{1+\epsilon})$, it follows that one can just as quickly (perhaps with a "larger" $\epsilon$) multiply $w M_{s_i}$ for any $N$-dimensional *row vector* $w$. Applying this to the rows of $K_{W'}$, we obtain our result. Alternatively, we can give a more pedestrian approach to finding $Q$; this takes a slightly higher complexity of $O(g^{\omega+\epsilon})$, but does not affect the final complexity of division. Simply produce all the matrices $M_{s_i}$, which requires complexity $O(g^{2+\epsilon}h)$, and then multiply them by a fast algorithm with the matrix $K_{W'}$. □

All of our later algorithms will be built up from the operations that we have introduced in the above two Proposition/Algorithms 2.6 and 2.7. We shall use the following terminology.

**Definition 2.8.** A *fast algorithm* is one that requires a complexity of $O(g^{3+\epsilon})$ field operations in $k$ using Representation A, and that requires a complexity of

$O(g^{\omega+\epsilon})$ using Representation B. We will also define *fast probabilistic algorithms* of Las Vegas type to be those whose expected running time is of the above complexity. (Recall that a probabilistic algorithm is called of Las Vegas type if it either returns "failure" with a probability that is bounded above by a fixed $c > 0$, or it returns an answer that is guaranteed to be correct. This is in contrast to Monte Carlo probabilistic algorithms, for which the answer in the first instance may be wrong, also with a bound on the probability of error.)

We conclude this section with a concrete example of a curve as given in Representation A and Representation B, in order to clarify the precise input to our algorithms. More examples can be found in Section 5 of [KM04b].

**Example 2.9.** Let $C$ be the elliptic curve given by the Weierstrass equation $y^2 = x^3 + 1$ over a field $k$ not of characteristic 2 or 3. We choose as our line bundle $\mathcal{L} = \mathcal{O}_C(4P_\infty)$, where $P_\infty \in C(k)$ is the point at infinity. We choose bases for $V$ and $V'$ (which we view as subsets of $k(C)$):

$$\begin{aligned} \{T_1, \ldots, T_4\} &= \{1, x, y, x^2\}, \\ \{U_1, \ldots, U_8\} &= \{1, x, y, x^2, xy, x^3, x^2y, x^4\}. \end{aligned} \tag{2.16}$$

Thus, using Representation A, we would have $T_2 \cdot T_3 = U_5$ and $T_3 \cdot T_3 = U_1 + U_6$. The reader is encouraged to write down the matrices $M_i$ of (2.7), which will be the entire description of our curve $C$; in particular, our representation never works with the variables $x$ and $y$, but only with the multiplication table giving each $T_i \cdot T_j$ in terms of the $U_k$'s.

To illustrate Representation B, we take $k = \mathbf{Q}$, and take the divisor $Z$ of degree $N = 9$ to be

$$\begin{aligned} Z = (0,1) + (-1,0) + (2,3) + (2,-3) \\ + (2+\sqrt{2}, 5+4\sqrt{2}) + (2-\sqrt{2}, 5-4\sqrt{2}) + 3P_\infty. \end{aligned} \tag{2.17}$$

Note that the individual points need not be defined over $\mathbf{Q}$, but the divisor $Z$ is nonetheless rational over $\mathbf{Q}$. Here we have chosen the map $\varphi$ of (2.9) to be multiplication by $x^{-2}$ at $P_\infty$ and to be the identity away from $P_\infty$. In other words, the natural trivialization of $\mathcal{L} = \mathcal{O}_C(4P_\infty)$ on the complement of $P_\infty$ allows us to directly evaluate elements of $V$ or $V'$, viewed as elements of the function field, at the six "finite" points of $Z$; since the values of a $\mathbf{Q}$-rational element at the points $(2 \pm \sqrt{2}, 5 \pm 4\sqrt{2})$ are conjugate elements of the extension $\mathbf{Q}[\sqrt{2}]$, the values at these two points are completely described by a single element of $\mathbf{Q}[\sqrt{2}]$. This is equivalent to noting that these two conjugate points on $C(\overline{k})$ correspond to a single point on the scheme $C$, with residue field $\mathbf{Q}[\sqrt{2}]$.

As for evaluating at the remaining point $P_\infty$ (to third order), we "evaluate" an element $s \in V$ by evaluating the function field element $sx^{-2}$, which is regular at $P_\infty$, to third order at that point. More precisely, we take the first three terms $sx^{-2} = a_0 + a_1 t + a_2 t^2 + O(t^3)$ in the power series expansion of $sx^{-2}$ in terms of a uniformizer $t$ of the discrete valuation at $P_\infty$. (Specifically, we choose $t = x/y$, so that $x = t^{-2} + O(t^4)$ and $y = t^{-3} + O(t^3)$. Also, if we wanted to evaluate an element $s' \in V'$ at $3P_\infty$, we would need to take the third-order expansion of $s'x^{-4}$ in terms of $t$.) Putting all of this together, we see that the algebra $\mathcal{A}$ of "values at $Z$" can be identified with

$$\mathcal{A} \cong \mathbf{Q} \times \mathbf{Q} \times \mathbf{Q} \times \mathbf{Q} \times \mathbf{Q}[u]/(u^2 - 2) \times \mathbf{Q}[t]/(t^3), \tag{2.18}$$

where $u$ corresponds to $\sqrt{2}$, and the "values" of the basis elements of $V$ at $Z$ are

$$\begin{aligned}
&(1,1,1,1,1+0u,0+0t+0t^2)\in\mathcal{A} && \text{corresponding to } T_1\leftrightarrow 1,\\
&(0,-1,2,2,2+u,0+0t+t^2) && \text{corresponding to } T_2\leftrightarrow x,\\
&(1,0,3,-3,5+4u,0+t+0t^2) && \text{corresponding to } T_3\leftrightarrow y,\\
&(0,1,4,4,12+8u,1+0t+0t^2) && \text{corresponding to } T_4\leftrightarrow x^2.
\end{aligned} \tag{2.19}$$

Each element of $\mathcal{A}$ above corresponds to a column of the $9\times 4$ matrix $A_V$; for example, the third column is ${}^t(1,0,3,-3,5,4,0,1,0)$. The matrix $A_V$, along with the identification of $\mathcal{A}$ with $\mathbf{Q}^9$ via (2.18) (especially the polynomial equations $u^2-2=0$ and $t^3=0$), then constitute our description of $C$ in Representation B. Note that we have not bothered to slavishly follow (2.10) in the sense of writing the first four factors of $\mathcal{A}$ as quotients of univariate polynomial rings instead of as $\mathbf{Q}$ (e.g., by having the first four factors be $\mathbf{Q}[w]/(w)$ instead). What we have done instead is to combine ideas from Representation $\mathrm{B}_0$ and Representation B.

## 3. Representing divisors. Algorithms for divisor classes

We now turn to the representation of divisors on $C$. We begin with some notation. Given a divisor $D$ and a point $P\in C(\overline{k})$, we write $v_P(D)$ for the multiplicity of $P$ in $D$; hence $D=\sum_P v_P(D)P$, a finite sum. We write $(s)_{\mathcal{L}}$, or $(s)$ if $\mathcal{L}$ is understood, for the divisor of zeros of a nonzero section $s\in H^0(\mathcal{L})$:

$$(s)=(s)_{\mathcal{L}}=\sum_{P\in C(\overline{k})} v_{\mathcal{L},P}(s)P. \tag{3.1}$$

Here $v_{\mathcal{L},P}(s)$ is the valuation of $s$ at the point $P\in C(\overline{k})$. Note that $(s)$ is an effective divisor, with $\deg(s)=\deg\mathcal{L}=\Delta$. Moreover, the linear equivalence class of $(s)$ is the same as that of the line bundle $\mathcal{L}$, and so is independent of the choice of $s$. Note also that since $s\in V$ is rational over $k$, so is the divisor $(s)$, even though the individual points where $(s)$ vanishes might be defined over an extension of $k$.

**Definition 3.1.** Let $D$ be a $k$-rational effective divisor on $C$.

(1) We define the ($k$-rational) subspaces

$$\begin{aligned}
W_D&=\{s\in V\mid \forall P\in C(\overline{k}), v_{\mathcal{L},P}(s)\geq v_P(D)\}=H^0(\mathcal{L}(-D))\subset V,\\
W'_D&=\{s'\in V'\mid \forall P\in C(\overline{k}), v_{\mathcal{L}^{\otimes 2},P}(s')\geq v_P(D)\}=H^0(\mathcal{L}^{\otimes 2}(-D))\subset V'.
\end{aligned} \tag{3.2}$$

Thus $W_D$ and $W'_D$ consist respectively of those linear or quadratic functions on $C$ that vanish at $D$, counting multiplicity. We allow $D=0$, in which case $W_D=V$, $W'_D=V'$.

(2) Take a subset $S\subset V$ containing at least one nonzero element. We say that $S$ is an *ideal generating set* (abbreviated to IGS) for $D$, or equivalently that $D$ is the *divisor of common zeros* of $S$, if

$$\forall P\in C(\overline{k}),\qquad v_P(D)=\min\{v_P(s)\mid s\in S\}. \tag{3.3}$$

We occasionally abuse terminology and call $S$ an IGS for $W_D$.

Note that the divisor of common zeros of $S$ is the same as that of the $k$-subspace of $V$ spanned by $S$. The terminology IGS comes from the interpretation of a divisor $D$ on (an affine part of) $C$ as an ideal in a Dedekind domain.

Clearly, an IGS for $D$ exists if and only if the line bundle $\mathcal{L}(-D)$ is base point free, in which case $W_D$ itself (or even just a basis for $W_D$) will be an IGS. The divisor $D$ is then uniquely determined by any IGS $S$, as it can be viewed as the GCD of the divisors $\{(s) \mid 0 \neq s \in S\}$. Thus we represent our divisors as follows:

**Definition 3.2.** Assume that $D$ is an effective $k$-rational divisor. By abuse of terminology, we say that $W_D$ is *base point free* if the line bundle $\mathcal{L}(-D)$ is base point free.

(1) If $W_D$ is base point free, then a *full representation* of $D$ is any matrix $A_{W_D}$ whose columns (as in Remark 2.4) are a basis for the subspace $W_D$.
(2) If $W_D$ is base point free, then a *brief representation* of $D$ is any IGS $\{s_1, \ldots, s_h\}$ for $D$, where we store the $s_i \in V$ as column vectors in $k^N$.

In particular, if $W$ is a subspace of $V$ whose divisor of common zeros is $D$, then any basis for $W$ can be viewed as a brief representation of $D$. The following proposition collects some elementary facts that play an important role in our algorithms.

**Proposition 3.3.** *Let $D$ be an effective $k$-rational divisor of degree $d$ (we allow $D = 0$). Recall that $\Delta = \deg \mathcal{L} \geq 2g + 2$.*

(1) *If $d \leq \Delta - 2g$, then $W_D$ is base point free. Furthermore, $\dim W_D = \delta - d$ has codimension $d$ in $V$.*
(2) *If $d \leq 2\Delta - 2g$, then a similar statement holds for the subspace $W'_D \subset V'$.*
(3) *Take a nonzero $s \in V$ with $(s)_{\mathcal{L}} = E$. Then the simple multiplication $s \cdot W_D$ is*

$$s \cdot W_D = W'_{D+E}. \tag{3.4}$$

*If, furthermore, $d \leq \Delta - 2g$, then both $W_D$ and $W'_{D+E}$ are base point free.*
(4) *Let $S = \{s_1, \ldots, s_h\}$ be an IGS for $D$. Let $E$ be an effective $k$-rational divisor, preferably but not necessarily such that $W'_{D+E}$ is base point free. Then the division $W'_{D+E} \div S$ is*

$$W'_{D+E} \div S = W_E. \tag{3.5}$$

*Proof.* This follows from easy considerations about valuations and the Riemann-Roch theorem; the main ideas are present in [KM04a]. Incidentally, one can also define $W'_F \div S$ for arbitrary divisors $F$; the result is then $W_{F \setminus D}$, in the sense of Proposition/Algorithm 3.9 of [KM04a]. □

Our next goal is to explain that, with good probability, a random selection of relatively few elements of a base point free space $W_D$ is an IGS for $D$. Moreover, it is easy to test whether any given subset of $W_D$ is an IGS in the setting of our application. This enables us to convert easily between the full and brief representations of $D$. We first clarify what we mean by a random selection of elements of $W_D$, and then state our result precisely.

**Definition 3.4.** Let $\Sigma \subset k$ be a finite subset, and let $|\Sigma|$ be its cardinality. (If $k$ is itself finite, we usually take $\Sigma = k$.) Let $W \subset V$ be a subspace, and choose once and for all a basis $\{w_1, \ldots, w_r\}$ for $W$. We define a $\Sigma$-*random element* $t \in W$ to be an element of the form

$$t = c_1 w_1 + \cdots + c_r w_r, \qquad c_1, \ldots, c_r \in \Sigma, \tag{3.6}$$

where the $c_i$ are chosen independently and randomly with respect to the uniform probability distribution on $\Sigma$. Our notation does not indicate the dependence on the choice of basis $\{w_1, \ldots, w_r\}$, even though this affects the distribution, because the final results on random selection of an IGS are independent of this choice of basis. Note that choosing a $\Sigma$-random element $t$ requires $O(r \log |\Sigma|)$ random bits to produce $c_1, \ldots, c_r$, followed by $O(rg) = O(g^2)$ field operations in $k$ for the linear combination. We will mainly consider sets $\Sigma$ that are not too large: $|\Sigma| = O(g)$; it is also reasonable to take $|\Sigma| = O(1)$, which is the case if $k$ is a finite field.

**Theorem 3.5.** *Let $D$ be an effective $k$-rational divisor with $d = \deg D \leq \Delta - 2g$.*

(1) *Take a finite set $\Sigma \subset k$ as above. Define*

$$h = 1 + \lceil \log 2(\Delta - d) / \log |\Sigma| \rceil \,. \tag{3.7}$$

*Take any nonzero $s_1 \in W_D$, and choose, $\Sigma$-randomly and independently, $h-1$ elements $s_2, \ldots, s_h \in W_D$. Then with probability greater than or equal to $1/2$, the set $\{s_1, \ldots, s_h\}$ is an IGS for $D$.*

(2) *Independently of part 1, assume that $2g - 1 \leq d \leq \Delta$. Let $h$ be any integer, and take elements $s_1, \ldots, s_h \in W_D$. Then $\{s_1, \ldots, s_h\}$ is an IGS for $D$ if and only if the sum of products $s_1 \cdot V + \cdots + s_h \cdot V$ satisfies*

$$s_1 \cdot V + \cdots + s_h \cdot V = W'_D. \tag{3.8}$$

*Proof.* Part 1 follows from Proposition 4.3 below, with $\mathcal{M} = \mathcal{L}(-D)$ and $\eta = 1/2$. Note that the result still holds even if we choose $s_2, \ldots, s_h$ independently and $\Sigma$-randomly from a subspace $W \subset W_D$ whose divisor of common zeros is $D$. Part 2 is Proposition 4.11. □

*Remark* 3.6. Since both $\Delta$ and $d$ are of size $O(g)$, we therefore can obtain a randomly chosen IGS of size $h = O\big(1 + (\log g / \log |\Sigma|)\big) = O(g^\epsilon)$ in fewer than two attempts on average. This is a considerable improvement over using a basis of $W_D$, which would contain $O(g)$ elements, and which would slow down the algorithms of Proposition/Algorithms 2.6 and 2.7. This (along with the insight to use Representation B) is the source of the essential speedup in this article, compared to the algorithms of [KM04a].

Using the framework of Section 2 and this section, we now describe how to convert between the full and brief representations of a divisor $D$. We also introduce the important "flipping" algorithm.

**Proposition/Algorithm 3.7** (Deflation)**.** *Assume given a subspace $W_D \subset V$ that is the full representation of a divisor $D$ with $2g - 1 \leq \deg D \leq \Delta - 2g$. Then there exists a fast probabilistic Las Vegas algorithm that computes a brief representation $\{s_1, \ldots, s_h\}$ of $D$, with $h = O(g^\epsilon)$. We call this a* deflation *of $D$; even though the deflation is not unique, we still write*

$$\mathrm{Defl}(W_D) = \{s_1, \ldots, s_h\}, \qquad \text{where } \{s_1, \ldots, s_h\} \text{ is any IGS for } D. \tag{3.9}$$

*Proof.* We know that $\deg D = \dim V - \dim W_D$. This means that we know the dimension $\dim W'_D = \dim V' - \deg D$, even though we have not yet computed the subspace $W'_D$. We now run the following algorithm:

(1) Compute the value of $h$ from (3.7), and randomly choose $s_1, \ldots, s_h \in W_D$ as in Theorem 3.5 above.

(2) Form the sum of products $W' = s_1 \cdot V + \cdots + s_h \cdot V$ by our fast algorithm. If $\dim W' \neq \dim W'_D$, then our choice of $\{s_1, \ldots, s_h\}$ was not an IGS, so return to step 1. Once the $\dim W' = \dim W'_D$, stop and output the $\{s_i\}$.

The complexity of step 1 (including generating the random bits and forming each $s_i$) is $O(g^2 h) = O(g^{2+\epsilon})$, which can be brought down slightly if one views producing $\{s_2, \ldots, s_h\}$ as a matrix multiplication of $A_{W_D}$ by a random matrix with entries in $\Sigma$. As for step 2, we have $W' \subset W'_D$, so checking the criterion of (3.8) amounts to comparing dimensions. Our choice of the $\{s_i\}$ passes this test with probability at least $1/2$, so the expected number of times that we go through the loop is at most 2. □

Converting back from a brief to a full representation of a divisor, which we call "inflation," requires an IGS for $V$. This should be computed once and for all as part of our precomputations when we store the representation of $C$ and $\mu$ for our algorithms. The rest of our algorithms do not use inflation, but we include it for completeness. As for the IGS for $V$, we do not need it to implement the group operations on divisor classes on $C$, but we do need to have it available for the "membership test" of Section 4, which tests whether a given subspace $W \subset V$ is equal to some $W_D$.

**Lemma/Algorithm 3.8** (IGS for $V$)**.** *There exists a polynomial-complexity, but not "fast", Las Vegas algorithm that can be done exactly once as a precomputation to produce an IGS for $V$. We shall call the* (*nonunique*) *result* $\mathrm{Defl}(V)$.

*Proof.* As we wish to produce an IGS for the empty divisor $D = 0$, we cannot use part 2 of Theorem 3.5 here. We need to go beyond the linear and quadratic spaces $V$ and $V'$ to a "cubic" space $V'' = H^0(\mathcal{L}^{\otimes 3})$. Write the product of $s \in V$ and $t' \in V'$ as $s * t' \in V''$; then the condition for $\{s_1, \ldots, s_h\} \subset V$ to be an IGS for $V$ is

$$s_1 * V' + \cdots + s_h * V' = V''. \tag{3.10}$$

There is no problem in choosing the $\{s_i\}$ from $V$ that have a probability of at least $1/2$ of being an IGS for $V$. Carrying out the modified sum of products in (3.10), however, needs a knowledge of the space $V''$ and of the higher multiplication map $* : V \times V' \to V''$; the problem is to produce this data, after which checking (3.10) is easy. (The data giving $V''$ and $*$ can incidentally be discarded once we find an IGS for $V$.) To find this data, we can use Representation A by Remark 2.3. Then, as in Proposition 2.1, we let $\{T_1, \ldots T_\delta\}$ be a basis for $V$, and work with the polynomial algebra $k[T_1, \ldots, T_\delta]$. The kernel of $\mu$ allows us to find generators of the ideal $I_C$, and we can identify $V$, $V'$, and $V''$ respectively as the portions of the graded algebra $k[T_1, \ldots, T_\delta]/I_C$ in degrees 1, 2, and 3, with the obvious multiplications. Thus finding $V''$ and $*$ can be done by Gröbner bases; the computations involve only linear algebra in the spaces of polynomials in $k[T_1, \ldots, T_\delta]$ of degree at most 3, whose dimension is $O(g^3)$. Thus the computation can be done with a complexity that is polynomial in $g$. □

**Proposition/Algorithm 3.9** (Inflation)**.** *Given a precomputed IGS for $V$, assume we are given a brief representation $\{s_1, \ldots, s_h\}$ of a divisor $D$, with $h = O(g^\epsilon)$. Assume that we know that* $\deg D \geq 2g - 1$. *Then there exists a (deterministic) fast algorithm to find the full representation $W_D$, which we call the* inflation

$$\mathrm{Infl}(\{s_1, \ldots, s_h\}) = W_D, \qquad \{s_1, \ldots, s_h\} \text{ is an IGS for } D. \tag{3.11}$$

*Proof.* The obvious algorithm is:

(1) Calculate the sum of products $W'_D = s_1 \cdot V + \cdots + s_h \cdot V$.
(2) Use the previously computed IGS, $\mathrm{Defl}(V)$, to find $W_D = W'_D \div \mathrm{Defl}(V)$.

□

The next Proposition/Algorithm is fundamental for our algorithms on divisors and divisor classes. Given $D$, it allows us to find a complementary (effective) divisor $\tilde{D}$ such that $D + \tilde{D}$ is in the linear equivalence class of $\mathcal{L}$.

**Proposition/Algorithm 3.10** (Flipping)**.** *Assume given $W_D$, where $2g - 1 \leq \deg D \leq \Delta - 2g$. Take a nonzero $s \in W_D$, and write the divisor of $s$ as $(s)_{\mathcal{L}} = D + \tilde{D}$. Then there exists a fast Las Vegas algorithm to compute the* flip*, $W_{\tilde{D}}$, of our divisor:*

$$\mathrm{Flip}(W_D, s) = W_{\tilde{D}}. \tag{3.12}$$

*Proof.* Compute $W_{\tilde{D}} = (s \cdot V) \div \mathrm{Defl}(W_D)$. This works because $s \cdot V = W'_{D+\tilde{D}}$. □

*Remark* 3.11. We will write $W_{\tilde{D}} = \mathrm{Flip}(W_D)$, without specifying $s$, if the precise choice of $s$ does not matter.

We can now describe the basic setup for implementing group operations on the Jacobian, or more precisely on the classes of $k$-rational divisors. We will describe our algorithms in the context of the "large model" of [KM04a], as well as a slight variant. It is possible to generalize our ideas to the "medium" and "small" models described in that article, but the large model is sufficient to demonstrate the asymptotic speedup of our new algorithms.

**Definition 3.12.** The *large model* of the curve $C$ is defined as follows. We implicitly assume that $g \geq 2$, although everything works (possibly with some increase in degrees of divisors) for $g \leq 1$.

(1) We choose a degree $d \geq 2g$, with $d = O(g)$ nonetheless, and we fix once and for all an effective $k$-rational divisor $D_0$ with $\deg D_0 = d$.
(2) We define our basic line bundle by $\mathcal{L} = \mathcal{O}_C(3D_0)$, and represent the spaces $V$ and $V'$ as well as the multiplication map $\mu$ using either Representation A or Representation B. Note that $\Delta = 3d$.
(3) Given an effective $k$-rational divisor $D$, we say that $D$ is *small* if $\deg D = d$, and *large* if $\deg D = 2d$.
(4) If $D$ is a small divisor, then let $x_D$ be the linear equivalence class of $D - D_0$ in the Jacobian of $C$. Then we represent the element $x_D$ of the divisor by the space $W_D$. Similarly, if $D$ is a large divisor, then define $x_D$ to be the linear equivalence class of $D - 2D_0$, and let the space $W_D$ represent $x_D$.
(5) We calculate and store ahead of time the spaces $W_{D_0}$ and $W_{2D_0}$, as well as an IGS for each space, and a specific $s_0$, unique up to a nonzero factor in $k$, such that $(s_0)_{\mathcal{L}} = 3D_0$. (Thus $s_0$ corresponds to the element $1 \in k(C)$, viewed as an element of $H^0\big(\mathcal{O}_C(3D_0)\big)$.)
(6) If we need to perform the membership test of Proposition/Algorithm 4.12, or inflation as in Proposition/Algorithm 3.9, then compute and store ahead of time an IGS $\mathrm{Defl}(V)$ for $V$ as mentioned above.

*Remark* 3.13. Some assorted remarks:

(1) If the divisor $D$ is small, then $W_D$ (respectively, $W'_D$) has codimension $d$ in $V$ (respectively, in $V'$). If $D$ is large, then the codimension is $2d$. Moreover, if $D$ is small, then its complementary divisor $\tilde{D} = \mathrm{Flip}(D)$ is large, and vice-versa. We see that $D$ and $\tilde{D}$ represent inverse points on the Jacobian, since $D + \tilde{D}$ is linearly equivalent to $3D_0$.
(2) We do not specifically need the spaces $W_{D_0}$ and $W_{2D_0}$. We can use instead spaces $W_{E_0}$ and $W_{E_1}$, where the divisor $E_0$ is linearly equivalent to $D_0$, and $E_1 = \mathrm{Flip}(W_{E_0}, s_0)$ for some nonzero choice of $s_0 \in W_{E_0}$. (It follows that $E_1$ is linearly equivalent to $2D_0$.)
(3) When choosing the divisor $D_0$ and the degree $d$, it is best to make $d$ as small as possible, i.e., $d = 2g$ or perhaps $d = 2g+2$ (which is useful in some contexts). It may, however, be difficult in practice to find effective divisors of a specific degree that are rational over the base field $k$, especially if $k$ is a number field (unless the curve $C$ comes equipped with a known rational point).
(4) Assume that we start with a different representation of $C$ before our precomputation (e.g., as an equation for a singular plane curve birational to $C$). We should also extend the precomputations of Remark 2.2 to compute some spaces $W_D$, for divisors $D$ that are supplied to us along with $C$ (e.g., as formal sums of points on the plane curve), and with which we wish to later do computations in the Jacobian of $C$.
(5) A side note: the divisor $D_1$ in the definition of Representation $\mathrm{B}_0$ and Remark 2.2 is $D_1 = 3D_0$.

We postpone until Section 4 a discussion of how to quickly test whether a given subspace $W \subset V$, having the correct dimension, actually is of the form $W_D$ for a small or large $D$ — that membership test requires slightly different techniques from the other algorithms, which in any case will be used much more often. Instead, we begin with a test for equality on the Jacobian. Observe in this and our later algorithms that we always perform a division by a deflation of a subspace, i.e., using a small IGS instead of the entire subspace representing a divisor.

**Proposition/Algorithm 3.14** (Equality of divisor classes)**.** *Assume given two spaces $W_D$ and $W_E$, corresponding to divisors $D$ and $E$ that are either both small or both large. The the following is a fast Las Vegas algorithm to test whether $D$ and $E$ are linearly equivalent, i.e., whether $x_D = x_E$ on the Jacobian of $C$:*

(1) *Take any nonzero $s \in W_D$ and calculate $W = (s \cdot W_E) \div \mathrm{Defl}(W_D)$.*
(2) *Then $D$ and $E$ are linearly equivalent if and only if the space $W$ is nonzero.*

*Proof.* This is Theorem/Algorithm 4.1 of [KM04a]. In brief, write $(s)_{\mathcal{L}} = D + \tilde{D}$, with $D + \tilde{D}$ linearly equivalent to $3D_0$. Then $s \cdot W_E = W'_{D+\tilde{D}+E}$, so we obtain $W = W_{\tilde{D}+E}$ upon division. Since $\deg(\tilde{D} + E) = 3d = \Delta$, the space $W_{\tilde{D}+E}$ is nonzero precisely when $\tilde{D} + E$ is linearly equivalent to $3D_0$, which is equivalent to $D$ and $E$ being linearly equivalent. Note that $\deg(\tilde{D} + E)$ is larger than our usual degree bounds; our computation of the space $W_{\tilde{D}+E}$ is nonetheless correct, as explained in [KM04a]. □

All group operations on the Jacobian reduce to a single operation, "addflip":

**Definition 3.15.** Given two elements $x, y$ in the Jacobian of $C$ (actually, in any abelian group that is written additively), we define their *addflip* to be

$$\text{Addflip}(x, y) = -(x + y). \tag{3.13}$$

Note that given this operation, it is of course immediate to compute inverses, via $-x = \text{Addflip}(x, 0)$, and hence to compute sums, via $x + y = -\text{Addflip}(x, y)$.

In the original large model from [KM04a], we represented an element of the Jacobian using only $W_D$ for a small divisor $D$. In that context, we can implement the addflip as follows.

**Proposition/Algorithm 3.16** (Addflip of small divisors)**.** *Assume given two subspaces $W_D$ and $W_E$, representing small divisors $D$ and $E$. Let $x_D, x_E$ be the corresponding elements of the Jacobian of $C$. Then the following is a fast Las Vegas algorithm to compute a space $W_F$, for a suitable small divisor $F$, such that $x_F = \text{Addflip}(x_D, x_E)$:*

1. *Choose a nonzero $s \in W_D$, and compute $W_{\tilde{D}} = \text{Flip}(W_D, s)$. (Note that $\tilde{D}$ is a large divisor.)*
2. *Compute $W_{D+E} = (s \cdot W_E) \div \text{Defl}(W_{\tilde{D}})$. (Note that $D + E$ is a large divisor.)*
3. *Flip the result to obtain $W_F = \text{Flip}(W_{D+E})$.*

*Proof.* This is Proposition/Algorithm 4.3 of [KM04a], using the second method of adding divisors (Theorem/Algorithm 3.13 of that earlier article). As in Proposition/Algorithm 3.14 above, we have $s \cdot W_E = W'_{D+\tilde{D}+E}$, so our computation of $W_{D+E}$ is correct. Step 3 shows that $D + E + F$ is linearly equivalent to $3D_0$, and hence $x_D + x_E + x_F = 0$ on the Jacobian. □

*Remark* 3.17. To evaluate $\text{Addflip}(0, x_E)$, we take $D = D_0$ and $s = s_0$. This allows us to skip step 1, and simplify step 2, since we already know a deflation of the space $W_{\tilde{D}} = W_{2D_0}$.

As a variant, we can represent all elements on the Jacobian using large divisors. The resulting algorithm for addflip is given below. Since $D$ is now large, the space $W_D$ has smaller dimension than in our original large model. This will make some computations faster, especially since we do fewer basic operations in this algorithm than in Proposition/Algorithm 3.16.

**Proposition/Algorithm 3.18** (Addflip of large divisors)**.** *Given two elements $x_D, x_E$ of the Jacobian, represented by $W_D, W_E$ for large divisors $D, E$, we can compute $W_F$ for a large divisor $F$ that represents $x_F = \text{Addflip}(x_D, x_E)$ by the following fast Las Vegas algorithm:*

1. *Compute $W_{\tilde{D}} = \text{Flip}(W_D)$. (Note that $\tilde{D}$ is a small divisor.)*
2. *Choose a nonzero $s \in W_E$, so $(s) = E + \tilde{E}$. Compute $W_{\tilde{D}+\tilde{E}} = (s \cdot W_{\tilde{D}}) \div \text{Defl}(W_E)$.*
3. *Our desired result is $W_F = W_{\tilde{D}+\tilde{E}}$.*

*Proof.* The inverses $-x_D$ and $-x_E$ in the Jacobian are given by the linear equivalence classes of $\tilde{D} - D_0$ and $\tilde{E} - D_0$. Thus the divisor $F = \tilde{D} + \tilde{E}$ represents $-x_D - x_E$. □

*Remark* 3.19. In Propositions/Algorithms 3.16 and 3.18, we perform divisions of the form $W' \div S$, where $S = \{s_1, \ldots, s_h\}$ is a randomly selected deflation of some divisor. G. Frey has observed that it is sufficient to use only step 1 of Proposition/Algorithm 3.7, without immediately verifying whether $S$ is an IGS. Then $W' \div S$ always contains the answer that would have resulted using an IGS. We can check whether $W' \div S$ has the correct dimension, which we know in advance from degrees of divisors. If we get the wrong dimension, we repeat the process.

## 4. Randomly selecting an IGS, with verification. Membership test

In the first part of this section, we are given an effective $k$-rational divisor $D$ for which $W_D$ is base point free, and we let $W \subset W_D$ be a subspace whose divisor of common zeros is $D$ (in most applications, $W = W_D$). We wish to study the probability that a suitable random selection of $s_1, \ldots, s_h \in W$ is an IGS for $D$. In order to clarify what is going on, we shall work with the line bundle $\mathcal{M} = \mathcal{L}(-D)$. Then we can view $W$ as a base point free subspace of $H^0(\mathcal{M})$—more precisely, as a base point free linear series of the line bundle $\mathcal{M}$. Hence we wish to determine the probability that there is no point common to all the divisors $(s_1)_{\mathcal{M}}, \ldots, (s_h)_{\mathcal{M}}$.

**Lemma 4.1.** *Let $\mathcal{M}$ be a base point free line bundle on $C$. Let $W \subset H^0(\mathcal{M})$ be a base point free subspace. Fix a nonzero element $s_1 \in W$.*

(1) *There exist proper subspaces $H_1, \ldots, H_\ell \subsetneq W$, with $\ell \leq \deg \mathcal{M}$, with the following property:*

$$\{s_2 \in W \mid \{s_1, s_2\} \text{ is NOT an IGS for } H^0(\mathcal{M})\} = H_1 \cup \ldots \cup H_\ell. \tag{4.1}$$

(2) *More generally, let $h \geq 2$, and view a selection of $s_2, \ldots, s_h \in W$ as a tuple $(s_2, \ldots, s_h)$ in the vector space $W^{h-1}$. Then, with the same $\{H_i\}$, we have*

$$\begin{aligned} &\{(s_2, \ldots, s_h) \in W^{h-1} \mid \{s_1, \ldots, s_h\} \text{ is NOT an IGS for } H^0(\mathcal{M})\} \\ &\qquad = (H_1)^{h-1} \cup \ldots \cup (H_\ell)^{h-1}. \end{aligned} \tag{4.2}$$

*Proof.* Let $P_1, \ldots, P_\ell \in C(\overline{k})$ be the *distinct* points where $s$ vanishes. Thus $\ell \leq \deg \mathcal{M}$. Define $H_i$ to be the $k$-rational subspace $\{t \in W | v_{\mathcal{M},P_i}(t) \geq 1\}$ of sections vanishing at $P_i$. Since $W$ is base point free, we have $H_i \subsetneq W$. Then both sides of (4.2) express the fact that all of $s_2, \ldots, s_h$ also vanish at one of the $P_i$. □

The next lemma, a result in linear algebra, is adapted from [BG04].

**Lemma 4.2.** *Let $W$ be a vector space over $k$, with basis $\{w_1, \ldots, w_r\}$. Take a finite subset $\Sigma \subset k$, and consider $\Sigma$-random elements of $W$ in the sense of Definition* 3.4. *Let $H_1, \ldots, H_\ell \subsetneq W$ be proper subspaces.*

(1) *For a $\Sigma$-random element $t \in W$,*

$$\Pr(t \in H_1 \cup \ldots \cup H_\ell) \leq \ell/|\Sigma|. \tag{4.3}$$

(2) *For independent $\Sigma$-random elements $t_1, \ldots, t_j \in W$,*

$$\Pr\Big((t_1, \ldots, t_j) \in (H_1)^j \cup \ldots \cup (H_\ell)^j\Big) \leq \ell/|\Sigma|^j. \tag{4.4}$$

*Proof.* Both statements easily reduce to the case $\ell = 1$, so we assume from now on that we only have one subspace $H = H_1 \subsetneq W$. We can find an $(r-1)$-dimensional

hyperplane $H' \subset W$ containing $H$. Hence there exist constants $a_1, \ldots, a_r \in k$, not all zero, such that

$$t = c_1 w_1 + \cdots + c_r w_r \in H \implies t \in H' \iff a_1 c_1 + \cdots + a_r c_r = 0. \tag{4.5}$$

Without loss of generality, say that $a_1 \neq 0$. Then for every choice of values of $c_2, \ldots, c_r \in \Sigma$, there exists exactly one value of $c_1 \in k$ for which $t \in H'$, hence at most one value of $c_1$ for which $t \in H$; furthermore, it is possible that $c_1 \notin \Sigma$. So at most $|\Sigma|^{r-1}$ choices of tuples $(c_1, \ldots, c_r) \in \Sigma^r$ lead to $t \in H$, whence $\Pr(t \in H) \leq 1/|\Sigma|$. It follows that $\Pr\Big((t_1, \ldots, t_j) \in H^j\Big) \leq 1/|\Sigma|^j$. This proves our result. □

Combining the two lemmas above, we immediately obtain:

**Proposition 4.3.** *Keep the assumptions and notation of Lemmas* 4.1 *and* 4.2 *above. Take* $0 < \eta < 1$*, and define*

$$h = 1 + \lceil (\log \deg \mathcal{M} - \log \eta) / \log |\Sigma| \rceil . \tag{4.6}$$

*For a fixed nonzero* $s_1 \in W$*, let* $s_2, \ldots, s_h \in W$ *be independently chosen* $\Sigma$*-random elements. Then*

$$\Pr\Big(\{s_1, \ldots, s_h\} \text{ is an IGS for } H^0(\mathcal{M})\Big) \geq 1 - \eta. \tag{4.7}$$

*Proof.* Immediate, once we note that $j = h - 1$ in our previous notation, and that $\ell \leq \deg \mathcal{M}$. □

**Corollary 4.4.** *If* $k$ *is infinite, then every base point free subspace* $W$ *contains an IGS with two elements.*

We are now ready for a more precise statement about random sections giving an IGS, when $k$ is a finite field. We thus take $\Sigma = k$; a $\Sigma$-random element of a vector space $W$ is thus a random element of the finite set $W$, chosen using the uniform distribution. We first note two simple facts.

**Lemma 4.5.** *Assume that* $k = \mathbf{F}_q$*. For* $\ell \geq 1$*, let* $N_\ell$ *be the number of degree* $\ell$ *irreducible divisors on* $C$ *(i.e., divisors of the form* $D = P_1 + \cdots + P_\ell$*, where the* $\ell$ *points* $\{P_1, \ldots, P_\ell\}$ *are a single Galois orbit). Then*

$$N_\ell \leq \frac{1}{\ell}(q^\ell + 1 + 2g q^{\ell/2}). \tag{4.8}$$

*Proof.* The $N_\ell$ irreducible divisors give rise to $\ell N_\ell$ distinct $\mathbf{F}_{q^\ell}$-rational points on $C$. However, $\big|C(\mathbf{F}_{q^\ell})\big| \leq q^\ell + 1 + 2g q^{\ell/2}$ by the simplest form of the Weil bounds (see for example Appendix C of [Har77]). □

**Lemma 4.6.** *Assume that* $k = \mathbf{F}_q$*, and that* $\deg \mathcal{M} = T + 2g - 1$ *with* $T \geq 1$*. Choose random* $s_1, \ldots, s_h \in H^0(\mathcal{M})$ *independently with the uniform distribution. Then the probability that the sections have a common zero (i.e., that they are* not *an IGS) is at most*

$$N_1 q^{-h} + N_2 q^{-2h} + \cdots + N_T q^{-Th} + N_{T+1} q^{-Th} + \cdots + N_{T+2g-1} q^{-Th}. \tag{4.9}$$

*Proof.* For each irreducible divisor $D$, the probability that a given section vanishes at $D$ is $\big|H^0(\mathcal{M}(-D))\big| / \big|H^0(\mathcal{M})\big| = q^{-c}$, where $c$ is the codimension of $H^0(\mathcal{M}(-D))$ in $H^0(\mathcal{M})$. Thus the probability that $h$ sections all vanish at $D$ is $q^{-ch}$. Now by Riemann-Roch, we have that $c = \deg D$ when $1 \leq \deg D \leq T$, and $c \geq T$ when $\deg D \geq T$. Moreover, we know that if $\deg D \geq T + 2g$, then $H^0(\mathcal{M}(-D)) = \{0\}$,

so in that case simultaneous vanishing at $D$ can happen only if all the sections are identically zero — but we have already accounted for this situation in considering divisors of smaller degree. Adding up for all irreducible $D$ the probability that the sections simultaneously vanish at $D$ yields the upper bound (4.9). □

*Remark* 4.7. In the above proof, we have not tried to bound "overcounting"; if $D_1$ and $D_2$ are distinct irreducible divisors, then the probability that $\{s_1, \ldots, s_h\}$ simultaneously vanish at $D_1 + D_2$ is counted twice in (4.9). When $T \to \infty$, the events of vanishing at two (or more) divisors $D_1, D_2$ become independent, with probabilities $q^{-\deg D_1}$ and $q^{-\deg D_2}$. So for $T$ large, a good heuristic estimate of the probability that $h$ random sections do not yield an IGS is given by

$$1 - \prod_{\ell=1}^{\infty}(1 - q^{-\ell h})^{N_\ell} = 1 - \frac{1}{Z_C(h)}, \tag{4.10}$$

where $Z_C(s)$ is the zeta function of $C$. This is analogous to a standard elementary statement that the "probability" that two integers $m, n \in \mathbf{Z}$ are relatively prime (i.e., that $\{m, n\}$ is an IGS!) is $\prod_{p \text{ prime}}(1 - p^{-2}) = 1/\zeta(2) = 6/\pi^2$. Now $Z_C(s)$ is a rational function of $q^{-s}$, and its expansion near $q^{-s} = 0$ (i.e., as $s \to \infty$) gives us $1 - 1/Z_C(h) = N_1 q^{-h} + O(q^{-2h})$. Thus if we want this quantity to be less than $\eta$, we can try the heuristic approximation $h = \lceil \log(N_1/\eta)/\log q) \rceil$. Now $N_1 \le q + 1 + 2g\sqrt{q}$, so if we fix $q$ and let $g$ become large, we obtain a value $h \approx \log(2g\sqrt{q}/\eta)/\log(q) = O(1 + \log(g/\eta)/\log q)$, in line with our results.

We can now state and prove our result Proposition 4.8 for finite fields. This result is significant, even though our algorithms rely on the simpler Proposition 4.3, because the value of $h$ in (4.11) does not depend on $T$, once $T$ is comparable to or larger than $g$. Also note that if $g$ or $q$ is large, then the constant 6 in (4.11) can be reduced significantly. However, the result of Proposition 4.8 only works if we randomly select our sections from the entire space $H^0(\mathcal{M})$, and not a subspace $W$.

**Proposition 4.8.** *Assume that $k = \mathbf{F}_q$, and that $\deg \mathcal{M} = T + 2g - 1$ with $g \ge 1$ and $T \ge 2$. Let $0 < \eta < 1$, and define*

$$h = \max\left(1 + \left\lceil \frac{2g-1}{T-1} \right\rceil, 1 + \left\lceil \frac{\log(6g/\eta)}{\log q} \right\rceil\right). \tag{4.11}$$

*Then a uniform random choice of $h$ sections from $H^0(\mathcal{M})$ is an IGS with probability $> 1 - \eta$.*

*Proof.* By Lemmas 4.5 and 4.6, the probability of *not* being an IGS is bounded above by the quantity

$$P = \sum_{\ell=1}^{T} \frac{1}{\ell}(q^\ell + 1 + 2gq^{\ell/2})q^{-\ell h} + \sum_{\ell=T+1}^{T+2g-1} \frac{1}{\ell}(q^\ell + 1 + 2gq^{\ell/2})q^{-Th}. \tag{4.12}$$

We wish to show that $P < \eta$. We use the following elementary estimates that hold for $N \ge M \ge 1$, $q \ge 2$, and $\sigma \ge 1$:

$$\begin{aligned} &\sum_{\ell=1}^{M} \frac{1}{\ell} q^{-\ell\sigma} < q^{-\sigma} + \frac{q^{-2\sigma}}{2(1-q^{-\sigma})} \le 1.5q^{-\sigma}; \qquad \sum_{\ell=M}^{N} \frac{1}{\ell} \le \frac{N-M+1}{M}; \\ &\sum_{\ell=M}^{N} \frac{1}{\ell} q^{\ell} \le \frac{1}{M} \cdot \frac{q^N}{1-q^{-1}} \le \frac{2q^N}{M}; \qquad \sum_{\ell=M}^{N} \frac{1}{\ell} q^{\ell/2} < \frac{3.5q^{N/2}}{M}. \end{aligned} \tag{4.13}$$

(The constant 3.5 is a simple upper bound for $1/(1-q^{-1/2})$ when $q \geq 2$.) From these, we easily estimate that

$$\begin{aligned} P &< 1.5\left[q^{-(h-1)} + q^{-h} + 2gq^{-(h-1/2)}\right] \\ &\quad + \frac{q^{-Th+T+2g-1}}{T+1}\left[2 + (2g-1)q^{-(T+2g-1)} + 7gq^{-(T+2g-1)/2}\right]. \end{aligned} \tag{4.14}$$

(Note that $h-1 \geq 1$.) Equation (4.11) now implies that $q^{1-h} \leq \eta/6g$, and also that $-Th+T+2g-1 \leq 1-h < 0$. Furthermore, since $T+2g-1 \geq 3$, we obtain

$$\begin{aligned} P &< 1.5\left[\frac{\eta}{6g} + \frac{\eta}{6gq} + \frac{2\eta}{6\sqrt{q}}\right] + \frac{\eta}{6(T+1)g}\left[2 + \frac{2g-1}{q^3} + \frac{7g}{q^{1.5}}\right] \\ &\leq \frac{\eta}{6}\left[\frac{1.5}{g} + \frac{1.5}{gq} + \frac{3}{\sqrt{q}} + \frac{(2-q^{-3})g^{-1} + 2q^{-3} + 7q^{-1.5}}{T+1}\right] < \eta, \end{aligned} \tag{4.15}$$

since $q \geq 2$, $g \geq 1$ and $T \geq 2$. This gives the desired result. □

In the second part of this section, we discuss how one can verify whether our random selection of sections is indeed an IGS. The same techniques also give our algorithm for membership testing. We prove both of these results after two preliminary lemmas. We return to considering a line bundle $\mathcal{L}$, of degree $\Delta \geq 2g+2$, and subspaces of the form $W_D \subset V, W'_D \subset V'$ for effective $k$-rational divisors $D$.

**Lemma 4.9.** *Let $D$ be an effective divisor for which $W_D$ is base point free. Then $\deg D \leq \Delta$. Moreover, we have the following relation between $\deg D$ and the codimension of $W_D$ in $V$:*

1. *If $\operatorname{codim} W_D \leq \Delta - 2g$, then $\deg D = \operatorname{codim} W_D$.*
2. *If $\operatorname{codim} W_D \geq \Delta - 2g + 1$, then $\deg D \geq \Delta - 2g + 1$.*

*Proof.* The first statement follows because $D$ is a "factor" of the divisor of any nonzero $s \in W_D$, but $\deg(s)_{\mathcal{L}} = \Delta$. The statements about the codimension are straightforward (extend scalars to $\overline{k}$, start with $D = 0$, and add one point at a time to $D$). □

**Lemma 4.10.** *Assume given nonzero $t_1, t_2 \in V$, and let $D$ be the divisor of common zeros of $t_1, t_2$. Define $W' = t_1 \cdot V + t_2 \cdot V$. Then $W' \subset W'_D$, and the codimension of $W'$ in $V'$ satisfies*

$$\operatorname{codim} W' = \dim H^0\big(\mathcal{O}_C(D)\big) - 1 + g = \deg D + \dim H^1\big(\mathcal{O}_C(D)\big). \tag{4.16}$$

*In particular, if $\deg D \geq 2g-1$, then $W' = W_D$.*

*Proof.* Write $(t_1)_{\mathcal{L}} = D + E_1$ and $(t_2)_{\mathcal{L}} = D + E_2$, where $E_1$ and $E_2$ are disjoint effective divisors. Now $t_1 \cdot V = W'_{D+E_1}$ and $t_2 \cdot V = W'_{D+E_2}$; hence trivially $W' \subset W'_D$. We now use $\operatorname{codim} W' = \operatorname{codim}(t_1 \cdot V) + \operatorname{codim}(t_2 \cdot V) - \operatorname{codim}(t_1 \cdot V \cap t_2 \cdot V)$ to show (4.16). By construction, $t_1 \cdot V \cap t_2 \cdot V = W'_{D+E_1+E_2}$. Now $D + E_1$ and $D + E_2$ are in the linear equivalence class of $\mathcal{L}$, so $\mathcal{L}^{\otimes 2}(-D - E_1 - E_2) \cong \mathcal{O}_C(D)$. Therefore, $\dim W'_{D+E_1+E_2} = \dim H^0\big(\mathcal{O}_C(D)\big)$, and its codimension is $\delta' - \dim H^0\big(\mathcal{O}_C(D)\big) = 2\Delta + 1 - g - \dim H^0\big(\mathcal{O}_C(D)\big)$. On the other hand, both $t_1 \cdot V$ and $t_2 \cdot V$ have codimension $\Delta$ in $V'$. This proves (4.16). As for the last statement, note that the assumption on $\deg D$ implies that $\operatorname{codim} W' = \deg D$. However, we always have $W' \subset W'_D$, and moreover $\operatorname{codim} W'_D = \deg D$ (use Lemma 4.9 to get $\deg D \leq \Delta \leq 2\Delta - 2g$). Thus $W' = W_D$, as desired. □

**Proposition 4.11.** *The criterion of part 2 of Theorem* 3.5 *is correct.*

*Proof.* It is enough to prove the statement after extending scalars to $\overline{k}$ (any infinite field will do). Let $S \subset W_D$ be the subspace spanned by $s_1, \ldots, s_h$. Then the divisor of common zeros of $S$ is $D + F$ for some effective divisor $F$, and $\{s_1, \ldots, s_h\}$ is an IGS for $D$ if and only if $F = 0$. By Corollary 4.4, there exist $t_1, t_2 \in S$ whose divisor of common zeros is also $D + F$. We have

$$t_1 \cdot V + t_2 \cdot V \subset s_1 \cdot V + \cdots + s_h \cdot V \subset W'_{D+F} \subset W'_D. \tag{4.17}$$

Apply the final statement of Lemma 4.10 to the divisor $D + F$, whose degree is at least $2g-1$ by the assumption on $\deg D$; we conclude that $s_1 \cdot V + \ldots s_h \cdot V = W'_{D+F}$, with codimension $\deg D + \deg F$. This yields the desired result. □

**Proposition/Algorithm 4.12** (Membership test)**.** *Given a subspace $W \subset V$, write $c = \operatorname{codim} W$ (in $V$), and assume that $2g \le c \le \deg \mathcal{L} - 2g$. Define $h = 1 + \lceil \log 2\Delta / \log |\Sigma| \rceil$. Let $D$ be the divisor of common zeros of $W$, so $W \subset W_D$. Then the following is a fast algorithm to check if $W = W_D$, under the assumption that we have precomputed an IGS* $\operatorname{Defl}(V)$ *for $V$ as in Lemma/Algorithm* 3.8*:*

(1) *Select $s_1, \ldots, s_h \in W$ in the usual way (take any $s_1 \neq 0$, and choose the rest $\Sigma$-randomly), and calculate*

$$U' = s_1 \cdot V + \cdots + s_h \cdot V. \tag{4.18}$$

*Write $c' = \operatorname{codim} U'$ (in $V'$). If $c' > c$, then go back to step 1. Otherwise, if $c' < c$, then conclude that $W \neq W_D$ and stop. Otherwise (if $c' = c$), continue.*

(2) *Compute $U = U' \div \operatorname{Defl}(V)$. If $U = W$, then conclude that $W = W_D$. Otherwise, conclude that $W \neq W_D$.*

*Proof.* By statement (1) of Lemma 4.9, we have $c \ge \operatorname{codim} W_D = \deg D$. Our choice of $h$ (which is still $O(g^\epsilon)$) implies that $\{s_1, \ldots, s_h\}$ is an IGS for $D$ with probability at least $1/2$, independently of $\deg D$. As in Proposition 4.11, we extend scalars to $\overline{k}$, and write $S = \operatorname{span}\{s_1, \ldots, s_h\}$, with divisor of common zeros $D + F$; we have $F = 0$ at least half of the time. Again, let $t_1, t_2 \in S$ have a divisor of common zeros $D + F$ to obtain the same inclusions as in (4.17).

We now discuss what happens in the two cases $W = W_D$ and $W \neq W_D$:

(1) If $W = W_D$, then $\deg D = c$, and we obtain as in Proposition 4.11 that $c' = \deg(D + F)$; thus $c' - c = \deg F \ge 0$. We therefore repeat the loop in step 1 at most twice on average until we have $F = 0$, at which point we also obtain $U' = W'_D$. It follows that the division in step 2 computes $U = W_D$, so the test correctly concludes that $W = W_D$.

(2) If $W \neq W_D$, then $\deg D < c$. We distinguish four scenarios:
   (a) $\deg(D + F) \le 2g - 1$: let $F'$ be any effective divisor for which $\deg(D + F + F') = 2g - 1$, and note that $\dim H^0\big(\mathcal{O}_C(D + F)\big) \le \dim H^0\big(\mathcal{O}_C(D + F + F')\big) = g$. By Lemma 4.10, we know that $c' \le \operatorname{codim} t_1 \cdot V + t_2 \cdot V \le 2g - 1 < c$. Hence the test correctly concludes that $W \neq W_D$ in step 1.
   (b) $2g - 1 \le \deg(D + F) < c$: in this and in the following scenarios, we have $c' = \deg(D+F)$ and $U' = W'_{D+F}$, as in the previous proposition. So in this particular scenario, $c' < c$, and we conclude that $W \neq W_D$ in step 1.

(c) $\deg(D+F) = c$: this time, $c' = c$, and we move on to step 2, where we compute $U = W_{D+F}$. It follows that $W \not\subset U$, because $D$ is the divisor of common zeros of $W$, whereas $F \neq 0$ (its degree is $c - \deg D$). Thus step 2 correctly concludes that $W \neq W_D$.

(d) $c < \deg(D+F)$: here $c' > c$, so we repeat step 1. This happens less than half of the time, since if $F = 0$, we must have already landed in scenario a or b above. Thus we loop in step 1 at most twice on average.

□

*Remark* 4.13. Our original "slow" algorithm for testing whether $W = W_D$, Theorem/Algorithm 3.14 of [KM04a], was to compute $\mathrm{Flip}(W)$ and to see if the result had the expected dimension. There, the flip was implemented using a division by a basis for $W$, which was an IGS for $D$. Unfortunately, we cannot do the same using a random selection of $h = O(g^\epsilon)$ elements from $W$ as our IGS, because we would not be able to verify quickly whether our random selection actually was an IGS (we do not know $\deg D$ in advance, and it is, moreover, likely that $\deg D < 2g - 1$).

## 5. Converting from Representation A to Representation B

Our goal in this section is to give a brief sketch, under some conditions on $k$ given below, of how we can convert a curve $C$ that is given using Representation A into a description of $C$ using Representation B. This is a precomputation that we only need to do once, so we will be satisfied with an efficient algorithm (as defined below), which is essentially polynomial time, but not necessarily of complexity $O(g^{3+\epsilon})$.

We emphasize, however, that if it is at all possible to find enough points in $C(k)$ so as to use the simpler form Representation $\mathrm{B}_0$, then we should do so, even if we do not bother with fast linear algebra. For example, this should not pose a problem if $k = \mathbf{F}_q$ with $q$ very large compared to $g$, since then $|C(\mathbf{F}_q)|$ is comparable to $q$.

*In this section, we maintain the following two assumptions about our field $k$.* Both of these assumptions hold if $k$ is a finite field or a number field.

(1) The field $k$ is perfect.

(2) There exists an efficient algorithm to compute the primary decomposition (including finding the radical) of a finite-dimensional $k$-algebra $\mathcal{A}$.

Given that $k$ is perfect, the second condition is (nontrivially) equivalent to our being able to efficiently factor polynomials in $k[x]$. Here an *efficient algorithm* means that if $N = \dim \mathcal{A} = O(g)$, then we have a Las Vegas algorithm with an expected complexity that is polynomial in $g$, where we need to measure complexity in terms of *both* field operations and factorizations of degree $O(N)$ polynomials in $k[x]$. As examples of algorithms for primary decomposition and the computation of radicals, we mention Chapter 8 of [BW93], as well as the articles [EG00], [Kem02], and [DGP99], and the articles cited in their bibliographies.

For an extended treatment of the material in this section, including many details omitted here as well as a fairly self-contained algorithm for primary decomposition, the reader is referred to Sections 6 and 7 of [KM04b].

Starting from Representation A, we can as before produce the projective coordinate ring of $C$, as in Proposition 2.1 and Lemma/Algorithm 3.8; this is

$$\bigoplus_{n \geq 0} H^0\big(\mathcal{L}^{\otimes n}\big) \cong k[T_1, \ldots, T_\delta]/I_C. \tag{5.1}$$

We choose the divisor $Z$ for Representation B to be

$$Z = 3(T_1)_{\mathcal{L}} = (T_1^3)_{\mathcal{L}^{\otimes 3}}. \tag{5.2}$$

Note that if we view $\mathcal{L} = \mathcal{O}_C(D_1)$, then we can take $Z = 3D_1$. Here $\deg Z = 3\Delta$, which allows us to faithfully represent elements of $V$ and $V'$ by their "values" at $Z$. The values in question belong to the algebra $\mathcal{A} = H^0(\mathcal{O}_Z)$, which has dimension $3\Delta = O(g)$.

**Proposition 5.1.** *We can efficiently find a description of $\mathcal{A}$ in terms of a basis and a multiplication table for $\mathcal{A} \times \mathcal{A} \to \mathcal{A}$ (similarly to (2.6) and (2.7)). In the process, we also obtain the images of $T_1, \ldots, T_\delta$ as linear combinations of our basis for $\mathcal{A}$, which allows us to identify $V$ with a specific subspace of $\mathcal{A}$.*

*Sketch of proof.* View $Z$ as a zero-dimensional subscheme of the projective space containing $C$. Its projective coordinate ring is then $k[T_1, \ldots, T_\delta]/\big(I_C + (T_1^3)\big)$. However, we need to find the *affine* coordinate ring $\mathcal{A}$ of $Z$. We first deal with an easy case, when $\{T_1, T_2\}$ is an IGS for $V$. (This can be arranged, for example, if $k$ has at least $2\Delta$ elements, since we can then choose $T_2$ randomly with a good chance of getting an IGS, which we can verify as in Lemma/Algorithm 3.8.) In this easy case, the scheme $Z$ lies entirely in the affine open subset of projective space given by $T_2 \neq 0$, so we can take

$$\mathcal{A} = H^0(\mathcal{O}_Z) = k[T_1, \ldots, T_\delta]/\big(I_C + (T_1^3) + (T_2 - 1)\big). \tag{5.3}$$

The images of $T_1, \ldots, T_\delta$ in $\mathcal{A}$ are the obvious ones. We can find a basis and multiplication table for $\mathcal{A}$ using Gröbner bases, or by a more direct approach that uses our linear algebra algorithms on subspaces of $H^0\big(\mathcal{L}^{\otimes n}\big)$ for $n \leq 8$, which more clearly shows that the algorithm is efficient.

As for the more general case, we need to consider all affine open subsets given by $T_j \neq 0$ for $2 \leq j \leq \delta$. (It suffices in fact to consider $2 \leq j \leq h$, where $\{T_1, T_2, \ldots, T_h\}$ is an IGS for $V$.) For each such $j$, we form the quotient ring of (5.3), but with $T_j$ instead of $T_2$. The quotient ring is then $H^0(\mathcal{O}_{Z_j})$, where $Z_j$ is the portion of $Z$ lying in the affine open set $\{T_j \neq 0\}$. It is then possible to put the $\{H^0(\mathcal{O}_{Z_j})\}$ together, while eliminating redundancy from the intersections $Z_i \cap Z_j$, to obtain $\mathcal{A}$ as above. (Roughly speaking, remove $Z_2$ from $Z$ using a division, then remove any part of $Z_3$ from what remains, and so forth, finding the affine algebra of each piece; then $\mathcal{A}$ is the product of these partial affine algebras.) All of this can again be done using only linear algebra on subspaces of $H^0\big(\mathcal{L}^{\otimes n}\big)$ for $n \leq 8$. □

Now that we have represented $\mathcal{A}$ in a form suitable for computation, we use our ability to find primary decompositions to decompose $\mathcal{A}$ into a product of local Artinian $k$-algebras:

$$\mathcal{A} = \mathcal{B}_1 \times \cdots \times \mathcal{B}_r. \tag{5.4}$$

This decomposition corresponds to writing $Z = e_1 Y_1 + \cdots + e_r Y_r$ for distinct irreducible divisors $Y_i$ (cf. Lemma 4.5). Thus the above decomposition expresses the canonical isomorphism

$$H^0(\mathcal{O}_Z) \cong H^0(\mathcal{O}_{e_1 Y_1}) \times \cdots \times H^0(\mathcal{O}_{e_r Y_r}). \tag{5.5}$$

Let $R$ be the affine coordinate ring of any fixed open subset of $C$ that contains $Z$. Then each irreducible divisor $Y_i$ corresponds to a maximal ideal $P_i$ of the

Dedekind domain $R$, and $Z$ corresponds to the ideal $J = P_1^{e_1} \cdots P_r^{e_r}$. Then the above decompositions are just the Chinese Remainder Theorem:

$$R/J \cong R/P_1^{e_1} \times \cdots \times R/P_r^{e_r}. \tag{5.6}$$

We will use the existence of $R$ and the $P_i$ to clarify our exposition, but we point out that we do not compute $R$ at all; all of our calculations occur in the finite-dimensional algebra $\mathcal{A}$ and in certain vector space subquotients such as the $\mathcal{B}_i$.

Specifically, the primary decomposition algorithm gives us an explicit basis for each $\mathcal{B}_i$, viewing $\mathcal{B}_i$ as a $k$-subspace of $\mathcal{A}$. We simultaneously obtain, via the computation of the radical, a basis for the maximal ideal $\mathfrak{p}_i$ of $\mathcal{B}_i$; here the inclusion $\mathfrak{p}_i \subset \mathcal{B}_i$ corresponds to $P_i/P_i^{e_i} \subset R/P_i^{e_i}$. Write $L_i$ for the residue field $\mathcal{B}_i/\mathfrak{p}_i \cong R/P_i$; in terms of $f_i = [L_i : k]$, we have $\dim_k \mathcal{B}_i = e_i f_i$. Using our multiplication table for $\mathcal{A}$, we can easily implement the ring operations in either $\mathcal{B}_i$ or $L_i$. We can also determine any $k$-linear dependencies between the elements of any finite subset $\{\beta_1, \ldots, \beta_\ell\} \subset \mathcal{B}_i$, or between their reductions $\{\overline{\beta}_1, \ldots, \overline{\beta}_\ell\} \subset L_i$.

We now sketch how to find an explicit isomorphism of each $\mathcal{B}_i$ with a $k$-algebra of the form $k[x]/(h_i(x))$, in order to obtain the isomorphism of (2.10). Finding such an isomorphism is equivalent to finding a "primitive element" for the algebra $\mathcal{B}_i$, which as we shall see is possible because $k$ is perfect and because of the relation with the Dedekind domain $R$. For notational convenience, we shall drop the subscript $i$.

**Proposition 5.2.** *Given, as above, $\mathfrak{p} \subset \mathcal{B}$ with $\dim_k \mathcal{B} = ef$, we can efficiently compute an element $\beta \in \mathcal{B}$ whose minimal polynomial $h(x) \in k[x]$ has degree $ef$.*

*Sketch of proof.* We first find a primitive element $\overline{\beta} \in L = \mathcal{B}/\mathfrak{p}$ and its irreducible minimum polynomial $g(x) \in k[x]$, where $\deg g(x) = f = [L : k]$. This is straightforward; for example, we can select random $\overline{\beta}$ (one can show that the probability of selecting a primitive element is good) and, for each candidate $\overline{\beta}$, find its minimal polynomial $g(x)$ by looking for $k$-dependencies between $\{1, \overline{\beta}, \ldots, \overline{\beta}^f\}$. We repeat this process until we find $\overline{\beta}$ for which $\deg g(x) = f$. We now look for a lift $\beta \in \mathcal{B}$ of $\overline{\beta}$ whose minimal polynomial is $h(x) = (g(x))^e$. This is trivial if $e = 1$, as any lift will do. If $e \geq 2$, then we see that it suffices to find a lift $\beta$ for which $g(\beta) \in \mathfrak{p} - \mathfrak{p}^2$ (since, in that case, $g(\beta) \in \mathcal{B}$ comes from an element of $R$ with valuation 1 at the prime $\mathfrak{p}$). Take an arbitrary lift $\beta_0$ of $\overline{\beta}$. Since $g(\overline{\beta}) = 0$, we know that $g(\beta_0) \in \mathfrak{p}$. If in fact $g(\beta_0) \notin \mathfrak{p}^2$, then we can take $\beta = \beta_0$. Otherwise, replace $\beta_0$ by $\beta_0 + \gamma$, where we take any $\gamma \in \mathfrak{p} - \mathfrak{p}^2$. This yields

$$g(\beta_0 + \gamma) = g(\beta_0) + g'(\beta_0)\gamma + O(\gamma^2) \equiv g'(\beta_0)\gamma \pmod{\mathfrak{p}^2}. \tag{5.7}$$

Since the extension $L/k$ is separable, we have $g'(\overline{\beta}) \neq 0$, from which $g'(\beta_0)$ is a unit in $\mathcal{B}$, and we obtain what we want. □

## References


[Abr96] Dan Abramovich, *A linear lower bound on the gonality of modular curves*, Internat. Math. Res. Notices (1996), no. 20, 1005–1011. MR1422373 (98b:11063)

[AHU75] Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman, *The design and analysis of computer algorithms*, Addison-Wesley Publishing Co., Reading, Mass.-London-Amsterdam, 1975. MR0413592 (54:1706)

[And02] Greg W. Anderson, *Abeliants and their application to an elementary construction of Jacobians*, Adv. Math. **172** (2002), no. 2, 169–205. MR1942403 (2004c:14056)

[BCS97] Peter Bürgisser, Michael Clausen, and M. Amin Shokrollahi, *Algebraic complexity theory*, Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], vol. 315, Springer-Verlag, Berlin, 1997. MR1440179 (99c:68002)

[BG04] Joel Brawley and Shuhong Gao, *On density of primitive elements for field extensions*, 2004 preprint, may be electronically downloaded from the web at the URL `http://www.math.clemson.edu/~sgao/pub.html`

[BW93] Thomas Becker and Volker Weispfenning, *Gröbner bases*, Graduate Texts in Mathematics, vol. 141, Springer-Verlag, New York, 1993. MR1213453 (95e:13018)

[Can87] David G. Cantor, *Computing in the Jacobian of a hyperelliptic curve*, Math. Comp. **48** (1987), no. 177, 95–101. MR866101 (88f:11118)

[Cho54] Wei-Liang Chow, *The Jacobian variety of an algebraic curve*, Amer. J. Math. **76** (1954), 453–476. MR0061421 (15:823a)

[CW90] Don Coppersmith and Shmuel Winograd, *Matrix multiplication via arithmetic progressions*, J. Symbolic Comput. **9** (1990), no. 3, 251–280. MR1056627 (91i:68058)

[DGP99] Wolfram Decker, Gert-Martin Greuel, and Gerhard Pfister, *Primary decomposition: algorithms and comparisons*, Algorithmic algebra and number theory (Heidelberg, 1997), Springer, Berlin, 1999, pp. 187–220. MR1672046 (99m:13049)

[EG00] W. Eberly and M. Giesbrecht, *Efficient decomposition of associative algebras over finite fields*, J. Symbolic Comput. **29** (2000), no. 3, 441–458. MR1751390 (2001a:16079)

[GH94] Phillip Griffiths and Joseph Harris, *Principles of algebraic geometry*, Wiley Classics Library (reprint of 1978 edition), John Wiley & Sons Inc., New York, 1994. MR1288523 (95d:14001)

[Har77] Robin Hartshorne, *Algebraic geometry*, Springer-Verlag, New York, 1977, Graduate Texts in Mathematics, No. 52. MR0463157 (57:3116)

[Hes99] Florian Hess, *Zur Divisorenklassengruppenberechnung in globalen Funktionenkörpern*, Ph.D. thesis, Technische Universität Berlin, 1999, may be downloaded from the web at `http://www.math.tu-berlin.de/~kant/publications/diss/diss_FH.ps.gz`

[Hes02] ———, *Computing Riemann-Roch spaces in algebraic function fields and related topics*, J. Symbolic Comput. **33** (2002), no. 4, 425–445. MR1890579 (2003j:14032)

[HI94] Ming-Deh Huang and Doug Ierardi, *Efficient algorithms for the Riemann-Roch problem and for addition in the Jacobian of a curve*, J. Symbolic Comput. **18** (1994), no. 6, 519–539. MR1334660 (96h:14077)

[Kem02] Gregor Kemper, *The calculation of radical ideals in positive characteristic*, J. Symbolic Comput. **34** (2002), no. 3, 229–238. MR1935080 (2003j:13039)

[KM04a] ———, *Linear algebra algorithms for divisors on an algebraic curve*, Math. Comp. **73** (2004), no. 245, 333–357 (electronic), `math.NT/0105182`. MR2034126 (2005a:14081)

[KM04b] Kamal Khuri-Makdisi, *Asymptotically fast group operations on Jacobians of general curves (previous draft, version 2)*, 2004 preprint, may be electronically downloaded from the web at the URL `http://arxiv.org/abs/math.NT/0409209v2`

[Laz89] Robert Lazarsfeld, *A sampling of vector bundle techniques in the study of linear series*, Lectures on Riemann surfaces (Trieste, 1987) (M. Cornalba, X. Gomez-Mont, and A. Verjovsky, eds.), World Sci. Publishing, Teaneck, NJ, 1989, pp. 500–559. MR1082360 (92f:14006)

[PARI] The PARI Group, Bordeaux, *PARI/GP*, may be downloaded from the web at the URL `http://pari.math.u-bordeaux.fr/`

[Ste04] W. Stein, *Modular Forms Database*, `http://modular.math.washington.edu/Tables`

[Vol94] Emil J. Volcheck, *Computing in the Jacobian of a plane algebraic curve*, Algorithmic number theory "ANTS-I" (Ithaca, NY, 1994) (Leonard M. Adleman and Ming-Deh Huang, eds.), Lecture Notes in Comput. Sci., vol. 877, Springer, Berlin, 1994, pp. 221–233. MR1322725 (96a:14033)

Mathematics Department and Center for Advanced Mathematical Sciences, American University of Beirut, Bliss Street, Beirut, Lebanon

*E-mail address*: `kmakdisi@aub.edu.lb`