Advances in Mathematics of Communications
Volume 4, No. 2, 2010, 261–279
doi:10.3934/amc.2010.4.261

EFFICIENT REDUCTION OF LARGE DIVISORS ON HYPERELLIPTIC CURVES

Roberto Avanzi

Faculty of Mathematics, Ruhr-Universität Bochum and
Horst Görtz Institut für IT-Sicherheit
Universitätsstraße 150, D-44780 Bochum, Germany

Michael J. Jacobson, Jr.

Department of Computer Science, University of Calgary
2500 University Drive NW, Calgary, Alberta, Canada T2N 1N4

Renate Scheidler

Department of Mathematics & Statistics, University of Calgary
2500 University Drive NW, Calgary, Alberta, Canada T2N 1N4

(Communicated by Neal Koblitz)

Abstract. We present an algorithm for reducing a divisor on a hyperelliptic curve of arbitrary genus over any finite field. Our method is an adaptation of a procedure for reducing ideals in quadratic number fields due to Jacobson, Sawilla and Williams, and shares common elements with both the Cantor and the NUCOMP algorithms for divisor arithmetic. Our technique is especially suitable for the rapid reduction of a divisor with very large Mumford coefficients, obtained for example through an efficient tupling technique. Results of numerical experiments are presented, showing that our algorithm is superior to the standard reduction algorithm in many cases.

1. Introduction and motivation

An important concern in the implementation of curve based cryptography, as well as computer algebra systems, is the choice of algorithms with respect to performance. Cryptographic systems are often chosen according to their speed at a given security level, and the working mathematician will surely profit from shorter waiting times in front of the computer. A concrete example is divisor reduction, which represents a key ingredient in hyperelliptic curve arithmetic. Since this procedure can be rather time consuming, it is desirable to optimize this process as much as possible.

In Mumford's representation [18], a degree zero divisor D on a hyperelliptic curve of genus g is represented by a pair of polynomials (Q, P), with Q monic of degree t and P usually (but not necessarily) of degree at most $t-1$. Here, t is the number of finite places in the support of D, counted with multiplicities. The polynomial Q is referred to as the *norm* of D, and D is said to be *reduced* if $t \leq g$. In the case of imaginary hyperelliptic curves, each divisor class contains a unique reduced

2000 *Mathematics Subject Classification:* Primary: 11R58, 14H45; Secondary: 14G50.
Key words and phrases: Hyperelliptic curve, divisor, reduction, continued fraction expansion, scalar multiplication.
The second and third author are supported by NSERC of Canada.

representative, and the purpose of reduction is to determine this representative. In the real hyperelliptic curve scenario, there are in general many reduced elements in each class, and reduction generates one of them. This paper presents a procedure for efficiently reducing a divisor whose norm has very large degree. Such a divisor can arise for example in the context of scalar multiplication as employed in many cryptographic and number theoretic applications, as explained below. Our method is an adaptation of the most efficient algorithm for reducing ideals in quadratic number fields, due to Sawilla et al. [21, 12], to the setting of hyperelliptic curves.

The reduction operation plays an important role in divisor arithmetic. For instance, in Cantor's seminal algorithm [2], the addition of two divisor classes is split into two steps: a *composition* of two degree zero divisors to obtain a divisor equivalent to the sum of the two input divisors, followed by the reduction of said composition. Composing two divisors whose norms have respective degrees s and t usually results in a divisor whose norm has degree $s + t$ and requires a quadratic number (in $s+t$) of base field operations. As result, the composition of two reduced divisors generally results in a divisor whose norm has degree $2g$, and this process, combined with subsequent reduction, requires a base field operation count that is quadratic in g. A different approach to divisor class addition was given by Shanks' NUCOMP algorithm [20, 11], which was originally introduced in the context of composing reduced imaginary binary quadratic forms [23]. NUCOMP interleaves reduction with composition, essentially performing reduction on intermediate operands occurring during the addition. This keeps their degrees well below $2g$ and additionally avoids other computational overhead arising in Cantor's technique. The total number of base field operations is still quadratic in g, but in practice, NUCOMP performs significantly faster than Cantor's method [14] in most cases.

Our focus in this paper is somewhat different from the scenario of computing the reduced composition of two divisors. Instead, the main application of the research described here is the efficient reduction of a divisor of very large norm, i.e. whose norm has degree significantly larger than $2g$. In practice, such a divisor is often obtained by composing several — i.e. more than two — not necessarily distinct divisors. To motivate this scenario, we consider an operation on divisors that is an essential ingredient in much of hyperelliptic curve cryptography and number theory: scalar multiplication.

Given a divisor D and an integer n, the *scalar multiplication* of D by n consists of computing (a divisor equivalent to) nD. The most common method to perform this operation is based on a suitable digital expansion of the scalar n. For instance, using a base 2 expansion $n = \sum_{i=0}^{\ell} n_i 2^i$, one can compute the n-fold of D by a Horner scheme, i.e. a *double-and-add* method. The most commonly used binary expansion is the w-NAF [3, 25, 1]; we refer to [8] for additional information on scalar multiplication techniques. On elliptic curves over fields of characteristic three [15], one would adopt a *triple-and-add* method. In the hyperelliptic curve scenario, this corresponds to composing three copies of the same divisor, see [9] for efficient divisor tripling. In general, the composition of p copies of the same divisor on a curve defined over a field of characteristic p is often an inexpensive process. This is because its main ingredient is the computation of the p-th power of a polynomial, which in characteristic p is very fast. One can even employ several bases simultaneously to expand a given scalar and use such an expansion for scalar multiplication [5, 6, 4].

In all these scalar multiplication techniques, one would first employ *tupling*, i.e. scalar multiplication of a given group element by the underlying base(s) and possibly

some of their products and powers. For example, in a double-base representation using 2 and 3 as bases, one would wish to precompute at the minimum the divisors $2D$ and $3D$, and possibly $5D$, $6D$, and other higher multiples of D as well. While this is usually quite straightforward, the resulting divisors will then have to be reduced. This reduction process needs to handle polynomials of very large degree: m-tupling a reduced divisor on a curve of genus g is expected to result in Mumford coefficients of degree as large as mg. This necessitates highly efficient reduction algorithms.

Reduction of a divisor on a hyperelliptic curve is closely linked to the regular continued fraction expansion of the corresponding irrational function in an appropriate field of Laurent series [14]. Such an expansion is expensive to compute. In contrast, the continued fraction expansion of a *rational* function is simply produced by the Euclidean algorithm and is thus efficiently computable. It is therefore desirable to replace the partial quotients in the expansion of the irrational function arising in the classical reduction procedure by those of a known close rational approximation in such a way that the process still leads to the correct reduced divisor.

This fact was first exploited in the context of arithmetic on ideals in quadratic number fields in [22] and of Jacobians of hyperelliptic curves in [2]. It is also the basis for the NUCOMP algorithm mentioned earlier. The idea was again employed by Sawilla et al. [21, 12] in the context of reducing ideals in quadratic number fields. Sawilla's technique is the best available reduction algorithm in this setting and represents the starting point for our divisor reduction procedure. In addition, our method shares similarities with Cantor's and uses ideas akin to those employed in NUCOMP as presented in [14]. However, we note that Cantor only considered imaginary hyperelliptic curves over fields of odd characteristic, whereas our description also includes real models and applies to any finite field.

As in both Sawilla's and Cantor's methods, the approximating rational function is the quotient of the two Mumford polynomials of the divisor to be reduced; note that NUCOMP uses a different rational approximation. The Euclidean algorithm is applied to these polynomials until half way to finding their greatest common divisor, at which point we obtain a divisor that is almost always reduced and is (in the setting of certain real hyperelliptic curves) at most one step away from being reduced; this is analogous to the NUCOMP situation. We discuss how to avoid this potential extra reduction step, and provide formulae for the final reduced divisor that eliminate the need to compute all the intermediate divisors and are computationally more efficient than Cantor's.

This paper is organized as follows. In Sections 2 through 6, we recall the necessary background on hyperelliptic curves and continued fractions, and explain the connection to divisor reduction. We then describe our reduction algorithm plus some variations, and present numerical experiments designed to test their efficiency, in Section 7. An alternative representation of a real hyperelliptic curve that provides some advantages for our reduction algorithm is given in Section 8. Conclusions and ideas for future research are found in Section 9.

2. Overview of hyperelliptic curves

A considerable amount of literature has been devoted to hyperelliptic curves and their cryptographic applications. We therefore only provide an overview of the material required here and refer the reader to [17, 10, 13, 14] for more details. Following the description of [13], we define a *hyperelliptic curve* of *genus* g over a

finite field $\mathbb{F}_q$ (with q any prime power) to be an absolutely irreducible, non-singular affine plane curve of the form

$$C : y^2 + h(x)y = f(x), \tag{2.1}$$

where $f, h \in \mathbb{F}_q[x]$ and h is usually (but need not be) taken to be zero if q is odd. Such a curve can take on the following two forms:

- *Imaginary model:* f is monic, $\deg(f) = 2g + 1$, and $\deg(h) \leq g$;
- *Real model:* $h = 0$ or h is monic and $\deg(h) = g+1$. Moreover, if q is odd, then f is monic and $\deg(f) = 2g + 2$. If q is even, then either $\deg(f) \leq 2g + 1$, or $\deg(f) = 2g+2$ and the leading coefficient of f is of the form $\mathrm{sgn}(f) = u^2 + u$ for some non-zero $u \in \mathbb{F}_q$.

The *coordinate ring* of C is the ring $\mathbb{F}_q[x, y]$; its field of fractions $\mathbb{F}_q(x, y)$ is the *function field* of C. The *hyperelliptic involution* on C takes y to $\overline{y} = -h(x) - y$, and hence extends to the conjugation map on $\mathbb{F}_q(x, y)$ that maps every element $\alpha = a + by \in \mathbb{F}_q(x, y)$ (with $a, b \in \mathbb{F}_q(x)$) to its *conjugate* $\overline{\alpha} = a - bh - by$.

Imaginary hyperelliptic curve models have one (ramified) place at infinity, denoted by ∞, whereas real models have two infinite places, ∞ and $\overline{\infty}$, both of degree one. In the latter case, there are two embeddings of $\mathbb{F}_q(x, y)$ into the field of Laurent series $\mathbb{F}_q\langle x^{-1}\rangle$, given by the valuations v_∞ and $v_{\overline{\infty}}$ corresponding to the two infinite places. Non-zero elements in $\mathbb{F}_q\langle x^{-1}\rangle$ have the form $\alpha = a_n x^n + \cdots + a_0 + a_{-1}x^{-1} + \cdots$ with $n \in \mathbb{Z}$, $a_i \in \mathbb{F}_q$ for $i \leq n$, and $a_n \neq 0$. Write $n = \deg(\alpha)$, $a_n = \mathrm{sgn}(\alpha)$, and $\lfloor \alpha \rfloor = a_n x^n + \cdots + a_1 x + a_0 \in \mathbb{F}_q[x]$. We choose the embedding of $\mathbb{F}_q(x, y)$ into $\mathbb{F}_q\langle x^{-1}\rangle$ with $\deg(y) = -v_\infty(y) = g + 1$; this establishes the notion of degree and sign for non-zero functions in $\mathbb{F}_q(x, y)$.

Every degree zero divisor D on C can be uniquely written in the form

$$D = \begin{cases} D_x - \deg(D_x)\infty & \text{if } C \text{ is imaginary,} \\ D_x - \deg(D_x)\overline{\infty} - \delta(D)(\infty - \overline{\infty}) & \text{if } C \text{ is real,} \end{cases}$$

where D_x is a *finite* divisor, i.e. a divisor on C whose support does not include any of the infinite places. Here, $\delta(D) = -v_\infty(D)$ is the value of D at the place ∞, referred to as the *distance* of D, if C is real. For C imaginary, D is thus uniquely determined by its finite part D_x, whereas for C real, D is uniquely determined by D_x and $\delta(D)$.

The exact definition of a *semi-reduced* divisor can be found in [2]; suffice it to state here that the semi-reduced divisors on C are exactly those divisors D whose finite portion D_x can be represented by a pair of polynomials $Q, P \in \mathbb{F}_q[x]$ with Q dividing $P^2 + hP - f$. Here, Q is unique up to constant factors in $\mathbb{F}_q$ — usually Q is chosen to be monic — and P is unique modulo Q. The pair (Q, P) is generally referred to as the *Mumford representation* of D_x. In literature sources that focus on imaginary hyperelliptic curves, the Mumford representation specifies that $\deg(P) < \deg(Q)$, but we will not impose this restriction here; in fact, in our algorithm, we generally have $\deg(P) = \deg(Q) + 1$. A semi-reduced divisor D is *reduced* if the Mumford representation (Q, P) of D_x satisfies $\deg(Q) \leq g$.

The *Jacobian* $\mathcal{J}$ of C over $\mathbb{F}_q$ is the group of degree zero divisor classes defined over $\mathbb{F}_q$ under linear equivalence. An easy consequence of the Riemann-Roch Theorem is the fact that every divisor class in $\mathcal{J}$ contains a reduced divisor D. Arithmetic in $\mathcal{J}$ can then be performed via these reduced representatives. Real hyperelliptic curve cryptography can also take place in the *(principal) infrastructure* of C, which

is the set of reduced principal divisors D with $0 \leq \delta(D) < R$. Here, R is the order of the divisor class of $\infty - \overline{\infty}$ in $\mathcal{J}$ and is refereed to as the *regulator* of C.

In the context of Jacobian or infrastructure arithmetic, and specifically public key cryptography, one is frequently faced with the following situation: a potentially large semi-reduced divisor E is given; here, "large" refers to the degree of the polynomial Q in the Mumford representation of E_x, i.e. $\deg(Q)$ is larger, and possibly significantly larger, than the genus g of C. The task is to find a reduced divisor D linearly equivalent to E as efficiently as possible. If C is imaginary, then D is unique. If C is real, then usually certain restrictions on $\delta(D)$ relative to $\delta(E)$ are imposed to guarantee uniqueness. One such common example is the condition $0 \leq \delta(E) - \delta(D) \leq g$ which can always be satisfied; in practice, one can even achieve $\delta(D) = \delta(E)$; see [19, 7, 13].

In order to treat the imaginary and real hyperelliptic curve scenarios simultaneously, we will henceforth write every semi-reduced divisor D as $D = (Q, P, (\delta))$ where (Q, P) is the Mumford representation of D_x, $\delta = \delta(D)$ is included in the representation of D if C is real, and δ is not included otherwise.

3. Continued fraction expansions

We review some basic required facts about continued fractions, confining our discussion to rational functions only, rather than the more general scenario of Laurent series considered in [14].

A *continued fraction expansion* is any symbolic expression of the form

$$[q_0, q_1, \ldots, q_n, \alpha_{n+1}] := q_0 + \cfrac{1}{q_1 + \cfrac{1}{\ddots \cfrac{}{q_n + \cfrac{1}{\alpha_{n+1}}}}}.$$

Fix polynomials $q_0, q_1, \ldots q_m, P_0, Q_0 \in \mathbb{F}_q[x]$ with Q_0 non-zero, and define $\alpha_0 = P_0/Q_0$ and $\alpha_{i+1} = (\alpha_i - q_i)^{-1}$ for $0 \leq i \leq m$. Then $\alpha_i \in \mathbb{F}_q(x)$, and $\alpha_0 = [q_0, q_1, \ldots, q_i, \alpha_{i+1}] = [q_0, q_1, \ldots, q_m]$.

Associated with the continued fraction expansion $\alpha_0 = [q_0, q_1, \ldots, q_m]$ are the following two sequences of polynomials in $\mathbb{F}_q[x]$:

$$\begin{array}{lll} A_{-2} = 0\,, & A_{-1} = 1\,, & A_i = q_i A_{i-1} + A_{i-2} \\ B_{-2} = 1\,, & B_{-1} = 0\,, & B_i = q_i B_{i-1} + B_{i-2} \end{array} \qquad (0 \leq i \leq m). \tag{3.1}$$

The fraction A_i/B_i satisfies $A_i/B_i = [q_0, q_1, \ldots q_i]$ for $0 \leq i \leq m$ and is hence known as the *i-th convergent* of α. The name is justified by the inequality

$$\deg\left(\alpha_0 - \frac{A_i}{B_i}\right) \leq -\deg(B_i B_{i+1}) < -2\deg(B_i) \qquad (0 \leq i \leq m-1). \tag{3.2}$$

It will be useful to define two more sequences of polynomials:

$$\begin{array}{lll} C_{-2} = P_0\,, & C_{-1} = Q_0\,, & C_i = -q_i C_{i-1} + C_{i-2} \\ J_{-2} = -1\,, & J_{-1} = 0\,, & J_i = -q_i J_{i-1} + J_{i-2} \end{array} \qquad (0 \leq i \leq m). \tag{3.3}$$

The sequences given in (3.3) and (3.1) are related as follows:

$$(-i)^{i+1}J_i = B_i \qquad (-2 \le i \le m), \tag{3.4}$$

$$(-1)^{i+1}C_i = Q_0A_i + P_0B_i = C_{-1}A_i + C_{-2}B_i \qquad (-2 \le i \le m), \tag{3.5}$$

$$(-1)^{i+1}Q_0 = C_{i-1}J_i - C_iJ_{i-1} \qquad (-1 \le i \le m). \tag{3.6}$$

These identities are easily established using induction. Using (3.4) and (3.5), (3.2) can now be rewritten as

$$\deg(C_{i-1}J_i) \le \deg(Q_0) \quad (-1 \le i \le m). \tag{3.7}$$

Note that in the special case when

$$q_i = \left\lfloor \frac{C_{i-2}}{C_{i-1}} \right\rfloor \qquad (0 \le i \le m), \tag{3.8}$$

the sequences q_i and C_i represent the quotients and remainders, respectively, obtained when applying the Euclidean algorithm to $\alpha_0 = P_0/Q_0$. In this case, $\alpha_0 = [q_0, q_1, \ldots, q_m]$ is known as the *regular* continued fraction expansion of α_0, and we have $q_i = \lfloor \alpha_i \rfloor$ for $0 \le i \le m$. Since $\deg(C_i)$ strictly decreases and $\deg(J_i)$ strictly increases as i increases from 0 to m, (3.6) implies

$$\deg(C_{i-1}J_i) = \deg(Q_0) \text{ if } q_i = \left\lfloor \frac{C_{i-1}}{C_{i-1}} \right\rfloor \qquad (0 \le i \le m), \tag{3.9}$$

which is a stronger statement than (3.7).

Note that the length $m+1$ of the regular continued fraction expansion of P_0/Q_0 is usually of approximate order $\deg(Q_0)$. More exactly, since $\deg(C_i)$ decreases by at least one (and usually exactly one) as i increases, we see that $\deg(C_m) \le \deg(C_{-1}) - 1 - m$. Substituting $C_{-1} = Q_0$ and $C_m = \gcd(P_0, Q_0)$ yields $m + 1 \le \deg(Q_0) - \deg(\gcd(P_0, Q_0))$. Since this inequality is usually close to sharp, and $\gcd(P_0, Q_0)$ tends to have small degree, we see that $m + 1 \approx \deg(Q_0)$.

4. Continued fractions and hyperelliptic divisors

The relationship between continued fraction expansions and hyperelliptic divisor arithmetic was described in considerable detail in [14], but mainly in the context of divisor composition with subsequent reduction via the NUCOMP algorithm. Although the techniques used here share similarities with those employed in NUCOMP [14], the results and formulas appearing here are new, and we present a more general framework than [14].

Let C be a hyperelliptic curve over $\mathbb{F}_q$ as given in (2.1), and as before, fix polynomials $q_0, q_1, \ldots q_m, P_0, Q_0 \in \mathbb{F}_q[x]$ with Q_0 non-zero. In addition, we require that Q_0 divides $f + hP_0 - P_0^2$, so (Q_0, P_0) is the Mumford representation of some divisor D_1 on C (of some distance $\delta_1 = \delta(D_1)$ if C is real). Recursively define a sequence of polynomials Q_i, P_i $(1 \le i \le m+1)$ as follows:

$$P_i = h - P_{i-1} + q_{i-1}Q_{i-1}, \quad Q_i = \frac{f + hP_i - P_i^2}{Q_{i-1}}. \tag{4.1}$$

Put $\Psi_1 = 1$, and for $2 \le i \le m+2$,

$$\Psi_i = \prod_{j=1}^{i-1} \psi_j \text{ with } \psi_j = \frac{P_j + y}{Q_{j-1}}. \tag{4.2}$$

Now set $D_i = D_{i-1} + \mathrm{div}(\psi_{i-1}) = D_1 + \mathrm{div}(\Psi_i)$ for $2 \le i \le m+2$, where $\mathrm{div}(\alpha)$ denotes the principal divisor of any non-zero function $\alpha \in \mathbb{F}_q(x,y)$. Then it is easy to verify, using (4.1), that $D_i = (Q_{i-1}, P_{i-1}, (\delta_i))$, and $\delta_i = \delta(D_i) = \delta(D_1) + \deg(\Psi_i)$ if C is real. Now (4.1) easily yields $\psi_i\psi_{i-1} = q_{i-1}\psi_{i-1} + 1$, so (4.2) implies $\Psi_1 = 1$, $\Psi_2 = \psi_1$, and $\Psi_{i+1} = q_{i-1}\Psi_i + \Psi_{i-1}$ for $1 \le i \le m+1$. From (3.3), we thus obtain

$$\Psi_{i+1} = (-1)^i \frac{C_{i-1} + J_{i-1}(h+y)}{Q_0} \quad (0 \le i \le m+1). \tag{4.3}$$

Our aim is to obtain closed form formulae for Q_i and P_i in terms of the sequences C_i and J_i only, using (4.3). These formulae avoid the need to compute the intermediate Mumford basis coefficients Q_j, P_j $(1 \le i \le i-1)$. A similar idea was employed in the NUCOMP algorithm — see expression (8.1) of [14] — except that the sequence $(-1)^i A_i$, with A_i as given in (3.1), was used in place of J_i. In fact, there are many similar such expressions for the Mumford coefficients Q_i and P_i in terms of the remainder sequence C_i and one other related linear sequence. In our context, we chose the formulation in terms of C_i and J_i because in view of (4.3), it allows for the most straightforward treatment and minimizes notation as well as computational effort.

Proposition 4.1. *Let $Q_0, P_0 \in \mathbb{F}_q[x]$ with Q_0 dividing $f + hP_0 - P_0^2$, and let Q_i, P_i be defined by (4.1), and C_i, J_i by (3.3). Then*

$$Q_i = \frac{(-1)^i}{Q_0}(C_{i-1}^2 + C_{i-1}J_{i-1}h - J_{i-1}^2 f),$$
$$P_i = \frac{(-1)^i}{Q_0}(C_{i-2}C_{i-1} + C_{i-2}J_{i-1}h - J_{i-2}J_{i-1}f),$$

for $0 \le i \le m+1$.

Proof. From (4.1), it is easy to verify that $\psi_i\overline{\psi}_i = -Q_i/Q_{i-1}$, and hence $\Psi_{i+1}\overline{\Psi}_{i+1} = (-1)^i Q_i/Q_0$. Thus, by (4.3),

$$\begin{aligned} Q_i &= (-1)^i Q_0 \Psi_{i+1}\overline{\Psi}_{i+1} \\ &= (-1)^i Q_0 \frac{C_{i-1} + J_{i-1}(h+y)}{Q_0} \frac{C_{i-1} - J_{i-1}y}{Q_0} \\ &= \frac{(-1)^i}{Q_0}(C_{i-1}^2 + C_{i-1}J_{i-1}h - J_{i-1}^2 f). \end{aligned}$$

Furthermore, again by (4.3),

$$\begin{aligned} \frac{P_i + y}{Q_{i-1}} = \psi_i = \frac{\Psi_{i+1}}{\Psi_i} &= \frac{\Psi_{i+1}\overline{\Psi}_i}{\Psi_i\overline{\Psi}_i} \\ &= -\frac{(-1)^i \dfrac{C_{i-1} + J_{i-1}(h+y)}{Q_0}(-1)^{i-1}\dfrac{C_{i-2} - J_{i-2}y}{Q_0}}{(-1)^{i-1}\dfrac{Q_{i-1}}{Q_0}}. \end{aligned}$$

Now 1 and y form an $\mathbb{F}_q(x)$-basis of $\mathbb{F}_q(x,y)/\mathbb{F}_q(x)$. The formula for P_i can now be read off by comparing the basis coefficient of 1 on both sides. □

The polynomials P_i and Q_i are related as follows:

Lemma 4.2. *Let $Q_0, P_0 \in \mathbb{F}_q[x]$ with Q_0 dividing $f + hP_0 - P_0^2$, and let Q_i, P_i be defined by (4.1), and C_i, J_i by (3.3). Then*

$$P_iC_{i-1} - Q_iC_{i-2} = J_{i-1}f,$$
$$P_iJ_{i-1} + Q_iJ_{i-2} = C_{i-1} + J_{i-1}h,$$

for $0 \le i \le m+1$.

Proof. Multiply the expressions for P_{i+1} and Q_{i+1} in Proposition 4.1 by C_{i-1} and C_{i-2}, respectively, to obtain

$$(-1)^iQ_0P_iC_{i-1} = C_{i-2}C_{i-1}^2 + C_{i-2}C_{i-1}J_{i-1}h - C_{i-1}J_{i-2}J_{i-1}f,$$
$$(-1)^iQ_0Q_iC_{i-2} = C_{i-2}C_{i-1}^2 + C_{i-2}C_{i-1}J_{i-1}h - C_{i-2}J_{i-1}^2f.$$

Subtracting these two equalities yields

$$(-1)^iQ_0(P_iC_{i-1} - Q_iC_{i-2}) = J_{i-1}f(C_{i-2}J_{i-1} - C_{i-1}J_{i-2}).$$

Our claim now follows from (3.6). The second identity can be obtained similarly. □

5. Divisor reduction via continued fractions

Consider now the case where $P_0/Q_0 = [q_0, q_1, \ldots, q_m]$ is the regular continued fraction expansion of P_0/Q_0. Conventional divisor reduction as described for example in [14] applies uniformly to both real and hyperelliptic curves (although in practice, one would not use the technique for imaginary curves).

The idea is to compute a reduced (or almost reduced) divisor $D_{i+1} = (Q_i, P_i, (\delta_{i+1}))$ linearly equivalent to some starting divisor $D_1 = (Q_0, P_0), (\delta_1))$ by repeatedly applying (4.1) until a polynomial Q_i of degree at most g (or possibly $g+1$) is reached. Each iteration of (4.1) reduces the degree of Q_i by at least 2, except for the last step which might only reduce it by 1. When C is real, the relative distance $\delta_{i+1} - \delta_1$ can easily be obtained alongside as well. In almost all situations, a reduced divisor is obtained; for just one scenario (which only occurs when $\deg(f) = 2g+2$), this procedure might only produce a minimal degree of $g+1$ for Q_i, but a slight change in the expression (4.1) for P_i in the last step yields a reduced divisor. The target divisor is reached after at most $\lceil(\deg(Q_0) - g)/2\rceil$ iterations; this bound is generally sharp, especially for large base fields $\mathbb{F}_q$.

The recursive nature of (4.1) requires the computation of the Mumford basis coefficients of all the intermediate divisors $D_2, D_3, \ldots, D_i$, which is very costly. This can be avoided by computing Q_i and P_i via the much faster linear recurrences (3.3), using the expressions given in Proposition 4.1. However, the termination condition $\deg(Q_i) \le g$ is now useless — one cannot know if this holds at any given point without actually computing Q_i, which is exactly what we wish to avoid. In this section, we replace this condition by a more suitable termination condition involving either one of the sequences J_i or C_i. We also explain how the distance of the reduced target divisor can be found.

The ideas described above — using the partial quotients of the regular continued fraction expansion of a suitable rational function and certain simple linear recurrences to avoid computing intermediate divisors — were already employed in the NUCOMP algorithm. So it is once again not surprising that the techniques utilized in this section are quite similar to those of [14, Section 8]. The main difference is that NUCOMP is based on the continued fraction expansion of a different rational function that was first suggested by Schnorr and Seysen ([22], see Appendix A of [21]). The numerator and denominator of this rational function, when expressed

in lowest terms, arise from the formulae for divisor addition, and their degrees are bounded by the genus g of the curve when the input divisors are reduced. For this reason, when the aim is to find the reduced composition of two reduced divisors, NUCOMP is more efficient than composing the two divisors and subsequently applying the reduction method presented here. As mentioned earlier, our algorithm is instead best suited to the situation when the input divisor D_1 is very large, i.e. the Mumford coefficients Q_0 and P_0 have degree significantly exceeding $2g$.

We continue to let C be a hyperelliptic curve of genus g over $\mathbb{F}_q$ as given in (2.1). Throughout this section, Q_0, P_0 are polynomials in $\mathbb{F}_q[x]$ with Q_0 non-zero and Q_0 dividing $f + hP_0 - P_0^2$. We also assume that $\deg(Q_0) \geq g+1$, as otherwise D_1 is already reduced. Henceforth, we restrict to the case of the regular continued fraction expansion of P_0/Q_0 and make the connection to reduction. That is, $P_0/Q_0 = [q_0, q_1, \ldots, q_m]$ where $q_i = \lfloor C_{i-2}/C_{i-1} \rfloor$ for $0 \leq i \leq m$, with C_i given by (3.3). We begin with some bounds on $\deg(Q_i)$.

Lemma 5.1. *Let Q_i, P_i $(1 \leq i \leq m+1)$ be given by Proposition 4.1, and let $r \geq 0$ be the maximal index such that*

$$\deg(J_r) \leq N \quad \textit{with} \quad N = \left\lfloor \frac{\deg(Q_0) - g}{2} \right\rfloor. \tag{5.1}$$

Then the following hold.

1. $\deg(Q_i) = \deg(C_{i-1}^2/Q_0) \geq g+2$ *for* $0 \leq i \leq r-1$.
2. *If* $\deg(J_r) = N$ *and* $\deg(Q_0) - g$ *is even, then* $\deg(Q_r) \leq g$.
3. *If* $\deg(J_r) < N$, *then* $\deg(Q_r) = \deg(C_{r-1}^2/Q_0) \geq g+2$ *and* $\deg(Q_{r+1}) \leq g$.
4. *If* $\deg(J_r) = N$, $\deg(Q_0) - g$ *is odd, and* $\deg(f) \leq 2g+1$, *then* $\deg(Q_r) = \deg(C_{r-1}^2/Q_0) = g+1$ *and* $\deg(Q_{r+1}) \leq g$.
5. *If* $\deg(J_r) = N$, $\deg(Q_0) - g$ *is odd, and* $\deg(f) = 2g+2$, *then* $\deg(Q_r) = \deg(C_{r-1}^2) = g+1$, $\deg(Q_{r+1}) = \deg(J_{r-1}^2 f/Q_0) = g+1$, *and* $\deg(Q_i) = \deg(J_{i-1}^2 f/Q_0) \geq g+3$ *for* $r+2 \leq i \leq m$.

Proof. We use the formula for Q_i given in Proposition 4.1 for our proof, and bound each of the summands in that expression separately, starting with the middle summand. Since $J_0 = -1$ and $\deg(Q_0) \geq g+1$, we see that $\deg(J_0) \leq N$, so the index $r \geq 0$ as defined in (5.1) exists. We also note that $\deg(Q_0) - g - 1 \leq 2N \leq \deg(Q_0) - g$, where the first inequality is an equality if $\deg(Q_0) - g$ is odd, and the second inequality is an equality if $\deg(Q_0) - g$ is even.

Recall that $\deg(J_i) = \deg(q_i J_{i-1}) \geq \deg(J_{i-1}) + 1$ and $\deg(C_{i-2}) = \deg(q_i C_{i-1}) \geq \deg(C_{i-1}) + 1$ for $1 \leq i \leq m$. Hence

$$\deg\left(\frac{C_{i-1}J_{i-1}h}{Q_0}\right) \leq \deg\left(\frac{C_{i-1}J_i h}{Q_0}\right) - 1 = \deg(h) - 1 \leq g$$

for $0 \leq i \leq m$, where the equality above follows from (3.9).

Suppose first that $0 \leq i \leq r-1$. Then by (3.9),

$$\deg\left(\frac{C_{r-2}^2}{Q_0}\right) = \deg\left(\frac{Q_0}{J_{r-1}^2}\right) \geq \deg\left(\frac{Q_0}{J_r^2}\right) + 2 \geq \deg(Q_0) - 2N + 2 \geq g+2,$$

and

$$\deg\left(\frac{J_{r-1}^2 f}{Q_0}\right) \leq \deg\left(\frac{J_r^2 f}{Q_0}\right) - 2 \leq 2N + (2g+2) - \deg(Q_0) - 2 \leq g.$$

Furthermore, $\deg(J_i) \leq \deg(J_{r-1})$ for $0 \leq i \leq r-1$ and and $\deg(C_i) \geq \deg(C_{r-2})$ for $0 \leq i \leq r-2$. Applying the strict triangle equality for degrees to the impression

for Q_i given in Proposition 4.1 yields $\deg(Q_i) = \deg(C_{i-1}^2/Q_0) \geq g+2$ for $0 \leq i \leq r-1$.

Next, consider Q_r and Q_{r+1}. Write $2N = \deg(Q_0) - g - e_1$ where $e_1 = 1$ if $\deg(Q_0) - g$ is odd and $e_1 = 0$ otherwise. Also, write $\deg(J_r) = N - e_2$ and $\deg(f) = 2g + 2 - e_3$ with $e_2, e_3 \geq 0$. Then by (3.9),

$$\deg\left(\frac{C_{r-1}^2}{Q_0}\right) = \deg\left(\frac{Q_0}{J_r^2}\right) = \deg(Q_0) - 2N + 2e_2 = g + e_1 + 2e_2.$$

Thus, $\deg(Q_r) \leq g$ if $e_1 = e_2 = 0$, i.e. if $\deg(Q_0) - g$ is even and $\deg(J_r) = N$, and $\deg(Q_r) = \deg(C_{r-1}^2/Q_0) \geq g+2$ if $e_2 = 1$. Again by (3.9),

$$\deg\left(\frac{C_r^2}{Q_0}\right) = \deg\left(\frac{Q_0}{J_{r+1}^2}\right) \leq \deg(Q_0) - 2(N+1) = g - 2 + e_1 \leq g$$

and

$$\deg\left(\frac{J_r^2 f}{Q_0}\right) = 2(N - e_2) + (2g + 2 - e_3) - \deg(Q_0) = g + 2 - e_1 - 2e_2 - e_3.$$

If $e_2 = 1$, i.e. $\deg(J_r) < N$, then $\deg(Q_{r+1}) \leq g$. If $e_2 = 0$ and $e_1 = 1$, then $\deg(Q_r) = \deg(C_{r-1}^2/Q_0) = g+1$ and $\deg(Q_{r+1}) = \deg(J_r^2 f/Q_0) = g + 1 - e_3$, which is at most g if $\deg(f) \leq 2g+1$, and is equal to $g+1$ if $\deg(f) = 2g+2$. In the latter case, $\deg(Q_i) = \deg(J_{i-1}^2 f/Q_0) \geq \deg(J_r^2 f/Q_0) + 2 \geq g+3$ for $r+2 \leq i \leq m+1$. □

Note that $r \leq \deg(J_r) \leq N$, and these inequalities are usually sharp. So we expect that $r \approx (\deg(Q_0) - g)/2$. As pointed out at the end of Section 3, we also usually expect $m+1 \approx \deg(Q_0)$. So for polynomials Q_0 of large degree, the index r occurs just under halfway into the regular continued fraction expansion of P_0/Q_0.

Using (3.9), and noting that $\deg(Q_0) - N = \lceil (\deg(Q_0) + g)/2 \rceil$, we easily derive the following alternate characterization of the index r of Lemma 5.1:

$$\deg(J_r) \leq \left\lfloor \frac{\deg(Q_0) - g}{2} \right\rfloor < \deg(J_{r+1}) \Leftrightarrow \deg(C_r) < \left\lceil \frac{\deg(Q_0) + g}{2} \right\rceil \leq \deg(C_{r-1}).$$

From Lemma 5.1, we infer the following.

Corollary 5.2. *With the notation of Lemma 5.1, set $D_i = (Q_{i-1}, P_{i-1}, (\delta_i))$. Then the following hold:*

1. *D_i is not reduced for $1 \leq i \leq r$.*
2. *D_{r+1} is reduced if and only if $\deg(J_r) = N$ and $\deg(Q_0) - g$ is even.*
3. *If D_{r+1} is not reduced, then D_{r+2} is reduced unless $\deg(Q_0) - g$ is odd, $\deg(J_r) = N$ and $\deg(f) = 2g+2$, in which case D_i is not reduced for all $1 \leq i \leq m+2$.*

We note the similarity of Lemma 5.1 to Lemma 8.1 and Corollary 8.1 of [14], and that of Corollary 5.2 to Proposition 8.1 of [14].

By Lemma 5.1, $\deg(Q_i)$ decreases by at least 2 whenever i increases, except for the step just before the minimal degree (g or $g+1$) is encountered, which may result in a decrease by one only. The only problem case, when none of the divisors D_i ($1 \leq i \leq m+2$) is reduced, happens when $\deg(f) = 2g+2$, $\deg(Q_0) - g$ is odd, and $\deg(J_r) = N$. Then D_{r+1} (and also D_{r+2}) is as close to being reduced as possible. We will address this last situation shortly and also revisit it in Section 8.

Corollary 5.3. *Let Q_i, P_i $(1 \le i \le m+1)$ be given by Proposition 4.1, and r by (5.1). Then $q_i = \lfloor P_i/Q_i \rfloor$ for $0 \le i \le r-1$. In addition, if $\deg(Q_r) \ge g+1$, then $q_r = \lfloor P_r/Q_r \rfloor$.*

Proof. By Lemma 4.2 and (3.9), we have

$$\frac{P_i}{Q_i} - \frac{C_{i-2}}{C_{i-1}} = \frac{J_{i-1}f}{Q_i C_{i-1}} = \frac{J_{i-1}J_i f}{Q_{i-1}Q_0}. \tag{5.2}$$

For $0 \le i \le r-1$, we have $\deg(J_{i-1}) < N$, $\deg(J_i) \le N$, and $\deg(Q_{i-1}) \ge g+2$. By (5.2),

$$\begin{aligned}\deg\left(\frac{P_i}{Q_i} - \frac{C_{i-2}}{C_{i-1}}\right) &< 2N + (2g+2) - (g+2) - \deg(Q_0) \\ &\le (\deg(Q_0) - g) + g - \deg(Q_0) \;=\; 0,\end{aligned}$$

so $\lfloor P_i/Q_i \rfloor = \lfloor C_{i-2}/C_{i-1} \rfloor = q_i$.

If $\deg(Q_r) \ge g+1$, then by Lemma 5.1, $\deg(J_r) < N$ or $\deg(Q_0) - g$ is odd, in which case $2N < \deg(Q_0) - g$. Either way, $2\deg(J_r) < \deg(Q_0) - g$, so $\deg(J_{r-1}J_r) \le \deg(J_r^2) - 1 < \deg(Q_0) - g - 1$. It follows from (5.2) that

$$\deg\left(\frac{P_r}{Q_r} - \frac{C_{r-2}}{C_{r-1}}\right) < (\deg(Q_0) - g - 1) + (2g+2) - (g+1) - \deg(Q_0) = 0,$$

so $\lfloor P_r/Q_r \rfloor = \lfloor C_{r-2}/C_{r-1} \rfloor = q_r$. □

The result of Corollary 5.3 is analogous to Theorem 7.1 of [14]. It shows that (4.1) with $q_i = \lfloor C_{i-2}/C_{i-1} \rfloor$ $(0 \le i \le m)$ is in fact identical to the conventional reduction method as described in [14]. Therefore, the divisors $D_i = (Q_{i-1}, P_{i-1})$ produced by the algorithm of [14] are identical to those produced by (4.1) and by Proposition 4.1.

In the case when $\deg(f) = 2g+2$, $\deg(Q_0) - g$ odd and $\deg(J_r) = N$, we obtain $\deg(Q_r) = \deg(Q_{r+1}) = g+1$ according to Lemma 5.1. In this case, a modified version of (4.1) needs to be applied to $D_{r+1} = (Q_r, P_r, \delta_{r+1})$ to obtain a reduced divisor as follows. First, precompute an element $s \in \mathbb{F}_q^*$ such that $s^2 = \operatorname{sgn}(f)$ if q is odd and $s^2 + s = \operatorname{sgn}(f)$ if q is even, so $s = \operatorname{sgn}(y)$ or $s = \operatorname{sgn}(-y-h)$. Specifically, $s = \pm 1$ if q is odd and $s = u$ or $u+1$ with $\operatorname{sgn}(f) = u^2 + u$ if q is even. Now set

$$q_r = \left\lfloor \frac{P_r + sx^{g+1}}{Q_r} \right\rfloor,\; P'_{r+1} = h - P_r + q_r Q_r,\; Q'_{r+1} = \frac{f + hP_{r+1} - P_{r+1}^2}{Q_r}. \tag{5.3}$$

These expressions are the analogue of Equation (8.5) of [14]. If both the pairs (Q_r, P_r) and (Q_{r-1}, P_{r-1}) are available, then the full division by Q_r in (5.3) can be avoided by using the recursion

$$\begin{aligned} P_r + sx^{g+1} &= q_r Q_r + R_r,\; \deg(R_r) < \deg(Q_r) \quad \text{(division with remainder)} \\ P'_{r+1} &= h + sx^{g+1} - R_r, \\ Q'_{r+1} &= Q_{r-1} + q_r(P_r - P_{r-1}). \end{aligned} \tag{5.4}$$

Lemma 5.4. *Suppose that $\deg(f) = 2g+2$, $\deg(Q_0) - g$ odd, $\deg(J_r) = N$, and let Q'_{r+1} be given by (5.3). Then $\deg(Q'_{r+1}) \le g$.*

Proof. We have

$$Q'_{r+1} = \frac{(f - shx^{g+1} - s^2x^{2g+2}) + R_r(h - R_r)}{Q_r}. \tag{5.5}$$

By Lemma 5.1, $\deg(Q_r) = g+1$, so $\deg(R_r) \leq \deg(Q_r) - 1 = g$. Therefore, both summands in the numerator of the right hand side of (5.5) have degree at most $2g+1$. Thus, $\deg(Q_r) = g+1$ implies $\deg(Q_{r+1}) \leq g$. □

We conclude this section by determining the distance of the reduced target divisor in the case when C is real.

Theorem 5.5. *Let $D_{i+1} = (Q_i, P_i, \delta_{i+1})$ be a reduced divisor, so either $D_{i+1} = D_{r+1} = (Q_r, P_r, \delta_{r+1})$ with Q_r, P_r given by Proposition 4.1 (with $i = r$), or $D_{i+1} = D'_{r+2} = (Q'_{r+1}, P'_{r+1}, \delta'_{r+2})$ with Q'_{r+1}, P'_{r+1} given by (5.3), as determined by Corollary 5.2. Then*

$$\delta_{i+1} = \delta_1 + \sum_{j=1}^{i-1} \deg(q_j) - \deg(Q_0) + \deg(P_i + y).$$

Proof. Recall that $\delta_{i+1} = \delta_1 + \deg(\Psi_{i+1})$ with Ψ_{i+1} given by (4.2). So we need to compute $\deg(\psi_j)$ for $1 \leq j \leq i$.

We have $\psi_j = \phi_j Q_j / Q_{j-1}$ with $\phi_j = (P_j + y)/Q_j$ for $1 \leq j \leq m+1$. Thus, $\deg(\psi_j) = \deg(\phi_j) - \deg(Q_{j-1}) + \deg(Q_j)$. Now by Corollary 5.3, $\deg(P_j) = \deg(q_j Q_j) \geq \deg(Q_j)+1 \geq g+2 > \deg(y)$, so $\deg(P_j+y) = \deg(P_j)$ for $0 \leq j \leq i-1$. It follows that $\deg(\phi_j) = \deg(q_j)$ for $0 \leq i \leq j-1$, and hence

$$\begin{aligned}
\delta_{i+1} &= \delta_1 + \sum_{j=1}^{i-1} \deg(\psi_j) + \deg(\psi_i) \\
&= \delta_1 + \sum_{j=1}^{i-1} \big(\deg(q_j) - \deg(Q_{j-1}) + \deg(Q_j)\big) + \deg(P_i + y) - \deg(Q_{i-1}) \\
&= \delta_1 + \sum_{j=1}^{i-1} \deg(q_j) - \deg(Q_0) + \deg(P_i + y).
\end{aligned}$$

□

Theorem 5.5 is analogous to the discussion about distances on p. 228 of [14]. We point out that we generally expect $\deg(P_i + y) = g+1$ in the above identity. Furthermore, the sum above can be computed concurrently with the recurrence for C_j by initializing $d_1 = \delta_1 - \deg(Q_0)$, computing $d_{j+1} = d_j + \deg(q_j)$ alongside q_j and C_j for $1 \leq j \leq i$, and finally setting $\delta_{i+1} = d_i + \deg(P_i + y)$.

We are now able to compute the Mumford representation of a reduced divisor D_{i+1} that is linearly equivalent to D_1, and the distance of D_{i+1} if C is real, directly from D_1. No intermediate divisors $D_2, D_3, \ldots$ are computed except in one case, where only D_i needs to be computed.

In the next section, we derive expressions for Q_i and P_i that represent an alternative to those given in Proposition 4.1. They make use of the recurrences C_i, J_i and A_i as given in (3.3) and (3.1), respectively, as well as another new sequence of polynomials E_i given in (6.2) below.

6. More mumford representation formulae

The discussion of NUCOMP in [14] used four linear sequences to express the Mumford coefficients of a divisor. Here, we proceed similarly and introduce one more auxiliary linear sequence of polynomials to accompany the already familiar

sequences C_i and J_i of (3.3) and A_i of (3.1). Using the same notation as in the previous section, put

$$R_0 = \frac{P_0^2 - hP_0 - f}{Q_0} = -Q_{-1} \tag{6.1}$$

and define

$$E_{-2} = R_0, \quad E_{-1} = h - P_0, \quad E_i = q_i E_{i-1} + E_{i-2} \qquad (0 \le i \le m). \tag{6.2}$$

Then

$$E_i = (h - P_0)A_i + R_0 B_i = E_{-1}A_i + E_{-2}B_i \qquad (-2 \le i \le m), \tag{6.3}$$

where A_i and B_i are as in (3.1). Two more auxiliary identities, easily verified by applying induction to (3.1), (3.3) and (6.2), will facilitate the development of our final formulae:

$$\begin{aligned} P_0 &= A_i C_{i-1} + A_{i-1} C_i, \\ P_0 - h &= J_i E_{i-1} + J_{i-1} E_i, \end{aligned} \qquad (-1 \le i \le m). \tag{6.4}$$

We are now ready to provide our desired expressions for P_i and Q_i. These formulae are reminiscent of (7.9) and (7.10) of [14].

Theorem 6.1. *Let $Q_0, P_0 \in \mathbb{F}_q[x]$ with Q_0 dividing $f + hP_0 - P_0^2$, and let Q_i, P_i $(1 \le i \le m+1)$ be defined by (4.1). For $-2 \le i \le m$, let C_i, J_i, A_i and E_i be defined by (3.3), (3.1), and (6.2), with $q_i = \lfloor C_{i-2}/C_{i-1} \rfloor$ for $0 \le i \le m$. Then*

$$\begin{aligned} Q_{i+1} &= A_i C_i + E_i J_i \\ P_{i+1} &= A_i C_{i-1} + E_{i-1} J_i = h - A_{i-1} C_i + E_i J_{i-1} \end{aligned} \qquad (-1 \le i \le m). \tag{6.5}$$

Proof. From (3.5) and (3.4), we obtain

$$A_i = \frac{(-1)^{i-1} C_i + B_i P_0}{Q_0} = \frac{(-1)^{i-1}}{Q_0} (C_i + J_i P_0). \tag{6.6}$$

Thus,

$$\begin{aligned} A_i C_i + E_i J_i &= A_i C_i + ((h - P_0)A_i + R_0 B_i) J_i \qquad \text{by (6.3)} \\ &= A_i (C_i + (h - P_0) J_i) + (-1)^{i-1} J_i^2 R_0 \qquad \text{by (3.4)} \\ &= \frac{(-1)^{i-1}}{Q_0} (C_i + J_i P_0)(C_i + (h - P_0) J_i) \\ &\qquad + \frac{(-1)^{i-1}}{Q_0} (J_i^2 (P_0^2 - hP_0 - f)) \qquad \text{by (6.6) and (6.1)} \\ &= \frac{(-1)^{i-1}}{Q_0} (C_i^2 + C_i J_i h - J_i^2 f) = Q_{i+1} \qquad \text{by Proposition 4.1.} \end{aligned}$$

Now $A_{-1}C_{-2} - E_{-2}J_{-1} = C_{-2} = P_0$ and $h - A_{-2}C_{-1} + E_{-1}J_{-2} = h - E_{-1} = P_0$. For $i \ge 0$, inductively

$$\begin{aligned} P_{i+1} &= q_i Q_i - P_i + h \qquad \text{by (4.1)} \\ &= q_i (A_{i-1} C_{i-1} + E_{i-1} J_{i-1}) - (h - A_{i-2} C_{i-1} + E_{i-1} J_{i-2}) + h \\ &\qquad \text{using the above expressions for } Q_i \text{ and } P_i \\ &= (q_i A_{i-1} + A_{i-2}) C_{i-1} + E_{i-1} (q_i J_{i-1} - J_{i-2}) \\ &= A_i C_{i-1} - E_{i-1} J_i \qquad \text{by (3.1) and (3.3).} \end{aligned}$$

By (6.4), we finally obtain

$$P_{i+1} = (P_0 - A_{i-1}C_i) - (P_0 - h - E_iJ_{i-1}) = h - A_{i-1}C_i + E_iJ_{i-1}. \qquad \square$$

7. The reduction algorithm, variations, and experiments

Finding a reduced divisor linearly equivalent to $D_1 = (Q_0, P_0)$ now simply requires applying the Euclidean algorithm to P_0/Q_0, i.e. computing the sequence of quotients q_i and remainders C_i, until the index r is found as defined in (5.1). Then the Mumford coefficients Q_r and P_r of the reduced or almost reduced divisor D_{r+1} need to be recovered.

In the preceding sections, several options for recovering these coefficients have been presented. These variations are distinguished by which recurrences are computed along with the remainders C_i and which formulae are used at the end. Although more versions are possible, we only list three below, having chosen not to consider those that involve more than one division by the large degree Q_0.

1. Compute only J_i recursively along with C_i, then compute Q_r via Proposition 4.1 and P_r via the second formula of Lemma 4.2. This requires one division by Q_0 and one by the much smaller polynomial J_r, plus a number of multiplications involving not too large operands.
2. Compute both A_i and J_i recursively along with C_i, then $E_r = (h - P_0)A_r + (-1)^{r+1}R_0J_r$, and finally Q_{r+1} and P_{r+1} via Theorem 6.1, using the second of the two formulae for P_{r+1}. This requires one division by Q_0 and one multiplication involving two large operands, $P_0(h - P_0)$, to compute R_0, two multiplications involving one small operand and one large operand, $(h-P_0)A_r$ and $(-1)^{r+1}R_0J_r$, to compute E_r, four multiplications involving medium size operands for the formulae of Theorem 6.1, and two recurrences.
3. Compute all three sequences A_i, J_i, E_i recursively along with the C_i. This requires one division by Q_0 and one multiplication involving two large operands, $P_0(h - P_0)$, to compute E_{-2}, four multiplications involving medium size operands for the formulae of Theorem 6.1, and three recurrences.

It is easy to see that at least one of the sequences A_i, E_i, J_i needs to be computed with C_i, since not all three sequences can be recovered from C_i alone. It is unknown whether D_{r+1} can be found without any polynomial divisions.

We implemented all three versions of our reduction algorithm for divisor reduction in imaginary hyperelliptic curves and compared their performance to that used in Cantor's algorithm. We used the computer algebra library NTL [24] for finite field and polynomial arithmetic and the GNU `C++` compiler version 4.1.2. The computations described below were performed on an Intel Core Duo 2.66 GHz processor running Linux. All four algorithms were implemented using curves defined over finite prime fields $\mathbb{F}_p$ and characteristic 2 finite fields $\mathbb{F}_{2^n}$.

For our experiments, we used finite fields $\mathbb{F}_q$ where q has $2, 4, 8, 16, 32$, and 64 bits. The odd q were taken to be the smallest prime of the given length. For each value of q, we selected 5 random imaginary hyperelliptic curves of genus 2–15, 20, 25, and 30. For each curve, we applied the four reduction algorithms to random prime divisors $D = (Q, P)$, Q irreducible, for which $\deg(Q) = mg$ with $m \in \{2, 4, 8, 16, 32, 64, 128, 256\}$. We recorded the average time required to reduce a prime divisor of a given size using each algorithm, taken over all prime divisors and curves for a fixed genus and q.

Of the three variations of our new reduction algorithm, the first variation was the most efficient in all cases except $m = 2$. In other words, the versions that require computing more than the J_i recursively do not seem to offer any performance improvements.

Unfortunately, the new reduction algorithm does not compare so favorably to the basic version due to Cantor; we address a possible reason for this in the last section. In the following tables, we give the average times per reduction (in milliseconds) for all the genera g and scalars m listed above. Table 7.1 gives the times for curves defined over $\mathbb{F}_p$ where $p = 65537$ is a 16-bit prime, and Table 7.2 gives the times for curves defined over $\mathbb{F}_{2^{16}}$. The results for the other finite fields we tested are similar, so these data are omitted for brevity. Times are given for Cantor's reduction algorithm and the first version of our reduction algorithm. Shaded cells indicate that for the given genus g and scalar m, the new reduction algorithm was the faster of the two.

TABLE 7.1. Average reduction times (in ms) for imaginary hyperelliptic curves defined over $\mathbb{F}_{65537}$.

g	Alg	m							
		2	4	8	16	32	64	128	256
2	Cantor	0.0012	0.0036	0.0080	0.0224	0.0640	0.2047	0.9343	2.5976
	New	0.0028	0.0044	0.0104	0.0236	0.0620	0.1815	0.6769	2.1992
3	Cantor	0.0020	0.0056	0.0144	0.0412	0.1232	0.4205	1.4731	5.5248
	New	0.0052	0.0080	0.0168	0.0400	0.1139	0.3677	1.2455	4.6535
4	Cantor	0.0028	0.0080	0.0216	0.0624	0.2217	0.7286	2.5484	9.6863
	New	0.0052	0.0108	0.0228	0.0612	0.2085	0.6327	2.1720	8.0392
5	Cantor	0.0032	0.0104	0.0304	0.0904	0.3063	1.0549	3.8730	14.7647
	New	0.0060	0.0136	0.0316	0.0856	0.2678	0.8967	3.2222	12.2941
6	Cantor	0.0044	0.0140	0.0400	0.1235	0.4249	1.4790	5.5227	21.2500
	New	0.0064	0.0160	0.0392	0.1120	0.3610	1.2395	4.5455	17.3333
7	Cantor	0.0048	0.0172	0.0500	0.1580	0.5538	1.9681	7.3939	28.8889
	New	0.0088	0.0200	0.0524	0.1504	0.4813	1.6733	6.1515	23.5556
8	Cantor	0.0060	0.0200	0.0616	0.2016	0.7046	2.5291	9.5600	37.2308
	New	0.0092	0.0232	0.0624	0.1882	0.6177	2.1693	8.1600	30.9231
9	Cantor	0.0068	0.0240	0.0740	0.2570	0.8824	3.1892	12.1500	47.2727
	New	0.0108	0.0280	0.0776	0.2438	0.7757	2.7432	10.0500	39.0909
10	Cantor	0.0080	0.0272	0.0880	0.3085	1.0540	3.8413	14.8485	58.6667
	New	0.0120	0.0300	0.0880	0.2829	0.9158	3.2698	12.2424	47.7778
11	Cantor	0.0096	0.0320	0.1029	0.3563	1.2623	4.6214	17.8571	71.0000
	New	0.0136	0.0360	0.1025	0.3265	1.0909	3.9612	14.9286	57.7500
12	Cantor	0.0100	0.0360	0.1175	0.4081	1.4819	5.4667	21.1667	83.6667
	New	0.0148	0.0412	0.1147	0.3660	1.2530	4.5556	17.1667	68.3333
13	Cantor	0.0108	0.0416	0.1362	0.4737	1.7014	6.3896	24.6000	97.3333
	New	0.0160	0.0460	0.1325	0.4173	1.4514	5.3506	20.3000	81.0000
14	Cantor	0.0136	0.0460	0.1538	0.5421	1.9370	7.3433	28.7778	114.0000
	New	0.0180	0.0504	0.1476	0.4752	1.6457	6.1791	23.6667	92.4000
15	Cantor	0.0140	0.0504	0.1728	0.6343	2.2617	8.5172	32.9333	130.0000
	New	0.0200	0.0572	0.1713	0.5524	1.9159	7.1034	27.2000	106.5000
20	Cantor	0.0204	0.0804	0.2921	1.0508	3.9523	14.9697	58.4444	232.0000
	New	0.0300	0.0840	0.2783	0.9237	3.3175	12.3636	47.7778	189.3333
25	Cantor	0.0292	0.1173	0.4650	1.6329	6.1205	23.0909	90.3333	360.0000
	New	0.0400	0.1247	0.4256	1.3734	5.0361	19.0000	74.6667	290.0000
30	Cantor	0.0380	0.1599	0.6914	2.3258	8.6207	33.3333	132.0000	522.0000
	New	0.0524	0.1785	0.5725	1.9005	7.0690	27.0667	105.0000	420.0000

TABLE 7.2. Average reduction times (in ms) for imaginary hyperelliptic curves defined over $\mathbb{F}_{2^{16}}$.

		m							
g	Alg	2	4	8	16	32	64	128	256
2	Cantor	0.0076	0.0204	0.0608	0.1934	0.6640	2.4390	9.2453	36.0000
	New	0.0156	0.0368	0.0804	0.2277	0.7122	2.5268	9.4340	36.5714
3	Cantor	0.0132	0.0384	0.1168	0.3949	1.4057	5.2747	20.3333	80.0000
	New	0.0316	0.0608	0.1440	0.4360	1.4800	5.4286	21.0000	82.6667
4	Cantor	0.0172	0.0580	0.1912	0.6649	2.4433	9.1321	35.7143	142.5000
	New	0.0384	0.0820	0.2227	0.6995	2.5025	9.4340	37.1429	145.5000
5	Cantor	0.0244	0.0832	0.2826	1.0040	3.7077	14.2941	56.0000	221.3333
	New	0.0524	0.1124	0.3200	1.0482	3.8615	14.7059	58.2222	230.6667
6	Cantor	0.0304	0.1098	0.3887	1.4028	5.3118	20.4167	80.0000	319.0000
	New	0.0616	0.1391	0.4170	1.4366	5.3763	21.1667	83.6667	331.0000
7	Cantor	0.0392	0.1450	0.5143	1.8797	7.0882	27.5556	108.4000	432.0000
	New	0.0740	0.1792	0.5500	1.9398	7.2941	28.5556	113.2000	451.0000
8	Cantor	0.0464	0.1796	0.6468	2.4251	9.2075	35.8571	142.5000	566.0000
	New	0.0820	0.2129	0.6742	2.4444	9.3208	37.0000	147.5000	582.0000
9	Cantor	0.0584	0.2221	0.8137	3.0488	11.5238	45.4545	178.6667	708.0000
	New	0.0988	0.2648	0.8385	3.0488	11.8095	46.7273	186.0000	740.0000
10	Cantor	0.0660	0.2650	0.9925	3.6889	14.1765	56.0000	220.0000	880.0000
	New	0.1080	0.3022	1.0037	3.7333	14.5294	57.5556	230.0000	910.0000
11	Cantor	0.0784	0.3191	1.1937	4.4643	17.1034	67.5000	266.0000	1064.0000
	New	0.1260	0.3510	1.1937	4.4642	17.4483	69.2500	275.0000	1098.0000
12	Cantor	0.0888	0.3684	1.3963	5.2632	20.4167	80.0000	318.0000	1260.0000
	New	0.1360	0.3958	1.3753	5.2421	20.6667	83.0000	329.0000	1306.0000
13	Cantor	0.1048	0.4352	1.6135	6.1481	24.0000	94.0000	374.0000	1490.0000
	New	0.1568	0.4659	1.6074	6.1481	24.4000	97.6667	388.0000	1554.0000
14	Cantor	0.1156	0.4918	1.8737	7.0704	27.6667	108.8000	430.0000	1716.0000
	New	0.1684	0.5177	1.8316	7.0423	28.0000	112.4000	450.0000	1792.0000
15	Cantor	0.1341	0.5546	2.1200	8.0645	31.7500	125.0000	496.0000	1972.0000
	New	0.1929	0.5801	2.3440	8.0968	31.8750	127.5000	514.0000	2042.0000
20	Cantor	0.2154	0.9494	3.6806	14.3429	55.5556	220.6667	882.0000	3498.0000
	New	0.2865	0.9494	3.5417	14.1143	57.3333	230.0000	916.0000	3676.0000
25	Cantor	0.3248	1.4428	5.6559	22.1739	87.0000	346.0000	1374.0000	5484.0000
	New	0.4002	1.4179	5.4624	21.7391	88.3333	357.0000	1432.0000	5684.0000
30	Cantor	0.4535	2.0280	8.1212	31.7500	125.5000	496.0000	1980.0000	7890.0000
	New	0.5290	1.9231	7.6364	31.5000	127.5000	512.0000	2050.0000	8204.0000

As can be seen from the tables, our new reduction algorithm does consistently out-perform Cantor's algorithm once m is sufficiently large in the odd characteristic case. It also appears that its relative performance improves slightly as the genus increases. Both these phenomena are what one would expect; as more reduction steps are required, the benefits of replacing them by the cheaper Euclidean algorithm steps should accumulate. The even characteristic case does not look as promising, with our algorithm only out-performing Cantor's for a few cases with relatively large genus.

8. Eliminating the extra step – alternative real hyperelliptic curve models

We revisit the unfortunate scenario where Proposition 4.1 does not yield a reduced divisor. Recall that this situation was encountered exactly when $\deg(J_r) = N$, $\deg(Q_0) - g$ is odd, and $\deg(f) = 2g + 2$. Suppose that C is given by (2.1)

with $\deg(f) = 2g + 2$. We generally expect $\deg(J_r)$ to increase in steps of one as i increases, so $\deg(J_r) = N$ will likely occur. We have no influence over this. This leaves the investigation of what to do if $\deg(Q_0) - g$ is odd.

Suppose that $D_1 = (Q_0, P_0, (\delta_1))$ is a scalar multiple of some reduced divisor $D = (Q, P, (\delta))$; this is a common situation in hyperelliptic curve arithmetic and cryptography. Say $D_1 = mD$. Then one expects $\deg(Q) = g$ and $\deg(Q_0) = mg$. If m is odd — this is more likely to occur when the order q of the base field is odd — then $\deg(Q_0) - g = (m - 1)g$ is even, so our problem case does not happen.

The case m even, i.e. $\deg(Q_0) - g$ odd, is more likely to occur over fields of even characteristic. In this case, one can use a hyperelliptic curve model (with the same polynomial h) that is isomorphic to C and avoids the problem scenario. For our purposes, it suffices to ensure $\deg(f) \leq 2g + 1$, but it is possible to obtain a much lower bound, namely $\deg(f) \leq g$. In fact, if q is odd, this also produces an isomorphic model with a lower degree right hand side, but at the expense of introducing a y-coefficient.

Theorem 8.1. *Let $C : y^2 + h(x) = f(x)$ be a real hyperelliptic curve of genus g over any finite field $\mathbb{F}_q$ as given in (2.1) with $\deg(f) = 2g + 2$. Then C is isomorphic to a real hyperelliptic curve $C' : z^2 + H(x)z = F(x)$ over $\mathbb{F}_q$, where $\deg(F) \leq g$, $\deg(H) = g + 1$, and $\mathbb{F}_q[x, z] = \mathbb{F}_q[x, y]$.*

Proof. Recall that $y \in \mathbb{F}_q\langle x^{-1}\rangle$ and $\deg(y) = g + 1$. Set $Y = \lfloor y \rfloor$, $z = y - Y$, $H = h + 2Y$, and $F = f - Y^2 - hY$. Then $\mathbb{F}_q[x, y] = \mathbb{F}_q[x, z]$, $\deg(y - Y) < 0$ and $\deg(Y + y + h) = g + 1$. Since

$$F = (y^2 + hy) - (Y^2 + hY) = (y - Y)(Y + y + h),$$

we have $\deg(F) \leq g$. Substituting $y = z + Y$ into C yields $z^2 + Hz = F$, as desired. □

If q is even, then $H = h$, and C' requires far less storage than C, namely $2g + 2 - \deg(F) \geq g + 2$ fewer elements in $\mathbb{F}_q$. For q odd, there may be no reduction in storage when switching from C to C'. The model C requires storing the $2g + 3$ coefficients of f, while C' needs space for the $g + 2$ coefficients of h and up to $g + 1$ coefficients of F.

In practice, finding Y requires computing the first $g+1$ coefficients of the Laurent expansion of y in $\mathbb{F}_q\langle x^{-1}\rangle$. Write $y = sx^{g+1} + y_g x^g + \cdots + y_0 + y_{-1}x^{-1} + \cdots$ with $s = \operatorname{sgn}(y)$; note that $s = \pm 1$ if q is odd. Then it is easy to see that y_i satisfies a linear equation in the higher-indexed coefficients y_j $(i+1 \leq j \leq g)$ whose coefficients involve s^{-1} as well as the coefficients of f, and of h if q is even. So finding Y requires solving $g + 1$ linear equations over $\mathbb{F}_q$ to successively find $y_g, y_{g-1}, \ldots, y_0$. If q is even, then a quadratic equation over $\mathbb{F}_q$ may also need to be solved to find s, plus one inversion to obtain s^{-1}.

We conclude with the remark that this kind of variable transformation has no analog for imaginary hyperelliptic curves. In addition to eliminating the need for an extra reduction step in our reduction algorithm, this model warrants further investigation in that it might offer some advantages for divisor arithmetic, possibly saving some field operations in the context of low-genus explicit formulae. For q even, arithmetic on C' will certainly be no slower than that on C; for q odd, it is unclear how the two models compare. In either case, it would be interesting to explore in detail how divisor arithmetic on C' compares to that on C in terms of efficiency; this is the subject of ongoing research.

9. Conclusions and further work

As shown in Section 7, our reduction algorithm does offer some performance improvements for reducing sufficiently large divisors. When using curves over prime fields, our algorithm is faster than Cantor's algorithm in all cases as soon as the size of the divisor to reduce is sufficiently large. For even characteristic fields, our results are not as convincing, with improvements realized in only a few cases.

It should be possible to improve the performance of our algorithm further. In the number field version described in [12], the best performance is only realized when using an accelerated partial extended GCD algorithm. An adaptation of Lehmer's algorithm [16] works well, especially with large operands, as it replaces most of the multi-precision integer operations with single precision operations. The half-GCD algorithm translates some of the ideas of Lehmer's algorithm to the polynomial case. A version of this algorithm that stops part-way through the computation should greatly improve the efficiency of our reduction algorithm, and hopefully make it more competitive in the even characteristic case.

Another open question is whether our reduction algorithm can be used to improve the m-tuple based scalar multiplication techniques mentioned in Section 1. To properly test this, it will be necessary to develop explicit formulae for the most cryptographically interesting scenarios, i.e., genus at most 3 and certain specific sizes of non-reduced divisors. In addition, these should be implemented using dedicated finite field arithmetic routines tailored to specific fields of interest. This, as well as the aforementioned issues, are the subject of ongoing research.

Acknowledgments

The authors thank Alf van der Poorten for first directing us to the alternative hyperelliptic curve model presented in Section 8. We also appreciate the feedback from an anonymous referee.

References

[1] R. Avanzi, *A note on the signed sliding window integer recoding and its left-to-right analogue*, in "Selected Areas in Cryptography" (eds. H. Handschuh and M.A. Hasan), Springer, (2005), 130–143.

[2] D. G. Cantor, *Computing in the Jacobian of a hyperelliptic curve*, Math. Comput., **48** (1987), 95–101.

[3] H. Cohen, A. Miyaji and T. Ono, *Efficient elliptic curve exponentiation*, in "Information and Communications Security" (eds. Y. Han, T. Okamoto and S. Qing), Springer, (1997), 282–290.

[4] V. S. Dimitrov, L. Imbert and P. K. Mishra, *Efficient and secure elliptic curve point multiplication using double base chains*, in "Advances in Cryptology - ASIACRYPT 2005" (ed. B. Roy), Springer, (2005), 59–79.

[5] V. S. Dimitrov, G. A. Jullien and W. C. Miller, *An algorithm for modular exponentiation*, Inf. Proc. Letters, **66** (1998), 155–159.

[6] K. Eisenträger, K. Lauter and P. L. Montgomery, *Fast elliptic curve arithmetic and improved Weil pairing evaluation*, in "Topics in Cryptology — CT-RSA 2003" (ed. M. Joye), Springer, (2003), 343–354.

[7] S. Galbraith, M. Harrison and D. J. Mireles Morales, *Efficient hyperelliptic arithmetic using balanced representation for divisors*, in "Algorithmic Number Theory – ANTS-VIII" (eds. A. van der Poorten and A. Stein), Springer, Berlin, (2008), 342–356.

[8] D. Gordon, *A survey of fast exponentiation methods*, J. Algorithms, **27** (1998), 129–146.

[9] L. Imbert, M. J. Jacobson, Jr. and A. Schmidt, *Fast ideal cubing in imaginary quadratic number and function fields*, Adv. Math. Commun., **4** (2010), 237–260.

[10] M. J. Jacobson, Jr., A. J. Menezes and A. Stein, *Hyperelliptic curves and cryptography*, in "High Primes and Misdemeanors: Lectures in Honour of the 60th Birthday of Hugh Cowie Williams" (eds. A.J. van der Poorten and A. Stein), Fields Inst. Comm., **41** (2004), 255–282.
[11] M. J. Jacobson, Jr. and A. J. van der Poorten, *Computational aspects of NUCOMP*, in "Algorithmic Number Theory" (eds. C. Fieker and D.R. Kohel), Springer, Berlin, (2002), 120–133.
[12] M. J. Jacobson, Jr., R. E. Sawilla and H. C. Williams, *Efficient ideal reduction in quadratic fields*, Intern. J. Math. Comp. Sci., **1** (2006), 83–116.
[13] M. J. Jacobson, Jr., R. Scheidler and A. Stein, *Cryptographic protocols on real and imaginary hyperelliptic curves*, Adv. Math. Commun., **1** (2007), 197–221.
[14] M. J. Jacobson, Jr., R. Scheidler and A. Stein, *Fast arithmetic on hyperelliptic curves via continued fraction expansions*, in "Advances in Coding Theory and Cryptology" (eds. T. Shaska, W.C. Huffman, D. Joyner and V. Ustimenko), World Scientific Publishing Co. Pte. Ltd., Hackensack, New Jersey, (2007), 201–244.
[15] N. Koblitz, *An elliptic curve implementation of the finite field digital signature algorithm*, in "Advances in Cryptology—CRYPTO '98" (ed. H. Krawczyk), Springer, Berlin, (1998), 327–337.
[16] D. H. Lehmer, *Euclid's algorithm for large numbers*, Amer. Math. Monthly, **45** (1938), 227–233.
[17] A. J. Menezes, Y.-H. Wu and R. J. Zuccherato, *An elementary introduction to hyperelliptic curves*, in "Algebraic Aspects of Cryptography," Springer-Verlag, Berlin, (1998), 155–178.
[18] D. Mumford, "Tata Lectures on Theta II," Birkhäuser, 1984.
[19] S. Paulus and H.-G. Rück, *Real and imaginary quadratic representations of hyperelliptic function fields*, Math. Comput., **68** (1999), 1233–1241.
[20] A. J. van der Poorten, *A note on NUCOMP*, Math. Comput., **72** (2003), 1935–1946.
[21] R. E. Sawilla, "Fast Ideal Arithmetic in Quadratic Fields," Master Thesis, University of Calgary, 2004, available online at http://www.sawilla.com/docs/Sawilla_thesis.pdf.
[22] C. P. Schnorr and M. Seysen, *An improved composition algorithm*, unpublished manuscript, August 1983.
[23] D. Shanks, *On Gauss and composition*, in "Number Theory and Applications" (ed. R.A. Mollin), Kluwer Academic Publishers, (1989), 163–204.
[24] V. Shoup, *NTL: A library for doing number theory*, Software, 2001, available online at http://www.shoup.net/ntl.
[25] J. A. Solinas, *An improved algorithm for arithmetic on a family of elliptic curves*, in "Advances in Cryptology CRYPTO '97" (ed. B.S. Kaliski, Jr.), Springer, (1997), 357–371.

Received June 2009; revised March 2010.

E-mail address: roberto.avanzi@ruhr-uni-bochum.de
E-mail address: jacobs@cpsc.ucalgary.ca
E-mail address: rscheidl@math.ucalgary.ca