LMS J. Comput. Math. 19 (Special issue A) (2016) 391–405 © 2016 Authors
doi:10.1112/S1461157016000139

Computing canonical heights on elliptic curves in quasi-linear time

J. Steffen Müller and Michael Stoll

Abstract

We introduce an algorithm that can be used to compute the canonical height of a point on an elliptic curve over the rationals in quasi-linear time. As in most previous algorithms, we decompose the difference between the canonical and the naive height into an archimedean and a non-archimedean term. Our main contribution is an algorithm for the computation of the non-archimedean term that requires no integer factorization and runs in quasi-linear time.

1. *Introduction*

Let E denote an elliptic curve defined over a number field K. The canonical height is a quadratic form $\hat{h}\colon E(K)\otimes\mathbb{R}\to\mathbb{R}$, first constructed by Néron [13] and Tate (unpublished). For several applications, such as computing generators for $E(K)$ and computing the regulator appearing in the conjecture of Birch and Swinnerton–Dyer, one needs to compute $\hat{h}(P)$ for points $P\in E(K)$.

To this end, one typically chooses a Weierstrass equation W for E over K with $\mathcal{O}_K$-integral coefficients and decomposes $\hat{h}(P)$ (or $\hat{h}(P)-h(P)$, where h is the naive height on E with respect to W) into a sum of local terms, one for each place of K. For simplicity, let us assume that $K=\mathbb{Q}$. There are several efficient algorithms for the computation of the contribution at infinity (see § 5). A very simple and efficient algorithm of Silverman [14] can be used to compute the non-archimedean contributions separately. However, in order to determine the non-archimedean places which contribute to $\hat{h}(P)$ (or $\hat{h}(P)-h(P)$), the algorithm of [14] assumes that the prime factorization of the discriminant $\Delta(W)$ is known, which renders this approach inefficient when the coefficients of W are large. This observation motivated Silverman's article [16], where it is shown how to compute $\hat{h}(P)$ without the need to factorize $\Delta(W)$. Nevertheless, the algorithm of [16] requires the prime factorization of $\gcd(c_4(W),c_6(W))$ in order to find a globally minimal Weierstrass equation for E.

In this note, we introduce an algorithm for the computation of $\hat{h}(P)$ that does not require any factorization into primes at all and runs in time that is quasi-linear in the size of the input data and the desired precision of the result. More precisely, let $\|W\|$ denote the largest absolute value of the coefficients of W and let d denote the number of desired bits of precision after the binary point. We denote the time needed to multiply two d-bit integers by $\mathsf{M}(d)$. The following result is our main theorem. Recall the 'soft-O' notation: $f(n)\in\tilde{O}(g(n))$ means that there are constants $c,m>0$ such that, for n sufficiently large, $|f(n)|\leqslant cg(n)(\log g(n))^m$. Using fast multiplication algorithms, $\mathsf{M}(d)\in\tilde{O}(d)$. We also use the notation $f(n)\ll g(n)$ to express the fact that there is a constant $c>0$ such that $|f(n)|\leqslant cg(n)$ for n sufficiently large.

Received 15 January 2016.

2010 Mathematics Subject Classification 11G50 (primary), 14G40, 11G05, 11Y16 (secondary).

Contributed to the Twelfth Algorithmic Number Theory Symposium (ANTS-XII), Kaiserslautern, Germany, 29 August–2 September 2016.

THEOREM 1.1. *Let E be given by a Weierstrass equation W with coefficients in $\mathbb{Z}$ and let $P \in E(\mathbb{Q})$. Then we can compute $\hat{h}(P)$ to d bits of (absolute) precision in time*

$$\begin{aligned}&\ll \log(d+h(P))\,\mathsf{M}(d+h(P)) + (\log\log\|W\|)^2\,\mathsf{M}((\log\log\|W\|)(\log\|W\|))\\ &\quad + \log(d+\log\|W\|)^2\,\mathsf{M}(d+\log\|W\|)\\ &\in \tilde{O}(d+h(P)+\log\|W\|).\end{aligned}$$

Since the size of the input is measured by $h(P)+\log\|W\|$ (the first term gives the size of P, the second term gives the size of W) and the size of the output is measured by $\log h(P)+d$, this means that we can compute $\hat{h}(P)$ in quasi-linear time.

The strategy of the proof is to first find an algorithm for the computation of the local non-archimedean contributions that does not assume minimality (see Proposition 4.3). Building on this, the non-archimedean contribution to $\hat{h}(P)-h(P)$ can be computed upon observing that it is a sum of rational multiples of logarithms of prime numbers, which can be determined by working globally modulo a suitable power of $\Delta(W)$. Combining this with a complexity analysis of the fastest known algorithm for the computation of the local height at infinity due to Bost and Mestre [**6**], Theorem 1.1 follows. We note that Marco Caselli, working on his PhD under the supervision of Cremona, is currently extending the Bost–Mestre algorithm to also deal with complex places.

The paper is organized as follows. In § 2, we set up some notation and introduce the notion of Kummer coordinates of points on elliptic curves. Heights and their local decompositions are recalled in § 3. In § 4, we discuss an algorithm that allows us to compute a non-archimedean local summand of $\hat{h}(P)-h(P)$ efficiently, without assuming minimality, and we estimate its running time. Section 5 contains a discussion of the algorithm of Bost–Mestre for the computation of the local height at infinity and of its running time. We then combine the non-archimedean and the archimedean results into an efficient algorithm for the computation of $\hat{h}(P)$ in § 6, leading to a proof of Theorem 1.1. Finally, in § 7, we discuss the practicality of our algorithm.

2. *Kummer coordinates*

Let K be a field and consider an elliptic curve E/K, given by a Weierstrass equation

$$W\colon y^2+a_1xy+a_3y = x^3+a_2x^2+a_4x+a_6, \tag{2.1}$$

where $a_1,a_2,a_3,a_4,a_6 \in K$. As usual, let

$$\begin{aligned}b_2 &= a_1^2+4a_2,\\ b_4 &= 2a_4+a_1a_3,\\ b_6 &= a_3^2+4a_6,\\ b_8 &= a_1^2a_6+4a_2a_6-a_1a_3a_4+a_2a_3^2-a_4^2,\end{aligned}$$

and let

$$\Delta(W) = -b_2^2b_8-8b_4^3-27b_6^2+9b_2b_4b_6$$

denote the discriminant of the equation W. We define the functions g and f for $P \in E(K)\setminus\{O\}$ by

$$\begin{aligned}g(P) &= x(P)^4-b_4x(P)^2-2b_6x(P)-b_8,\\ f(P) &= 4x(P)^3+b_2x(P)^2+2b_4x(P)+b_6.\end{aligned}$$

Then, for $P \in E(K)\setminus E[2]$, $x(2P)=g(P)/f(P)$. We now extend this to all $P \in E(K)$.

Note that $\mathbb{P}^1$ is the Kummer variety $E/\{\pm 1\}$ of E. An explicit covering map $E \to \mathbb{P}^1$ is given by

$$\begin{aligned} \kappa: \qquad E &\longrightarrow \mathbb{P}^1 \\ (x:y:1) &\longmapsto (x:1) \\ O &\longmapsto (1:0). \end{aligned}$$

We call $(x_1, x_2) \in \mathbb{A}^2(K)\backslash\{(0,0)\}$ a pair of *Kummer coordinates* for $P \in E(K)$ if $\kappa(P) = (x_1 : x_2)$.

The degree 4 homogenizations of g and f are

$$\begin{aligned} \delta_1(x_1,x_2) &= x_1^4 - b_4x_1^2x_2^2 - 2b_6x_1x_2^3 - b_8x_2^4, \\ \delta_2(x_1,x_2) &= 4x_1^3x_2 + b_2x_1^2x_2^2 + 2b_4x_1x_2^3 + b_6x_2^4, \end{aligned}$$

respectively. For $(x_1, x_2) \in \mathbb{A}^2_K$, we set

$$\delta(x_1,x_2) = (\delta_1(x_1,x_2), \delta_2(x_1,x_2)).$$

It follows that if (x_1, x_2) is a pair of Kummer coordinates for $P \in E(K)$, then $\delta(x_1, x_2)$ is a pair of Kummer coordinates for $2P$.

3. *Heights*

Let K be a number field and let E/K be an elliptic curve given by a Weierstrass equation W, as in (2.1). We denote by M_K the set of places of K. For a place $v \in M_K$, we normalize the associated absolute value $|\cdot|_v$ so that it restricts to the usual absolute value on $\mathbb{Q}$ when v is an infinite place and so that $|p|_v = p^{-1}$ when v is a finite place above p. We write $n_v = [K_v : \mathbb{Q}_w]$ for the local degree, where w is the place of $\mathbb{Q}$ below v. Then we have the product formula $\prod_{v \in M_K} |x|_v^{n_v} = 1$ for all $x \in K^\times$. The *naive height* of $P \in E(K)\backslash\{O\}$ with respect to W is given by

$$h(P) = \frac{1}{[K:\mathbb{Q}]} \sum_{v \in M_K} n_v \log \max\{|x_1|_v, |x_2|_v\},$$

where (x_1, x_2) is a pair of Kummer coordinates for P. Note that $h(P)$ does not depend on the choice of (x_1, x_2), by the product formula.

The limit

$$\hat{h}(P) = \lim_{n\to\infty} \frac{h(nP)}{n^2}$$

exists and is called the *canonical height* (or *Néron–Tate height*) of P.

For the computation of $\hat{h}(P)$, the limit construction is not suitable due to slow convergence and exponential growth of the size of the coordinates. Instead, one decomposes $\hat{h}(P)$ into local terms. We now recall how this can be achieved, following [10]. For $v \in M_K$ and $Q \in E(K_v)$, we set

$$\Phi_v(Q) = \frac{\max\{|\delta_1(x_1,x_2)|_v, |\delta_2(x_1,x_2)|_v\}}{\max\{|x_1|_v, |x_2|_v\}^4},$$

where $(x_1, x_2) \in \mathbb{A}^2(K_v)\backslash\{0, 0\}$ is a pair of Kummer coordinates for Q. Since δ_1 and δ_2 are homogeneous of degree 4, $\Phi_v(Q)$ does not depend on the choice of (x_1, x_2). The function Φ_v is continuous and bounded on $E(K_v)$, so it makes sense to define

$$\Psi_v(Q) = -\sum_{n=0}^{\infty} 4^{-n-1} \log \Phi_v(2^n Q),$$

which is, likewise, continuous and bounded. Note that for $P \in E(K)$,

$$h(2P) - 4h(P) = \frac{1}{[K:\mathbb{Q}]} \sum_{v \in M_K} n_v \log \Phi_v(P),$$

and Tate's telescoping trick yields the formula

$$\hat{h}(P) = h(P) - \frac{1}{[K:\mathbb{Q}]} \sum_{v \in M_K} n_v \Psi_v(P), \tag{3.1}$$

which we will use to compute the canonical height.

It is also possible to decompose the canonical height into a sum of local height functions. For $v \in M_K$ and $Q \in E(K_v)\backslash\{O\}$, we define the *local height* of Q as

$$\hat{\lambda}_v(Q) = \log \max\{1, |x(Q)|_v\} - \Psi_v(Q).$$

Then (3.1) immediately implies that

$$\hat{h}(P) = \frac{1}{[K:\mathbb{Q}]} \sum_{v \in M_K} n_v \hat{\lambda}_v(P) \tag{3.2}$$

for $P \in E(K)\backslash\{O\}$.

REMARK 1. Several normalizations for the local height on elliptic curves can be found in the literature (see the discussion in [**10**]). Our normalization corresponds to that used in [**10**], so in particular, our canonical height is twice the canonical height in Silverman's paper [**14**] and in his books on elliptic curves. More precisely, we have

$$\hat{\lambda}_v(Q) = 2\hat{\lambda}_v^{\mathrm{SilB}}(Q) + \tfrac{1}{6}\log|\Delta(W)|_v,$$

where $\hat{\lambda}_v^{\mathrm{SilB}}$ is the normalization used in Silverman's book [**15**, Chapter VI]. The advantages of our normalizations are discussed in [**10**]; the crucial advantage of $\hat{\lambda}_v^{\mathrm{SilB}}$ is its independence of the chosen Weierstrass equation.

In § 5, we need to know how local heights change under isogenies.

PROPOSITION 3.1 (Bernardi [**1**]). *Let E and E' be elliptic curves defined over K_v, given by Weierstrass equations W and W', respectively. Let $\varphi\colon E \longrightarrow E'$ be an isogeny of degree n. If $Q \in E(K_v)$ satisfies $\varphi(Q) \neq 0$, then*

$$\hat{\lambda}_v(\varphi(Q)) = n\hat{\lambda}_v(Q) - \log|F_\varphi(Q)|_v - \log|m'(\varphi)|_v,$$

where

$$F_\varphi(Q) = \prod_{R \in \ker(\varphi)\backslash\{O\}} (x(Q) - x(R))$$

and

$$m'(\varphi) = \lim_{Q \to O} \frac{x(Q)}{x(\varphi(Q))}.$$

4. *Non-archimedean local error functions*

In this section, we let K be a non-archimedean local field with normalized additive valuation $v\colon K \twoheadrightarrow \mathbb{Z} \cup \{\infty\}$. Let $\mathcal{O}$ denote the valuation ring of K, let k denote the residue class field of $\mathcal{O}$ and let π be a uniformizing element of $\mathcal{O}$. Consider an elliptic curve E/K given by a Weierstrass equation W, as in (2.1), with coefficients in $\mathcal{O}$.

Given $P \in E(K)$, we choose a pair of Kummer coordinates (x_1, x_2) for P and define

$$\varepsilon(x_1, x_2) = \min\{v(\delta_1(x_1, x_2)), v(\delta_2(x_1, x_2))\} - 4\min\{v(x_1), v(x_2)\} \in \mathbb{Z}.$$

Note that ε does not depend on the choice of Kummer coordinates, so we can define $\varepsilon(P) = \varepsilon(x_1, x_2)$ for any such choice. The function ε is non-negative, bounded and continuous in the v-adic topology. Hence we can define

$$\mu(P) = \sum_{n=0}^{\infty} \frac{1}{4^{n+1}} \varepsilon(2^n P) \in \mathbb{R}. \tag{4.1}$$

It follows that μ is non-negative, bounded and continuous as well. One can show that, in fact, $\mu(P) \in \mathbb{Q}$ (compare Table 1).

Remark 2. If K is the completion of a number field at a non-archimedean place v, then $n_v \log \Phi_v(P) = -\varepsilon(P)(\log \#k)$ and $n_v \Psi_v(P) = \mu(P)(\log \#k)$ for $P \in E(K)$, where Φ_v and Ψ_v are as defined in § 3.

If we have bounds for $\varepsilon(P)$ and for the denominator of $\mu(P)$, then we can use (4.1) to compute $\mu(P)$.

Lemma 4.1. *Assume that $M \geqslant 2$ and B are non-negative integers such that:*
(1) *$M'\mu(P) \in \mathbb{Z}$ for some $0 < M' \leqslant M$; and*
(2) *$\max\{\varepsilon(P) : P \in E(K)\} \leqslant B$.*
Set

$$m = \left\lfloor \frac{\log(BM^2/3)}{\log 4} \right\rfloor.$$

Then $\mu(P)$ is the unique fraction with denominator $\leqslant M$ in the interval $[\mu_0, \mu_0 + 1/M^2]$, where

$$\mu_0 = \sum_{n=0}^{m} 4^{-n-1} \varepsilon(2^n P).$$

Proof. We know that $\mu(P)$ is a fraction with denominator bounded by M. Two distinct fractions with this property have distance greater than $1/M^2$ (here we use $M \geqslant 2$), so there

Table 1. *Non-zero values of and upper bounds α for μ for minimal Weierstrass equations.*

Type	$v(\Delta)$	μ	α
I_m	$m \geqslant 2$	$i(m-i)/m,\ i = 1, \ldots, m-1$	$m/4$
III	$\geqslant 3$	$1/2$	$1/2$
IV	$\geqslant 4$	$2/3$	$2/3$
I_m^*	$\geqslant 6 + m$	$1,\ (m+4)/4$	$(m+4)/4$
IV^*	$\geqslant 8$	$4/3$	$4/3$
III^*	$\geqslant 9$	$3/2$	$3/2$

is at most one such fraction in the given interval. On the other hand, we know that

$$\mu_0 \leqslant \mu(P) \leqslant \mu_0 + \sum_{n>m} 4^{-n-1}B = \mu_0 + \frac{B}{3 \cdot 4^{m+1}} \leqslant \mu_0 + \frac{1}{M^2}. \qquad \square$$

We now discuss how to bound $\varepsilon(P)$ and the denominator of $\mu(P)$.

LEMMA 4.2. *For $P \in E(K)$:*
(i) $0 \leqslant \mu(P) \leqslant \frac{1}{4}v(\Delta(W))$;
(ii) $0 \leqslant \varepsilon(P) \leqslant v(\Delta(W))$; *and*
(iii) *the denominator of $\mu(P)$ is bounded from above by $v(\Delta(W))$.*

Proof. Note that if $v(\Delta(W)) = 0$ and $v(x_1, x_2) = 0$, then also $v(\delta(x_1, x_2)) = 0$, since one can express a power of Δ as the resultant of δ_1 and δ_2. Therefore, if the Weierstrass equation W is minimal, then $\varepsilon(P)$ vanishes if and only if P has non-singular reduction; in particular,

$$\varepsilon(P) = 0 \Rightarrow \varepsilon(2P) = 0. \tag{4.2}$$

One can also prove (4.2) using explicit formulas which do not require minimality of W. If W is minimal, then, by the above, the value $\mu(P)$ vanishes if and only if P has non-singular reduction. More generally, it depends only on the component of the special fiber of the Néron model of E that P reduces to (see [14]). Hence the same is true for $\varepsilon(P)$, since

$$\varepsilon(P) = 4\mu(P) - \mu(2P). \tag{4.3}$$

For minimal W, the non-zero values that μ can take and an upper bound α are given in Table 1. These are taken from [10, 15].

Let $v_{\min}$ denote the minimal discriminant of E over $\mathcal{O}$. In general,

$$0 \leqslant \mu(P) \leqslant \alpha + \tfrac{1}{6}(v(\Delta(W)) - v_{\min}),$$

by [10, Proposition 8], and (i) follows from a straightforward computation. Because of (4.3), this also proves (ii). By the proof of [10, Proposition 8], a transformation from one integral Weierstrass equation to another does not change $\mu(P)$ mod $\mathbb{Z}$, so (iii) follows from Table 1. $\square$

Lemmas 4.1 and 4.2 lead to an algorithm for the computation of $\mu(P)$. A pair (x_1, x_2) of Kummer coordinates for P is said to be *primitive* if $\min\{v(x_1), v(x_2)\} = 0$. Recall that π denotes a uniformizer of K.

ALGORITHM 1.

```
1. Set $B := v(\Delta)$.
2. If $B \leqslant 1$, then return 0. Otherwise set $m := \lfloor \log(B^3/3)/\log 4 \rfloor$.
3. Set $\mu_0 := 0$. Let $(x_1, x_2)$ be primitive Kummer coordinates for $P$ with $(m+1)B + 1$
   $v$-adic digits of precision.
4. For $n := 0$ to $m$ do:
    a. Compute $(x_1', x_2') := \delta(x_1, x_2)$ (to $(m+1)B + 1$ $v$-adic digits of precision).
    b. Set $\ell := \min\{v(x_1'), v(x_2')\}$.
    c. If $\ell = 0$, then return $\mu_0$.
    d. Set $\mu_0 := \mu_0 + 4^{-n-1}\ell$.
    e. Set $(x_1, x_2) := \pi^{-\ell}(x_1', x_2')$.
5. Return the unique fraction with denominator at most $B$ in the interval $[\mu_0, \mu_0 + 1/B^2]$.
```

We now show that the algorithm is correct and estimate its running time.

PROPOSITION 4.3. *Algorithm 1 computes $\mu(P)$. Its running time is*

$$\ll (\log v(\Delta))\,\mathsf{M}((\log v(\Delta))v(\Delta)(\log \#k))$$

as $v(\Delta) \to \infty$, with an absolute implied constant.

Proof. If $B = v(\Delta) \leqslant 1$, then $\mu = \varepsilon = 0$, by Table 1. Otherwise, the loop in step 4 computes the sum in Lemma 4.1 (where now $M = B \geqslant 2$). When $\ell = 0$ in step 4c, then $\varepsilon(2^{n'}P) = 0$ for all $n' \geqslant n$ by (4.2), and hence the infinite sum defining μ is actually a finite sum and equals μ_0. (This step could be left out without affecting the correctness or the worst-case complexity of the algorithm.) Lemma 4.2 shows that B is an upper bound for ε and that $M = B$ is an upper bound for the denominator of μ. So the algorithm computes $\mu(P)$, provided the precision of $(m+1)B+1$ v-adic digits is sufficient. For this, note that the precision loss at each duplication step is given by $\varepsilon(2^nP)$ and is therefore bounded by B. So, after at most $m+1$ steps in the loop, the resulting (x_1, x_2) still has at least one digit of precision.

It remains to estimate the running time. We assume that elements of $\mathcal{O}$ are represented as truncated power series in π, whose coefficients are taken from a complete set of representatives for the residue classes. Operations on these coefficients can be performed in time $\ll \mathsf{M}(\log \#k)$. Then steps b to e in the loop take negligible time compared with step a, which involves a fixed number of additions and multiplications of elements given to a precision of $(m+1)B+1$ digits, leading to a complexity of

$$\ll \mathsf{M}(((m+1)B+1)(\log \#k))$$

operations for each pass through the loop. The total running time is therefore

$$\ll (m+1)\,\mathsf{M}(((m+1)B+1)(\log \#k)) \ll (\log v(\Delta))\,\mathsf{M}((\log v(\Delta))v(\Delta)(\log \#k))$$

as $v(\Delta) \to \infty$. □

REMARK 3. We stress that our algorithm does not require W to be minimal. If we know that W is minimal, then Silverman's algorithm [14, § 5], which only involves the computation of the valuations of a bounded number of polynomials in the coefficients of W and the coordinates of P, can be used to compute $\mu(P)$.

5. *Archimedean local heights*

Let K be an archimedean local field with valuation v. The following methods have been proposed for the computation of the local height $\hat{\lambda} = \hat{\lambda}_v$ on an elliptic curve E/K, given by a Weierstrass equation (2.1):

- an elegant series approach due to Tate and modified by Silverman [14];
- a more complicated series approach based on theta functions (see [9, Algorithm 7.5.7]);
- an algorithm based on the arithmetic geometric mean (AGM) and 2-isogenies introduced by Bost and Mestre in an unpublished manuscript [6], which currently requires v to be real (see also Bradshaw's PhD thesis [7]).

Tate's series converges linearly. The theta series has a better rate of convergence and is also faster in practice if the elliptic integrals arising in the algorithm are computed using the AGM. If v is real and one is interested in high precision, then the method of Bost and Mestre is preferable, as it converges quadratically. We now describe this algorithm and provide a complexity analysis. Let v be real and let $|\cdot|$ denote the usual absolute value on $K = \mathbb{R}$. We want to compute $\hat{\lambda}(P)$ for a point $P \in E(\mathbb{R})$; for simplicity, we only consider the case $2P \neq O$. Note that the function μ considered in [6] satisfies $\mu = \frac{1}{2}\hat{\lambda}$.

Applying a transformation, we may assume that E is given by a Weierstrass equation

$$W\colon y^2 = x(x^2 + ux + v),$$

where $u, v \in \mathbb{R}$. If all points of order two on E are real, then we set $E_0 = E$. Otherwise, consider the isogeny $E \to E_0$ defined by

$$(x, y) \mapsto \left(\frac{x^2 + ux + v}{x}, y\frac{x^2 - v}{x^2}\right), \tag{5.1}$$

where now E_0 has full 2-torsion over $\mathbb{R}$ and is given by the Weierstrass equation

$$y^2 = x(x^2 - 2ux + u^2 - 4v).$$

By Proposition 3.1, it suffices to compute the local height of the image of P on E_0 to obtain $\hat{\lambda}(P)$. For the algorithm, we need a Weierstrass equation

$$W_0\colon y^2 = x(x + a_0^2)(x + b_0^2)$$

for E_0, where $0 < b_0 < a_0 \in \mathbb{R}$. We may assume that P lies on the connected component $E_0^0(\mathbb{R})$ of the identity; if not, we can apply the algorithm to $2P \in E_0^0(\mathbb{R})$ and compute $\hat{\lambda}(P)$ using

$$\hat{\lambda}(2P) = 4\hat{\lambda}(P) - \log|2y(P)|. \tag{5.2}$$

We define the AGM sequences (a_n) and (b_n) by

$$a_n = \frac{a_{n-1} + b_{n-1}}{2}, \quad b_n = \sqrt{a_{n-1}b_{n-1}},$$

and we let $M(a_0, b_0)$ denote their common limit. For $n \geqslant 1$ we recursively define an elliptic curve E_n over the reals by the Weierstrass equation

$$W_n\colon y^2 = x(x + a_n^2)(x + b_n^2),$$

and we define a 2-isogeny $\varphi_{n-1}\colon E_n \to E_{n-1}$ by

$$(x, y) \longmapsto \left(\frac{x(x + b_n^2)}{x + a_n^2}, y\frac{(x + a_{n-1}a_n)(x + b_{n-1}a_n)}{(x + a_n^2)^2}\right).$$

Then the sequence of curves $(E_n)_n$ converges to a singular cubic curve E_∞ with equation

$$W_\infty\colon y^2 = x(x + M(a_0, b_0)^2)^2.$$

Moreover, the sequence of isogenies $(\varphi_n)_n$ converges to the identity map on $E_\infty(\mathbb{R})$.

Now let $\hat{\lambda}_n$ denote the local height on $E_n(\mathbb{R})$. Then Proposition 3.1 asserts that

$$\hat{\lambda}_{n-1}(\varphi_{n-1}(P_n)) = 2\hat{\lambda}_n(P_n) - \log(x(P_n) + a_n^2), \tag{5.3}$$

whenever we have $x(\varphi_{n-1}(P_n)) \neq 0$.

Bost and Mestre use (5.3) to give a formula for $\hat{\lambda}(P)$. Note that φ_{n-1} maps $E_n(\mathbb{R})$ onto the connected component $E_{n-1}^0(\mathbb{R})$ and that points on $E_{n-1}^0(\mathbb{R})$ always have a unique preimage on $E_n^0(\mathbb{R})$ under φ_{n-1}. Setting $P_0 = P$, we therefore get a well-defined sequence of preimages $P_n = (x_n, y_n) \in E_n^0(\mathbb{R})$, which converges to a point $P_\infty = (x_\infty, y_\infty) \in E_\infty(\mathbb{R})$. Here x_n can be calculated using

$$x_n = \tfrac{1}{2}\left(x_{n-1} - a_{n-1}b_{n-1} + \sqrt{(x_{n-1} + a_{n-1}^2)(x_{n-1} + b_{n-1}^2)}\right).$$

From (5.3), we deduce that

$$\hat{\lambda}(P) = \hat{\lambda}_0(P) = \log \lim_{n\to\infty} \frac{(x_n + a_n^2)^{2^{n-1}}}{\prod_{m=1}^{n-1}(x_m + a_m^2)^{2^{m-1}}},$$

or, equivalently,

$$\hat{\lambda}(P) = \log(x_1 + a_1^2) + \sum_{n=1}^{\infty} 2^n \log \frac{x_{n+1} + a_{n+1}^2}{x_n + a_n^2}. \tag{5.4}$$

Because of the quadratic convergence of the AGM, these formulas can be used to compute $\hat{\lambda}(P)$ to an accuracy of 2^{-d} in $\ll \log(d + \log \|W\|)$ steps. This has already been shown by Bradshaw (see [7, § 6.1]). We give a slightly different error estimate. First, note that

$$a_n - b_n \leqslant 2^{1-2^n}(a_0 - b_0).$$

Because $x_n > 0$ and $0 < b_0 < b_n < a_n$, this implies that

$$\frac{a_n^2 - b_n^2}{x_n + a_n^2} \leqslant 2^{1-2^n}(a_0 - b_0)\frac{a_n + b_n}{x_n + a_n^2} \leqslant 2^{2-2^n}\left(\frac{a_0}{b_0} - 1\right). \tag{5.5}$$

Now set

$$s_n := 1 - \frac{x_{n+1} + a_{n+1}^2}{x_n + a_n^2} \quad \text{and} \quad \vartheta := \frac{a_0}{b_0} - 1 + \sqrt{\frac{a_0}{b_0} - 1}.$$

Then $0 < s_n < 1$ and $\vartheta \ll \|W\|$. The sequence s_n converges rapidly to zero for large n, since (5.5) implies that

$$\begin{aligned} s_n &= \left| \frac{1}{2}\left(\sqrt{\frac{x_n + b_n^2}{x_n + a_n^2}} + \frac{x_n + ((a_n^2 + b_n^2)/2)}{x_n + a_n^2} \right) - 1 \right| \\ &\leqslant \frac{1}{2}\left| \sqrt{\frac{x_n + b_n^2}{x_n + a_n^2}} - 1 \right| + \frac{1}{2}\left| \frac{x_n + ((a_n^2 + b_n^2)/2)}{x_n + a_n^2} - 1 \right| \\ &\leqslant \frac{1}{2}\sqrt{\frac{a_n^2 - b_n^2}{x_n + a_n^2}} + \frac{1}{4}\left(\frac{a_n^2 - b_n^2}{x_n + a_n^2} \right) \\ &\leqslant 2^{-2^{n-1}}\sqrt{\frac{a_0}{b_0} - 1} + 2^{-2^n}\left(\frac{a_0}{b_0} - 1 \right) \\ &\leqslant 2^{-2^{n-1}}\vartheta\,. \end{aligned} \tag{5.6}$$

In particular, $s_n \leqslant \frac{1}{2}$ for $n \geqslant \log_2(\log_2 \vartheta + 1) + 1$, so that $|\log(1 - s_n)| \leqslant 2s_n$ for such n. We can use this to bound the tail of the series in (5.4). Namely,

$$\begin{aligned} \left| \sum_{n=N+1}^{\infty} 2^n \log \frac{x_{n+1} + a_{n+1}^2}{x_n + a_n^2} \right| &\leqslant \sum_{n=N+1}^{\infty} 2^n |\log(1 - s_n)| \\ &\leqslant \vartheta \sum_{n=N+1}^{\infty} 2^{1+n-2^{n-1}} \end{aligned}$$

if $N \geqslant \log_2(\log_2 \vartheta + 1)$. For $n \geqslant 4$, $n - 2^{n-1} \leqslant -2^{n-2}$, so

$$\left| \sum_{n=N+1}^{\infty} 2^n \log \frac{x_{n+1} + a_{n+1}^2}{x_n + a_n^2} \right| \leqslant \vartheta \sum_{n=N+1}^{\infty} 2^{1-2^{n-2}} \leqslant 2^{2-2^{N-1}}\vartheta \tag{5.7}$$

follows, provided $N \geqslant \max\{3, \log_2(\log_2 \vartheta + 1)\}$.

Having computed $\hat{\lambda}(P)$ for $P \in E(\mathbb{R})$, we get $\Psi_\infty(P)$ from

$$\Psi_\infty(P) = \log \max\{1, |x(P)|\} - \hat{\lambda}(P). \tag{5.8}$$

PROPOSITION 5.1. *The algorithm above computes $\Psi_\infty(P)$ to d bits of precision in time*

$$\ll \log(d + \log \|W\|)^2 \, \mathsf{M}(d + \log \|W\|).$$

Proof. First, suppose that we have already computed a_0, b_0 and x_0 and that P lies on the connected component $E_0^0(\mathbb{R})$. By (5.4) and (5.7),

$$\left| \hat{\lambda}(P) - \log(x_1 + a_1^2) - \sum_{n=1}^{N} 2^n \log \frac{x_{n+1} + a_{n+1}^2}{x_n + a_n^2} \right| \leqslant 2^{-d}$$

for

$$N = \max\{3, \log_2(d + 2 + \log_2 \vartheta) + 1\} \ll \log(d + \log \|W\|).$$

For every $n \leqslant N$, we have to apply a fixed number of additions, multiplications and square roots to compute a_{n+1}, b_{n+1} and x_{n+1}, which can be done to d' bits of precision in time $\ll \mathsf{M}(d')$, and we have to compute $\log(1 - s_n)$. Because of precision loss due to the multiplication by 2^n, we need to compute $\log(1 - s_n)$ to an additional n bits, so we need an initial precision of

$$d + N \ll d + \log(d + \log \|W\|)$$

bits. A logarithm can be computed to d' bits of precision in time $\ll (\log d')\, \mathsf{M}(d')$ using one of several quadratically converging algorithms based on the AGM (see [4, Chapter 7]). Therefore, and by (5.6), we can compute $\log(1 - s_n)$ to $d + n$ bits of precision in time

$$\ll \log(d + \log(d + \log \|W\|))\, \mathsf{M}(d + \log(d + \log \|W\|)).$$

The computation of $\log(x_1 + a_1^2)$ to d bits of precision takes time

$$\ll \log(d + \log \|W\|)\, \mathsf{M}(d + \log \|W\|).$$

Hence, given a_0, b_0 and x_0 to $d + N$ bits of precision, we can compute $\hat{\lambda}(P)$ to d bits of precision in time

$$\ll \log(d + \log \|W\|) \cdot (\mathsf{M}(d + \log \|W\|) + \log(d + \log(d + \log \|W\|))\, \mathsf{M}(d + \log(d + \log \|W\|))).$$

We can then find $\Psi_\infty(P)$ using (5.8) in time $\ll \log(d)\, \mathsf{M}(d)$, which is negligible.

To compute a_0, b_0 and x_0 from a given Weierstrass equation, we need to find the roots of at most two polynomials of degree $\leqslant 3$ with real coefficients, transform the corresponding Weierstrass equation and find the image of our point P under these transformations. The roots of a polynomial of fixed degree to d' bits of precision can be found in time $\ll \mathsf{M}(d')$ (see [4, Theorem 6.4]); the same holds for the evaluation of a polynomial of fixed degree. To counter loss of precision, we start with an initial precision of $\ll d + \log \|W\| + \log(d + \log \|W\|)$ bits, so we can compute a_0, b_0 and x_0 to $d + N$ bits of precision in time

$$\ll \mathsf{M}(d + \log \|W\| + \log(d + \log \|W\|)),$$

which is dominated by the complexity of the remaining parts of the algorithm. The logarithmic correction terms coming from (5.2) and from Proposition 3.1 applied to the isogeny (5.1) and to the change of model needed to find W_0 can be computed to a sufficient number of bits of precision in time $\ll \log(d + \log \|W\|)\, \mathsf{M}(d + \log \|W\|)$. Hence the result follows. □

REMARK 4. For large n, computing $\log(1 - s_n)$ using an AGM-based algorithm might be less efficient than using a power series such as

$$\log x = 2 \sum_{k=0}^{\infty} \frac{1}{2k+1} \left(\frac{x-1}{x+1} \right)^{2k+1}.$$

The reason is that, by (5.6), the numbers $1 - s_n$ are very close to 1, so only few terms of the power series have to be computed.

6. *Computing the canonical height of rational points*

We will combine the results of §§ 4 and 5 into an efficient algorithm for computing the canonical height of a point P on an elliptic curve E over a number field, which will prove Theorem 1.1. For simplicity, we take this number field to be $\mathbb{Q}$ in the following. We assume that our curve is given by a Weierstrass equation (2.1) W with coefficients in $\mathbb{Z}$, but we make no minimality assumption.

One usually computes $\hat{h}(P)$ using the decomposition (3.2) into local heights $\hat{\lambda}_v(P)$. The local height $\hat{\lambda}_\infty(P)$ can be computed using the algorithm of Bost–Mestre discussed in § 5 or one of the other approaches mentioned there. If the factorization of $\Delta(W)$ is known, we can use [**14**, § 5] to compute the local heights $\hat{\lambda}_p(P)$ efficiently. Alternative, but less efficient, algorithms can be found in [**17**, **18**]. If we know that W is minimal (for which some factorization is required; see the introduction), then we can use [**16**] to compute $\sum_p \hat{\lambda}_p(P)$ without factorizing $\Delta(W)$. Another approach to computing $\hat{h}(P)$ without factorization is discussed in [**11**], but their method does not yield a polynomial-time algorithm.

Our goal is to devise an algorithm for the computation of $\hat{h}(P)$ that runs in time that is quasi-linear in $\log \|W\|$, $h(P)$ along with the required precision d, which is measured in bits after the binary point. We note that $h(P)$ is the logarithm of a rational number, so it can be computed in time $\ll \log(h(P) + d)\, \mathsf{M}(h(P) + d)$. In the previous section, we showed that there is a quasi-linear algorithm for the computation of $\Psi_\infty(P)$ (see Proposition 5.1).

It remains to see how the total contribution

$$\Psi^{\mathrm{f}}(P) := \sum_p \Psi_p(P) = \sum_p \mu_p(P) \log p$$

coming from the local error functions at finite places can be computed efficiently; here we write μ_p for the local height correction function over $\mathbb{Q}_p$, as in Definition 4.1.

Fix $P \in E(\mathbb{Q})$. We assume that $(x_1, x_2) \in \mathbb{Z}^2$ is a primitive (that is, $\gcd(x_1, x_2) = 1$) pair of Kummer coordinates for P. We set $g_n = \gcd(\delta(x_1^{(n)}, x_2^{(n)}))$, where $(x_1^{(n)}, x_2^{(n)}) \in \mathbb{Z}^2$ is a primitive pair of Kummer coordinates for $2^n P$. Then the definition of μ_p implies that

$$\Psi^{\mathrm{f}}(P) = \sum_{n=0}^{\infty} 4^{-n-1} \log g_n.$$

See [**12**] for a related approach in genus 2. By Lemma 4.2, we know that each g_n divides $\Delta(W)$. The key observation is that $\Psi^{\mathrm{f}}(P)$ is a rational linear combination of logarithms of positive integers, which can be computed exactly as follows.

ALGORITHM 2.

```
1. Set $(x_1', x_2') := \delta(x_1, x_2)$, $g_0 := \gcd(x_1', x_2')$ and $(x_1, x_2) := (x_1'/g_0, x_2'/g_0)$.
2. Set $D := \gcd(\Delta(W), g_0^\infty)$ and $B := \lfloor \log D / \log 2 \rfloor$.
3. If $B \leqslant 1$, then return 0. Otherwise set $m := \lfloor \log(B^5/3)/\log 4 \rfloor$.
```

```
4. For $n := 1$ to $m$ do:
   a. Compute $(x_1', x_2') := \delta(x_1, x_2) \bmod D^{m+1} g_0$.
   b. Set $g_n := \gcd(D, \gcd(x_1', x_2'))$ and $(x_1, x_2) := (x_1'/g_n, x_2'/g_n)$.
5. Use Bernstein's algorithm from [2] to compute a sequence $(q_1, \ldots, q_r)$ of pairwise coprime
   positive integers such that each $g_n$ (for $n = 0, \ldots, m$) is a product of powers of the $q_i$:
   $g_n = \prod_{i=1}^r q_i^{e_{i,n}}$.
6. For $i := 1$ to $r$ do:
   a. Compute $a := \sum_{n=0}^m 4^{-n-1} e_{i,n}$.
   b. Let $\mu_i$ be the simplest fraction between $a$ and $a + 1/B^4$.
7. Return $\sum_{i=1}^r \mu_i \log q_i$, a formal linear combination of logarithms.
```

PROPOSITION 6.1. *The preceding algorithm computes $\Psi^{\mathrm{f}}(P)$ in time*

$$\ll (\log\log D)^2 \,\mathsf{M}((\log\log D)(\log D)) + \mathsf{M}(h(P))(\log h(P)).$$

Proof. We note that if $B \leqslant 1$ in step 3, then either $g_0 = 1$ and $\Psi^{\mathrm{f}}(P) = 0$, or $D \in \{2, 3\}$. In the latter case, g_0 is a power of $p = 2$ or 3 and $v_p(\Delta(W)) = 1$, which would imply that $\varepsilon_p(P) = 0$, so $g_0 = 1$, and we get a contradiction.

Let p be a prime. If $p \nmid g_0$, then $\varepsilon_p(P) = 0$ and therefore $\mu_p(P) = 0$. So we now assume that p divides g_0. We have $v_p(\Delta(W)) = v_p(D) \leqslant B$. We see that $p^{(m+1)v_p(D)+1}$ divides $D^{m+1} g_0$, so computing modulo $D^{m+1} g_0$ will provide sufficient p-adic accuracy so that $v_p(g_n) = \varepsilon_p(2^n P)$ for all $n \leqslant m$ (compare the proof of Proposition 4.3 above). (One could replace $D^{m+1} g_0$ by $D^{m+1-n} g_0$ in step 4a.) Since all the g_n are power products of the q_i, there will be exactly one $i = i(p)$ such that $p \mid q_{i(p)}$; let $b_p = v_p(q_{i(p)})$. Then

$$\sum_{n=0}^m 4^{-n-1} \varepsilon_p(2^n P) = \sum_{n=0}^m 4^{-n-1} v_p(g_n) = b_p \sum_{n=0}^m 4^{-n-1} e_{i(p),n} = b_p a,$$

so

$$\mu_p(P) = \sum_{n=0}^\infty 4^{-n-1} \varepsilon_p(2^n P) = b_p a + \sum_{n=m+1}^\infty 4^{-n-1} \varepsilon_p(2^n P),$$

where the last sum is in $[0, 1/B^4]$ (this follows from $0 \leqslant \varepsilon_p \leqslant B$; see Lemma 4.2 and the definition of m, and compare the proof of Lemma 4.1). We know that the denominator of $\mu_p(P)$ is at most B (see Lemma 4.2), so the denominator of $\mu_p(P)/b_p$ is at most B^2, since $b_p \leqslant v_p(D) \leqslant B$. On the other hand, $a \leqslant \mu_p(P)/b_p \leqslant a + 1/(b_p B^4) \leqslant a + 1/B^4$, which implies that $\mu_p(P)/b_p$ is the unique fraction in $[a, a + 1/B^4]$ with denominator at most B^2, so $\mu_p(P)/b_p = \mu_{i(p)}$, by Step 6b. Now

$$\sum_p \mu_p(P) \log p = \sum_p \mu_{i(p)} b_p \log p = \sum_{i=1}^r \mu_i \sum_{p \mid q_i} b_p \log p = \sum_{i=1}^r \mu_i \log q_i.$$

It remains to estimate the running time. The computation of $\delta(x_1, x_2)$ can be done in time $\ll \mathsf{M}(h(P))$; the same is true for the division in step 1. The gcd in step 1 can be computed in time $\ll \mathsf{M}(h(P)) \log h(P)$. The computations in steps 2 and 3 take negligible time compared with step 4. Each pass through the loop in step 4 takes time $\ll \mathsf{M}((m+2)\log D) \log((m+2)\log D)$, so the total time for the loop is

$$\ll m\, \mathsf{M}(m(\log D)) \log(m \log D) \ll (\log\log D)^2 \,\mathsf{M}((\log\log D)(\log D)).$$

The algorithm in [2] computes suitable q_i for a pair a, b of positive integers in time $\ll (\log ab)(\log\log ab)^2$. We iterate this algorithm, applying it first to g_0 and g_1, then to each of the resulting q_i and g_2 and so on. Note that $g_n \leqslant D$ for all n. Because there are always $\ll \log D$ terms in the sequence of q_i, this leads to a contribution of $\ll \log D(\log\log D)^3$ for step 5. This is dominated by the time for the loop. The remaining steps take negligible time. $\square$

In practice, the efficiency of this approach can be improved as follows.

- We trial factorize $\Delta(W)$ up to some bound T to split off the contributions of all sufficiently small primes p. We can then compute the corresponding μ_p using the algorithm of Proposition 4.3 or the algorithm of [14] (see Remark 3). In step 3, we can then set $B := \lfloor \log D'/\log T \rfloor$, where D' is the unfactorized part of D. Note that, in practice, the trial division can take quite a bit more time than it saves, in particular, when the equation has large coefficients, so this modification should be used with care.
- We update our list of 'building blocks' q_i after each pass through the loop in step 4 using the new g_n; we do the computation modulo suitable powers of the q_i instead of modulo $D^{m+1}g_0$. We can also use separate values of B and m for each q_i, which will usually be smaller than those given above.
- In this way, we can integrate steps 4, 5 and 6 into one loop.
- We can replace B^5 in the definition of m by $2B^4$. Then $\mu_p(P) \leqslant b_p a + 1/(2B^3)$ and $a \leqslant \mu_p(P)/b_p \leqslant a + 1/(2b_p B^3)$. If $\mu_p(P)/b_p = r/s$ with $s \leqslant Bb_p$, then we have the bound $a \leqslant r/s \leqslant a + 1/(2sB^2) \leqslant a + 1/(2s^2)$. There can be at most one fraction r/s with $s \leqslant B^2$ satisfying this: if r'/s' is another such fraction, then

  $$\frac{1}{ss'} \leqslant \left|\frac{r}{s} - \frac{r'}{s'}\right| \leqslant \frac{1}{2\min\{s, s'\}B^2},$$

  which leads to the contradiction $\max\{s, s'\} \geqslant 2B^2$. We can then find $\mu_i = \mu_p(P)/b_p$ as the first convergent r/s of the continued fraction expansion of a that is $\geqslant a$ and satisfies $r/s \leqslant a + 1/(2sB^2)$.

Combining Propositions 5.1 and 6.1, we finally obtain an efficient algorithm for computing the canonical height $\hat{h}(P)$ of a point $P \in E(\mathbb{Q})$.

Proof of Theorem 1.1. The first term is the time needed to compute $h(P)$. The second term comes from the complexity bound for the computation of $\Psi^f(P)$ (using $\log D \ll \log \|W\|$) from Proposition 6.1. The third term is the bound for the computation of $\Psi^\infty(P)$ given in Proposition 5.1.

7. *Implementation and Examples*

We have implemented our algorithm using the computer algebra system Magma [5]. In the current implementation, the factorization into coprimes in the algorithm preceding Proposition 6.1 uses a relatively simple algorithm due to Buchmann and Lenstra [8, Proposition 6.5] instead of the faster algorithm of [2] (or of [3]). In practice, the running time of this part of the algorithm appears to be negligible.

Let us compare our implementation to Magma's built-in command `CanonicalHeight` (version 2.21-2). The latter uses the method of Bost–Mestre for the computation of the archimedean local height. For the finite part of the height, a globally minimal Weierstrass model is computed. The non-archimedean contributions are then computed separately using the algorithm from [14]; the relevant primes are found by factorizing $\gcd(\delta_1(x_1, x_2), \delta_2(x_1, x_2))$, where (x_1, x_2) is a primitive pair of Kummer coordinates for a point P. The same strategy is currently used in `Pari/GP`. The computer algebra system `Sage` contains an implementation of, essentially, Silverman's original algorithm for the computation of canonical heights from [14]; in particular, it factorizes the discriminant.

EXAMPLE 1. Consider the family E_a of curves given by the Weierstrass equation

$$W_a \colon y^2 = x^3 - ax + a,$$

where a is an integer, and the non-torsion point $P = (1, 1)$ on E_a. To compute $\hat{h}(P)$, Magma needs to find a globally minimal model for E_a, which boils down to deciding whether a sixth power of a prime divides a. Hence, for random integers a of large absolute value, the Magma implementation becomes slow. For instance, taking a to be 5340200419833800017985460942490398389444339691251749039558531515293241873258929634112121245344691478, which has 100 digits and is of the form $a = 2 \cdot 37 \cdot a'$ with a' composite, Magma's built-in `CanonicalHeight` takes about an hour, but our implementation needs only 0.001 s to compute $\hat{h}(P)$ to 30 decimal digits of precision. For these computations, and the ones below, we used a Dell Latitude E7440 Laptop with 8 GB of memory and an i5-4300U CPU with two cores having 1.9 GHz each. For a equal to 1156498933847959533988831879398816130438976947840284525292584250252938021952046963963000864858057914442064403481185654247216831580683337015346748079666961851320095362381105272849374580830071701975985 0, which has 200 digits and factorizes as $a = 2 \cdot 3^2 \cdot 5^2 \cdot a'$ with a' composite, the computation of $\hat{h}(P)$ using our implementation takes 0.003 s, whereas Magma needs about 5 h and 30 min.

Finally, we look at the 500-digit number $a =$ 28276805523181086329328141188416415606304708589734778175789716618240877758692981130319935379836208245099552401602995135086123374392032954117627785768748616863628083464269023575658346783517541505391502873826466503688549496039448522504993529003411479688448361012236852968621731549025539014813988793465901530315058422265303601784166137772255014978074155871467151125861241065343517294351129616001349317877081170285257729773270941059335530220433045635898507554473398924420918799034729911478550230429211184, which factorizes as the product $a = 2^4 \cdot 23 \cdot 71 \cdot a'$ with a' composite. Our implementation needs 0.009 s to compute $\hat{h}(P)$; Magma's `CanonicalHeight` did not terminate in six weeks. For this a, the computation of the canonical height of $50P$, which has naive height $h(50P) \approx 1437536.77$, took 0.215 s, whereas it took Magma 2.83 s to even compute $50P$!

For random a having 5000 digits, the computation of $\hat{h}(P)$ to the standard precision of 30 decimal digits usually takes about 0.7 s. Our implementation is usually faster than `CanonicalHeight` if a has at least 18 decimal digits. Note that in contrast to our implementation, the Magma implementation of the algorithm of Bost–Mestre for $\hat{\lambda}_\infty$ is written in C.

The family E_a has no special properties or applications that we are aware of; it was merely chosen for testing purposes.

Acknowledgements. We would like to thank Jean-François Mestre for providing us with a copy of the unpublished manuscript [6], Mark Watkins for answering our questions about the computation of canonical heights in Magma, and Elliot Wells for pointing out an inaccuracy in Algorithm 2 and the complexity analysis in Proposition 6.1, and Marco Caselli for pointing out an inaccuracy in the statement of Proposition 3.1.

References

1. D. BERNARDI, 'Hauteur p-adique sur les courbes elliptiques', *Seminar on number theory,* Paris 1979–1980, Progress in Mathematics 12 (Birkhäuser, Boston, MA, 1981) 1–14.
2. D. J. BERNSTEIN, 'Research announcement: Faster factorization into coprimes', Preprint, 2004.
3. D. J. BERNSTEIN, 'Factoring into coprimes in essentially linear time', *J. Algorithms* 54 (2005) 1–30.
4. J. M. BORWEIN and P. B. BORWEIN, *Pi and the AGM*, Canadian Mathematical Society Series of Monographs and Advanced Texts 4 (John Wiley & Sons, Inc., New York, 1998).
5. W. BOSMA, J. CANNON and C. PLAYOUST, 'The Magma algebra system. I. The user language', *J. Symbolic Comput.* 24 (1997) no. 3–4, 235–265.
6. J.-B. BOST and J.-F. MESTRE, Calcul de la hauteur archimédienne des points d'une courbe elliptique par un algorithme quadratiquement convergent et application au calcul de la capacité de l'union de deux intervalles, unpublished manuscript, 1993.

7. R. W. Bradshaw, 'Provable computation of motivic L-functions', PhD Thesis, University of Washington, 2010.
8. J. A. Buchmann and H. W. Lenstra Jr, 'Approximating rings of integers in number fields', *J. Théor. Nombres Bordeaux* 6 (1994) no. 2, 221–260.
9. H. Cohen, *A course in computational algebraic number theory* (Springer, Berlin, Heidelberg, New York, 1993).
10. J. Cremona, M. Prickett and S. Siksek, 'Height difference bounds for elliptic curves over number fields', *J. Number Theory* 116 (2006) no. 1, 42–68.
11. G. Everest and T. Ward, 'The canonical height of an algebraic point on an elliptic curve', *New York J. Math.* 6 (2000) 331–342.
12. E. V. Flynn and N. P. Smart, 'Canonical heights on the Jacobians of curves of genus 2 and the infinite descent', *Acta Arith.* 79 (1997) no. 4, 333–352.
13. A. Néron, 'Quasi-fonctions et hauteurs sur les variétés abéliennes', *Ann. of Math.* (2) 82 (1965) 249–331.
14. J. H. Silverman, 'Computing heights on elliptic curves', *Math. Comp.* 51 (1988) no. 183, 339–358.
15. J. H. Silverman, *Advanced topics in the arithmetic of elliptic curves*, Graduate Texts in Mathematics 151 (Springer, New York, 1994).
16. J. H. Silverman, 'Computing canonical heights with little (or no) factorization', *Math. Comp.* 66 (1997) no. 218, 787–805.
17. H. M. Tschöpe and H. G. Zimmer, 'Computation of the Néron–Tate height on elliptic curves', *Math. Comp.* 48 (1987) no. 177, 351–370.
18. H. G. Zimmer, 'A limit formula for the canonical height of an elliptic curve and its application to height computations', *Number theory,* Banff, AB (de Gruyter, Berlin, 1990) 641–659.

J. Steffen Müller
Institut für Mathematik
Carl von Ossietzky Universität Oldenburg
26111 Oldenburg
Germany

jan.steffen.mueller@uni-oldenburg.de

Michael Stoll
Mathematisches Institut
Universität Bayreuth
95440 Bayreuth
Germany

Michael.Stoll@uni-bayreuth.de