EFFICIENT COMPUTATION OF p-ADIC HEIGHTS

DAVID HARVEY

Abstract

We analyse and drastically improve the running time of the algorithm of Mazur, Stein and Tate for computing the canonical cyclotomic p-adic height of a point on an elliptic curve $E/\mathbf{Q}$, where E has good ordinary reduction at $p \geqslant 5$.

1. *Introduction*

Mazur, Stein and Tate [**7**] recently introduced a fast algorithm for computing the canonical, cyclotomic p-adic height $h_p(P)$ of a rational point P on an elliptic curve $E/\mathbf{Q}$ — in this paper, referred to simply as the p-adic height — in the case that E has good ordinary reduction at $p \geqslant 5$. Their algorithm finds applications in, for example, numerical investigation of phenomena related to the p-adic Birch and Swinnerton-Dyer conjectures, since the p-adic height is part of the definition of the p-adic regulator term.

They did not give a bound for the running time of their algorithm; rather, the point of their paper is that the p-adic height can be computed feasibly at all, to reasonably high p-adic precision. In this paper we sketch an estimate for the running time of the algorithm of [**7**], and we give several improvements that drastically improve the asymptotic running time, both for large p and for high precision. With a contemporary desktop computer, it becomes straightforward to calculate a handful of p-adic digits of the height when p is around 10^{11}; and in the other direction, for small primes, one easily obtains thousands of p-adic digits.

We also carefully analyse the amount of p-adic precision that must be maintained throughout the calculation to guarantee a given number of correct digits of output; this was not discussed in much detail in [**7**]. Obtaining a sharp bound is particularly important when p is large, since computing additional unnecessary digits in intermediate calculations becomes very expensive.

In our running time estimates, we will generally ignore logarithmic factors, expressing all estimates in terms of the 'soft-oh' notation $\widetilde{O}(X)$, which means $O(X(\log X)^k)$ for some integer $k \geqslant 0$.

To state the main results, we introduce some notation. The elliptic curve $E/\mathbf{Q}$ of interest is given by the Weierstrass equation

$$y^2 + a_1 xy + a_3 y = x^3 + a_2 x^2 + a_4 x + a_6,$$

with the usual invariant differential $\omega = dx/(2y + a_1 x + a_3)$. We assume that the a_k are integers, but we do not assume that the equation is minimal. We assume that $p \geqslant 5$ and that E has good ordinary reduction at p. We treat E as fixed for the purposes of estimating running times.

The following result is proved in §3.

https://doi.org/10.1112/S1461157000000528 Published online by Cambridge University Press

LMS J. Comput. Math. **11** (2008) 40–59

THEOREM 1. *Let $N \geqslant 1$. The value of the p-adic modular form $\mathbf{E}_2(E, \omega)$ may be computed modulo p^N in time $\widetilde{O}(pN^2)$. If $p > 6N$, it may be computed in time $\widetilde{O}(p^{1/2}N^{5/2})$.*

The next result, proved in §4, concerns the p-adic sigma function $\sigma_p(t) \in \mathbf{Z}_p[[t]]$, as defined in [**8**]. This is a power series

$$\sigma_p(t) = t + c_2 t^2 + c_3 t^3 + \cdots,$$

where $t = -x/y$ is a parameter for the formal group of E. For any $r \geqslant 1$, let I_r be the ideal of $\mathbf{Z}_p[[t]]$ given by

$$I_r = (p^r, p^{r-1}t, \ldots, pt^{r-1}, t^r).$$

THEOREM 2. *Let $N \geqslant 4$. Given the value of $\mathbf{E}_2(E, \omega)$ modulo p^{N-3}, the p-adic sigma function may be computed modulo I_N in time $\widetilde{O}(N^2 \log p)$.*

Computing $\sigma_p(t)$ modulo I_N simply means that we are computing the coefficient of t^k modulo p^{N-k}, for each $k = 1, \ldots, N-1$. The running time is optimal up to logarithmic factors, since the amount of data needed to represent $\sigma_p(t)$ modulo I_N is proportional to $N^2 \log p$. Note that for $N \leqslant 3$, computing $\sigma_p(t)$ modulo I_N is trivial, since c_2 is given simply by $a_1/2$ (see §4).

Finally, we consider the problem of computing the p-adic height $h_p(P)$ modulo p^M of a point $P \in E(\mathbf{Q})$ for some $M \geqslant 2$, given $\sigma_p(t)$ as input. Following a suggestion of Christian Wuthrich, in this paper we normalise the p-adic height differently from [**7**]; our height is equal to the height of [**7**] multiplied by the factor $2p$.

Let $n_1 = \#E(\mathbf{F}_p)$, let n_2 be the least common multiple of the Tamagawa numbers of E, and let $n = \mathrm{LCM}(n_1, n_2)$. Put

$$M' = M + 2v_p(n),$$

where v_p denotes the usual additive p-adic valuation. Note that $v_p(n_1) \leqslant 1$, and we are assuming that $n_2 = O(1)$ since E is fixed, so $M' = M + O(1)$.

The following result is proved in §5. We denote by C_P the amount of data required to represent the coordinates of C_P.

THEOREM 3. *Let $P \in E(\mathbf{Q})$, and suppose that $\sigma_p(t)$ is known modulo $I_{M'+1}$. Then $h_p(P)$ may be computed modulo p^M in time $\widetilde{O}(C_P + M \log^2 p + M^2 \log p)$.*

Organisation of the paper

In §2 we outline the Mazur–Stein–Tate algorithm, and indicate the steps in their algorithm that we intend to accelerate. In §3, §4 and §5 we prove Theorems 1, 2 and 3 respectively, and give several detailed examples of the various components of the algorithm. In §6 we give an example of the whole algorithm in operation. Finally in §7 we give some examples of timings for an implementation of the algorithm.

2. *A sketch of the Mazur–Stein–Tate algorithm*

In this section we sketch the original algorithm of [**7**], and indicate the bottlenecks in the time complexity that this paper seeks to address.

The key insight of [**7**] is that it is possible to compute $\mathbf{E}_2(E,\omega)$ efficiently using Kedlaya's algorithm [**5**], and then to use the value of $\mathbf{E}_2(E,\omega)$ to deduce the p-adic sigma function. The p-adic height of a point $P \in E(\mathbf{Q})$ is then given in terms of $\sigma_p(t)$ by a simple formula. Their method for computing $\mathbf{E}_2(E,\omega)$ is discussed in the proof of Theorem 1 in §3. We now consider the other steps.

2.1. *Computing the p-adic sigma function*

The algorithm uses the fact that $\sigma_p(t)$ is the unique odd function in $\mathbf{Z}_p[[t]]$ of the form $\sigma_p(t) = t + \cdots$ satisfying the differential equation

$$x(t) + c = -\frac{d}{\omega}\left(\frac{1}{\sigma}\frac{d\sigma}{\omega}\right), \tag{1}$$

where c is the constant

$$c = \frac{a_1^2 + 4a_2 - \mathbf{E}_2(E,\omega)}{12},$$

and where $x(t) \in \mathbf{Z}_p[[t]]$ is the power series expansion of x at the origin. Since c is known from $\mathbf{E}_2(E,\omega)$, it becomes a matter of computing $x(t)$, and then solving (1) for $\sigma_p(t)$.

The series $x(t)$ is determined from an auxiliary power series $w(t) = \sum_{n\geqslant 0} s_n t^n$, using a certain recursive formula to compute the s_n. A bottleneck arises here, in that the recursive formula for s_n involves a term of the form $\sum_{i+j+k=n} s_i s_j s_k$, which requires $O(n^2)$ ring operations to evaluate. To compute the p-adic height to precision p^N, it turns out to be necessary to compute $O(N)$ coefficients, to precision about p^N each. Therefore computing $x(t)$ to the desired precision requires $O(N^3)$ ring operations, for a time cost proportional to at least $N^3 \log(p^N) = N^4 \log p$. In §4 we will give a different method for computing $x(t)$ whose running time is only $\widetilde{O}(N^2 \log p)$.

To solve (1), they first use the formal logarithm to to change variables from t to the parameter z on the *additive* group. After making this substitution, the differential equation takes the simpler form

$$x(z) + c = -\frac{d}{dz}\left(\frac{1}{\sigma(z)}\frac{d\sigma}{dz}\right),$$

which can be solved by integration and exponentation of power series. A bottleneck arises here also: to perform the change of variables, it is necessary to invoke several power series composition and reversion operations. To the author's knowledge, the Brent–Kung algorithm [**3**] is the best algorithm available for computing series reversions and compositions. It has complexity $\widetilde{O}(n^{3/2})$ in the number of terms, so when working to precision p^N the running time would be proportional to at least $N^{5/2} \log p$. In §4 we show that it is entirely unnecessary to change variables; the original equation (1) can be solved directly in time $\widetilde{O}(N^2 \log p)$.

2.2. *Computing the p-adic height*

Let $P \in E(\mathbf{Q})$ be a nonzero point. Its affine coordinates may be written uniquely in the form

$$P = (x(P), y(P)) = \left(\frac{\alpha(P)}{d(P)^2}, \frac{\beta(P)}{d(P)^3}\right),$$

where $(\alpha(P), d(P)) = (\beta(P), d(P)) = 1$ and $d(P) \geqslant 1$.

In [7], the authors first consider the case that P satisfies two conditions:

(A1) P reduces to the zero element of $E(\mathbf{F}_p)$; and

(A2) P reduces to a nonsingular point of $E(\mathbf{F}_\ell)$ for all primes ℓ at which E has bad reduction.

Condition (A1) implies that $t(P) = -x(P)/y(P)$ is divisible by p. Under these conditions, $h_p(P)$ is given by the formula

$$h_p(P) = 2\log_p\left(\frac{\sigma(P)}{d(P)}\right), \tag{2}$$

where $\log_p$ is the Iwasawa p-adic logarithm, and where $\sigma(P) = \sigma(t(P))$. (Recall that our normalisation for $h_p(P)$ differs from that of [7] by a factor of $2p$.)

To handle arbitrary nonzero $Q \in E(\mathbf{Q})$, they proceed as follows. As in Theorem 3, let $n_1 = \#E(\mathbf{F}_p)$, let n_2 be the least common multiple of the Tamagawa numbers of E, and let $n = \mathrm{LCM}(n_1, n_2)$. Then $P = nQ$ *does* satisfy (A1) and (A2), so one may use (2) to compute $h_p(P)$. From this one obtains $h_p(Q) = h_p(P)/n^2$, since h_p is a quadratic form.

The above procedure has a serious bottleneck when p is large. The difficulty is that one must actually compute the coordinates of nQ, which will generally have about n^2 digits (assuming that Q is non-torsion). From the Weil bound we have $\#E(\mathbf{F}_p) \approx p$, so n is roughly proportional to p. Therefore the time complexity is at least proportional to p^2, and for sufficiently large p will dwarf even the time required to compute $\mathbf{E}_2(E, \omega)$. In §5 we will show how to avoid this problem by judicious use of the division polynomials associated to E, achieving a running time only polylogarithmic in p.

3. Computing $\mathbf{E}_2(E, \omega)$

In this section we prove Theorem 1: given $N \geqslant 1$, we wish to compute $\mathbf{E}_2(E, \omega)$ to precision p^N.

The first step is to compute the matrix F of the absolute p-th power Frobenius on the basis $\{dx/y, x\,dx/y\}$ of the Monsky–Washnitzer cohomology of E, to precision p^N. This may be done using either Kedlaya's algorithm [5] in time $\widetilde{O}(pN^2)$, or if $p > 6N$ one may use a modification of Kedlaya's algorithm [4] that has running time $\widetilde{O}(p^{1/2}N^{5/2})$. The optimal crossover point between the two algorithms would depend on the implementations; one expects it to be roughly of the form $p = CN$ for some constant C.

As explained in [7, §3.2], the value of $\mathbf{E}_2(E, \omega)$ may be deduced from the matrix as follows. Write

$$F^N = \begin{pmatrix} A & B \\ C & D \end{pmatrix}.$$

Then

$$\mathbf{E}_2(E, \omega) = -12B/D \pmod{p^N}.$$

Note that D is a unit (see [7, §3.2]), so the result is correct modulo p^N. Using repeated squaring, computing F^N requires $O(\log N)$ matrix multiplications, each taking time $\widetilde{O}(N \log p)$, so the time for this step is only $\widetilde{O}(N \log p)$.

3.1. *A constant factor improvement*

Kedlaya's algorithm was originally designed for hyperelliptic curves, but because we are in the special setting of an elliptic curve, it is possible to obtain a constant factor speedup. Indeed, the *trace* of the matrix may be determined by any of various fast algorithms for counting $\#E(\mathbf{F}_p)$, and the determinant is known to be p. Since Kedlaya's algorithm computes the two columns of the matrix independently, the idea is simply to compute only one column, and fill in the other column from knowledge of the trace and determinant. (Alternatively, one could use the trace and determinant as a strong correctness check.) This idea was communicated to the author by William Stein, who attributes it to Eyal Goren.

The speedup will be at most a factor of two. In practice it will be less, since it only affects "Step 2" of Kedlaya's algorithm, not "Step 1" (in the language of [**5**, §4]). It only applies to Kedlaya's original algorithm; it does not apply to the algorithm of [**4**].

In practice some care is needed to avoid precision loss. A particularly badly conditioned example is given by the curve $y^2 = x^3 + 7x + 8$ at the prime $p = 11$, whose Frobenius matrix is given to precision $O(11^3)$ by

$$F = \begin{pmatrix} 11 \cdot 104 + O(11^3) & 11 \cdot 16 + O(11^3) \\ 11^2 \cdot 7 + O(11^3) & 185 + O(11^3) \end{pmatrix}.$$

If the trace and determinant are known, then the second column modulo 11^3 only determines the first column modulo 11^2, and the first column modulo 11^3 only determines the second column modulo 11.

This problem may be avoided by a simple change-of-basis argument, as follows. One first uses Kedlaya's algorithm to compute the second column only modulo p. The running time for this step is $\widetilde{O}(p)$. If the top-right entry is a unit, then no precision loss will arise; one simply uses Kedlaya's algorithm at full precision p^N to compute the second column, and this suffices to determine the first column modulo p^N from the trace and determinant.

In the unlucky event that the top-right entry is not a unit, one instead runs Kedlaya's algorithm at precision p^N with respect to the basis $\{dx/y, (1+x)dx/y\}$. In other words, one applies Kedlaya's reduction algorithm to the image under Frobenius of the differential $(1+x)dx/y$, and expresses the result as a linear combination of dx/y and $(1+x)dx/y$. The top-right entry of the matrix on this new basis is then guaranteed to be a unit, so the second column is sufficient to determine the first column modulo p^N (and the trace does not depend on the choice of basis). Conjugating by the change-of-basis matrix then yields the matrix on the original basis, to full precision.

4. *Computing the p-adic sigma function*

This section is devoted to proving Theorem 2. We assume that $N \geqslant 4$ and that $c = (a_1^2 + 4a_2 - \mathbf{E}_2)/12$ is known modulo p^{N-3}. We wish to compute $\sigma_p(t)$ modulo I_N, in time $\widetilde{O}(N^2 \log p)$.

Throughout this section we assume that asymptotically fast polynomial arithmetic is being used; specifically, that multiplication and division of polynomials of degree d over $\mathbf{Z}/p^r\mathbf{Z}$ may be accomplished in time $\widetilde{O}(dr \log p)$.

Let $x(t)$ and $y(t)$ be the power series expansions of x and y around the origin. Let $w(t) = -1/y(t)$, and let

$$s(t) = \frac{x'(t)}{2y(t) + a_1 x(t) + a_3},$$

so that $s(t)dt$ is the series expansion of the invariant differential ω. The first few terms of each series are given by

$$x(t) = t^{-2} - a_1 t^{-1} - a_2 + \cdots,$$
$$y(t) = -t^{-3} + a_1 t^{-2} + a_2 t^{-1} + \cdots,$$
$$w(t) = t^3 + a_1 t^4 + (a_1^2 + a_2)t^5 + \cdots,$$
$$s(t) = 1 + a_1 t + (a_1^2 + a_2)t^2 + \cdots.$$

(See also [9, Ch. IV], which discusses these expansions in some detail, using slightly different notation.)

Let $R = \mathbf{Z}/p^{N-3}\mathbf{Z}$. Our first task is to compute the above series over R, with $N+1$ terms each.

PROPOSITION 4. *The series $t^2 x(t)$, $t^3 y(t)$, $t^{-3} w(t)$ and $s(t)$ may be computed up to $O(t^{N+1})$, with coefficients in R, in time $\widetilde{O}(N^2 \log p)$.*

Proof. The series $w(t)$ satisfies the algebraic equation

$$w = t^3 + a_1 tw + a_2 t^2 w + a_3 w^2 + a_4 tw^2 + a_6 w^3,$$

and so it may be solved using Newton's method. We start with the initial approximation $w(t) = t^3$, and then repeatedly apply the Newton iteration

$$\begin{aligned} w' &= w - \frac{w - t^3 - a_1 tw - a_2 t^2 w - a_3 w^2 - a_4 tw^2 - a_6 w^3}{1 - a_1 t - a_2 t^2 - 2a_3 w - 2a_4 tw - 3a_6 w^2} \\ &= \frac{t^3 - a_3 w^2 - a_4 tw^2 - 2a_6 w^3}{1 - a_1 t - a_2 t^2 - 2a_3 w - 2a_4 tw - 3a_6 w^2}. \end{aligned}$$

The arithmetic is performed using truncated power series in $R[t]$. It is straightforward to check that the number of correct terms doubles with each iteration. Each iteration requires several power series multiplications and one reciprocal. As is typical in applications of Newton's method to power series, the total time to obtain n terms is a constant multiple of the time required for a length n polynomial multiplication. Therefore the total time to compute $O(N)$ terms of $w(t)$ with coefficients in R is $\widetilde{O}(N \log(p^{N-3})) = \widetilde{O}(N^2 \log p)$.

Given $t^{-3}w(t)$ up to $O(t^{N+1})$, we may then deduce $x(t) = t/w(t)$, $y(t) = -1/w(t)$ and $s(t) = x'(t)/(2y(t) + a_1 x(t) + a_3)$, also with coefficients in R, to the desired number of terms, in time $\widetilde{O}(N^2 \log p)$. $\square$

Next we consider the differential equation (1), which may be rewritten as

$$x(t) + c = \frac{-1}{s(t)} \left(\frac{\sigma_p'(t)}{\sigma_p(t)s(t)} \right)'.$$

It is convenient to rephrase this in terms of the unit power series

$$\theta(t) = t^{-1} \sigma_p(t) = 1 + \cdots \in \mathbf{Z}_p[[t]].$$

To solve for $\sigma_p(t)$ modulo I_N it suffices to solve for $\theta(t)$ modulo I_{N-1}. We have $\theta'(t)/\theta(t) = \sigma_p'(t)/\sigma_p(t) + t^{-1}$, so $\theta(t)$ satisfies the equation

$$x(t) + c = \frac{-1}{s(t)}\left(\frac{1}{s(t)}\left(\frac{-1}{t} + \frac{\theta'(t)}{\theta(t)}\right)\right)'. \tag{3}$$

Manipulating this equation formally, we obtain

$$\frac{\theta'(t)}{\theta(t)} = h(t) \qquad \in \mathbf{Z}_p[[t]], \tag{4}$$

where

$$h(t) = -\frac{1}{t} - s(t)\left(\int (x(t) + c)s(t)dt + C\right)$$

for some constant of integration C. (The formal integration operator is assumed to output a series with zero constant term.)

The constant C is determined by the condition that $\sigma_p(t)$ is an *odd* function. This condition means that $\sigma_p(i(t)) = -\sigma_p(t)$, where $i(t) = -t - a_1 t^2 + \cdots$ is the formal inverse law [**9**, IV §1]. It implies that the coefficient of t^2 in $\sigma_p(t)$ is $a_1/2$, so that the coefficient of t in $\theta(t)$ is $a_1/2$. Substituting this into (4) and using the expansions for $s(t)$ and $x(t)$ given earlier, one finds that $C = a_1/2$.

PROPOSITION 5. *The series $h(t)$ may be computed, with the coefficient of t^j known modulo $p^{N-3-\lfloor \log_p(j) \rfloor}$ for $1 \leqslant j < N$, and with the constant term known modulo p^{N-2}, in time $\widetilde{O}(N^2 \log p)$.*

In the above proposition, note that $\log_p(j)$ refers to the base p logarithm, not the p-adic logarithm.

Proof. First we compute $(x(t)+c)s(t)$. From Proposition 4 this series is known up to $O(t^{N-1})$ with coefficients in $R = \mathbf{Z}/p^{N-3}\mathbf{Z}$. Moreover, its integral has integer coefficients, since from (3) we have

$$(x(t) + c)s(t) = -\left(\frac{1}{s(t)}\left(\frac{-1}{t} + h(t)\right)\right)',$$

and both $s(t) = 1 + \cdots$ and $h(t)$ have integral coefficients. (This fact is the basis for the 'integrality algorithm' of [**7**].)

After integrating, the series

$$\int (x(t) + c)s(t)dt + \frac{a_1}{2}$$

is known up to $O(t^N)$, but the coefficient of t^j is known only modulo $p^{N-3-v_p(j)}$, because of the divisions by powers of p in the integration step. Then we must multiply by $-s(t)$ and subtract $1/t$, to obtain

$$h(t) = -\frac{1}{t} - s(t)\left(\int (x(t) + c)s(t)dt + \frac{a_1}{2}\right).$$

This series is also known up to $O(t^N)$; the effect of multiplying by $s(t)$ is to propagate the errors to higher order terms, so now the coefficient of t^j is known only modulo $p^{N-3-\lfloor \log_p(j) \rfloor}$.

The constant term must be determined modulo p^{N-2} separately; from the explicit series expansions given above, it is simply $a_1/2$. $\square$

Finally we must solve (4) for $\theta(t)$. Formally the solution is given by

$$\theta(t) = \exp\left(\int h(t)dt\right).$$

Computationally this is problematic because both the integration and exponentiation steps introduce denominators, and also some p-adic precision is lost. The following result gives an efficient means to solve equations of the form $F'/F = f$, without any denominators appearing in intermediate steps, and with good control over precision loss.

PROPOSITION 6. *Let* $k \geqslant 1$ *and* $1 \leqslant n < p^{k/2}$, *and let* $S = \mathbf{Z}/p^k\mathbf{Z}$. *Let* $f \in S[t]/(t^{n-1})$, *and suppose that there exists* $F \in S[t]/(t^n)$, *with* $F(0) = 1$, *such that* $F'/F = f$. *Then* F *may be determined modulo* J *in time* $\widetilde{O}(nk\log p)$, *where* J *is the ideal of* $S[t]/(t^n)$ *given by*

$$J = (p^{k-1}t^p, p^{k-2}t^{p^2}, \dots).$$

Remark. Note that J is the ideal that captures the types of p-adic error terms that occur in a power series integration. Namely, if $g \in S[t]/(t^{n-1})$, and if the coefficient of t^j in g is divisible by $j+1$ for each j, then g has an integral in $S[t]/(t^n)$. In general it is not uniquely determined, but it is determined at least modulo J.

Proof. The algorithm we describe is essentially that of Brent [**2**], with some additional analysis to track the p-adic error terms.

We begin with an initial approximation $F_0(t) = 1 \in S[t]/(t^n)$. We will refine it iteratively, obtaining a sequence $F_i \in S[t]/(t^n)$ such that $F_i - F \in J + (t^{2^i})$ for each $i \geqslant 0$. After $\lceil \log_2 n \rceil$ steps, we will have $F_i - F \in J$ as desired. Each step is dominated by a polynomial multiplication and a division of length 2^i, so the total time is a constant multiple of the time required for a single polynomial multiplication of length n, which is $\widetilde{O}(n\log(p^k)) = \widetilde{O}(nk\log p)$.

Now we explain the iterative step. Suppose that $F_i - F \in J + (t^{2^i})$. Since F is invertible, we have

$$F_i = (1 + \varepsilon)F$$

for some $\varepsilon \in J + (t^{2^i})$, and so

$$\frac{F_i'}{F_i} - f = \frac{F'}{F} + \frac{\varepsilon'}{1+\varepsilon} - f = \frac{\varepsilon'}{1+\varepsilon}.$$

Morally speaking we would like to integrate $F_i'/F_i - f$ to obtain $\log(1+\varepsilon)$, but the latter does not make sense in our ring. Instead, we write

$$\frac{\varepsilon'}{1+\varepsilon} = \varepsilon' - \varepsilon'\varepsilon + \frac{\varepsilon'\varepsilon^2}{1+\varepsilon}.$$

The hypothesis $n < p^{k/2}$ implies that $J^2 = 0$, so that $\varepsilon^2 \in t^{2^i}J + (t^{2^{i+1}})$. We also have $\varepsilon' \in J + (t^{2^i-1})$, so $\varepsilon'\varepsilon^2 \in (t^{2^{i+1}-1})$. Consequently

$$\frac{F_i'}{F_i} - f \in \varepsilon' - \varepsilon'\varepsilon + (t^{2^{i+1}-1}).$$

Therefore $F_i'/F_i - f$ may be integrated at least up to $O(t^{2^{i+1}})$; that is, we may compute a $G \in S[t]/(t^n)$ such that

$$G \in \varepsilon - \frac{\varepsilon^2}{2} + J + (t^{2^{i+1}}).$$

(The extra J term is introduced by errors in the integration.) Since $\varepsilon^2 \in J$ we have in fact

$$G \in \varepsilon + J + (t^{2^{i+1}}).$$

Now it is straightforward to define F_{i+1}; we simply take

$$F_{i+1} = F_i(1 - G).$$

From the above estimates this satisfies

$$\begin{aligned} F_{i+1} &\in F(1 + \varepsilon)(1 - \varepsilon) + J + (t^{2^{i+1}}) \\ &= F - \varepsilon^2 F + J + (t^{2^{i+1}}) \\ &= F + J + (t^{2^{i+1}}) \end{aligned}$$

as desired. $\qquad\square$

Now we may complete the proof of Theorem 2. Let $\hat{h}(t)$ be the approximation to $h(t)$ produced by Proposition 5, treated as an element of $\mathbf{Z}_p[[t]]$, and let

$$\hat{\theta}(t) = \exp\left(\int \hat{h}(t)\, dt\right) \in \mathbf{Q}_p[[t]].$$

Observe that

$$\frac{\hat{\theta}(t)}{\theta(t)} = \exp\left(\int (\hat{h}(t) - h(t))\, dt\right).$$

By Proposition 5, for $1 \leqslant j < N$ the coefficient of t^j in $\hat{h} - h$ has valuation at least $N - 3 - \lfloor \log_p j \rfloor$, so the coefficient of t^j in $\int(\hat{h} - h)$ has valuation at least

$$N - 3 - \lfloor \log_p(j-1) \rfloor - \lfloor \log_p j \rfloor.$$

Therefore the coefficient of t^j in $\exp(\int(\hat{h} - h)) - 1$ has valuation at least

$$N - 3 - \lfloor \log_p(j-1) \rfloor - \lfloor \log_p j \rfloor - \left\lfloor \frac{j}{p-1} \right\rfloor,$$

because $v_p(j!) \leqslant j/(p-1)$. Since $p \geqslant 5$ we have

$$3 + \lfloor \log_p(j-1) \rfloor + \lfloor \log_p j \rfloor + \left\lfloor \frac{j}{p-1} \right\rfloor \leqslant j + 1$$

for all $j \geqslant 2$. (It suffices to estimate the left hand side for $p = 5$. For large enough j, the $j/4$ term dominates; one must also check a few small values of j directly.)

This shows that the t^j coefficients of $\hat{\theta}(t)$ and $\theta(t)$ agree modulo p^{N-j-1} for $2 \leqslant j < N$. This holds for $j = 1$ also, because of the extra condition on the constant term of $h(t)$ in Proposition 5. In other words, we have $\hat{\theta} - \theta \in I_{N-1}$. In particular, the coefficients of $\hat{\theta}$ are integral up to $O(t^{N-1})$, and the hypotheses of Proposition 6 are satisfied with $F = \hat{\theta}$, $f = \hat{h}$, $k = N - 2$ and $n = N - 1$.

The output of Proposition 6 is $\hat{\theta}$ modulo J. This determines θ modulo I_{N-1}, except for the constant term, which is known to be $a_1/2$.

EXAMPLE 7. We will walk through the steps of computing $\sigma_p(t)$ for the curve

$$E : y^2 + xy + y = x^3 - 460x - 3830,$$

which is '26a2' in Cremona's database, and the prime $p = 5$. We will take $N = 9$; that is, we will determine the coefficients of $t, t^2, \ldots, t^8$ modulo $5^8, 5^7, \ldots, 5^1$ respectively.

First we must compute $w(t)$ up to $O(t^{13})$, using Proposition 4. All computations below are performed modulo 5^6, and the printed integers stand for 5-adic numbers up to that precision, except where explicitly stated. The successive approximations are

$$w(t) = t^3 + O(t^4),$$
$$w(t) = t^3 + t^4 + t^5 + 2t^6 + 15169t^7 + O(t^8),$$
$$w(t) = t^3 + t^4 + t^5 + 2t^6 + 15169t^7 + 14252t^8 + 9048t^9$$
$$+ 9516t^{10} + 9477t^{11} + 14344t^{12} + O(t^{13}).$$

From this we obtain the other series attached to E,

$$y(t) = -1/w(t) = -t^{-3} + t^{-2} + 1 + 15166t + 15166t^2$$
$$+ 11337t^3 + 6589t^4 + 9397t^5 + 8273t^6 + O(t^7),$$

$$x(t) = t/w(t) = t^{-2} - t^{-1} - t + 459t^2 + 459t^3$$
$$+ 4288t^4 + 9036t^5 + 6228t^6 + 7352t^7 + O(t^8),$$

$$s(t) = x'(t)/(2y(t) + a_1x(t) + a_3) = 1 + t + t^2 + 3t^3 + 14712t^4 + 12878t^5$$
$$+ 14267t^6 + 1881t^7 + 4058t^8 + 2267t^9 + O(t^{10}).$$

Now we apply Proposition 5. Let us assume that $\mathbf{E}_2 = 4303 + O(5^6)$ has been computed in advance, so we have

$$c = \frac{1^2 + 4 \cdot 0 - 4303}{12} = 7454 + O(5^6),$$

and then

$$(x(t) + c)s(t) = t^{-2} + 7454 + 7455t + 6996t^2 + 5820t^3 +$$
$$13590t^4 + 11924t^5 + 15504t^6 + 1081t^7 + O(t^8).$$

Observe that the t^{-1} term is zero, so the series may be integrated. Our choice of c ensures that the t^4 term is divisible by $p = 5$, so that the integral has coefficients in $\mathbf{Z}_5$:

$$\int (x(t) + c)s(t)dt + \frac{a_1}{2} = -t^{-1} + 7813 + 7454t + 11540t^2 +$$
$$2332t^3 + 1455t^4 + 2718t^5 + 12404t^6 + 4447t^7 + 13807t^8 + O(t^9).$$

Note that the t^5 coefficient is now only correct modulo 5^5, because we lost a digit during the integration (in fact the correct coefficient is 12093 (mod 5^6)). We obtain

$$\hat{h}(t) = \frac{-1}{t} - s(t)\left(\int (x(t)+c)s(t)dt + \frac{a_1}{2}\right)$$
$$= 7813 + 359t + 4446t^2 + 1197t^3 + 14708t^4$$
$$+ 6580t^5 + 6770t^6 + 1524t^7 + 2441t^8 + O(t^9).$$

The preceding multiplication by $s(t)$ caused the incorrect digit to wash through to the higher order terms, so now the t^5 through t^8 coefficients are only correct modulo 5^5. Also, the constant term $a_1/2$ is only correct modulo 5^6, but we need to increase its precision to 5^7, so $\hat{h}(t)$ becomes

$$\hat{h}(t) = 39063 + 359t + 4446t^2 + 1197t^3 + 14708t^4 + 6580t^5$$
$$+ 6770t^6 + 1524t^7 + 2441t^8 + O(t^9).$$

Now we run Brent's algorithm (Proposition 6) to solve for $\hat{\theta}(t)$, with the polynomial arithmetic performed modulo 5^7. The successive approximations are

$$\hat{\theta}(t) = 1 + O(t),$$
$$\hat{\theta}(t) = 1 + 39063t + O(t^2),$$
$$\hat{\theta}(t) = 1 + 39063t + 68539t^2 + 12965t^3 + O(t^4),$$
$$\hat{\theta}(t) = 1 + 39063t + 68539t^2 + 12965t^3 + 30804t^4 + 14720t^5$$
$$+ 10063t^6 + 25830t^7 + O(t^8).$$

Finally $\sigma_p(t)$ is obtained as $t\hat{\theta}(t)$, but Theorem 2 only guarantees the result is correct modulo I_N, so we have

$$\sigma_p(t) = t + (39063 + O(5^7))t^2 + (6039 + O(5^6))t^3$$
$$+ (465 + O(5^5))t^4 + (179 + O(5^4))t^5 + (95 + O(5^3))t^6$$
$$+ (13 + O(5^2))t^7 + (0 + O(5))t^8 + O(t^9).$$

5. Computing the p-adic height

In this section we prove Theorem 3. Recall that $n_1 = \#E(\mathbf{F}_p)$, n_2 is the least common multiple of the Tamagawa numbers of E, and $n = \text{LCM}(n_1, n_2)$. We are given a point $P \in E(\mathbf{Q})$ as input, whose coordinates require space C_P to store. Given the sigma function modulo $I_{M'+1}$ as input, where

$$M' = M + 2v_p(n),$$

we wish to compute $h_p(P)$ to precision p^M, in time $\widetilde{O}(C_P + M\log^2 p + M^2 \log p)$.

As noted in §2, computing nP directly is not feasible for large p. Therefore we take a more indirect approach. In what follows, the symbols $\alpha(P)$, $\beta(P)$, $d(P)$, $x(P)$, $y(P)$ and $t(P)$ are the same as those defined in §2.2.

First we compute $Q = n_2P$, so that Q satisfies the condition (A2) defined in §2.2.

Now let $m = n/n_2$. Because $mQ \; (= nP)$ satisfies both conditions (A1) and (A2), we may express $h_p(mQ)$ using (2). Since h_p is a quadratic form, we obtain

$$h_p(P) = \frac{2}{n^2} \log_p \left(\frac{\sigma_p(mQ)}{d(mQ)} \right). \tag{5}$$

(Note that $\log_p$ is now again the p-adic logarithm, not the base-p logarithm.) To determine $h_p(P)$ modulo p^M, it suffices to compute

$$\log_p \left(\frac{\sigma_p(mQ)}{d(mQ)} \right) = \log_p \left(\frac{-\alpha(mQ)}{\beta(mQ)} \left(1 + \sum_{k \geqslant 1} c_{k+1} t(mQ)^k \right) \right) \tag{6}$$

modulo $p^{M'}$, where we have expanded $\sigma_p(t)$ as

$$\sigma_p(t) = t + c_2 t^2 + c_3 t^3 + \cdots.$$

Note that $\alpha(mQ)/\beta(mQ)$ is a unit, since both $\alpha(mQ)$ and $\beta(mQ)$ are relatively prime to $d(mQ)$, which is divisible by p. Also $1 + \sum_{k \geqslant 1} c_{k+1} t(mQ)^k$ is a unit, again because $t(mQ) = -d(mQ)\alpha(mQ)/\beta(mQ)$ is divisible by p.

LEMMA 8. *Suppose that $u \in \mathbf{Z}_p^*$ is a unit and that $N \geqslant 1$ is an integer. To determine $\log_p u$ modulo p^N it suffices to know u modulo p^N.*

Proof. From $\log_p u = \frac{1}{p-1} \log_p(u^{p-1})$ we may assume that $u \equiv 1 \pmod{p}$. Since $\log_p(u+\varepsilon) = \log_p u + \log_p(1+\varepsilon/u)$, the result amounts to checking that if $v_p(\varepsilon) \geqslant N$, then $v_p(\log_p(1 + \varepsilon)) \geqslant N$. Using the power series expansion of $\log_p(1 + x)$, this follows from the elementary estimate $v_p(\varepsilon^n/n) \geqslant v_p\varepsilon$, that is, $v_p(n) \leqslant n - 1$ for all $n \geqslant 1$. □

By Lemma 8, it suffices to compute

$$\frac{\alpha(mQ)}{\beta(mQ)} \qquad \text{and} \qquad 1 + \sum_{k \geqslant 1} c_{k+1} t(mQ)^k$$

modulo $p^{M'}$. By hypothesis we have available the sigma function modulo $I_{M'+1}$, which means that we know c_j modulo $p^{M'+1-j}$ for $2 \leqslant j \leqslant M'$. Since p divides $t(mQ)$, it therefore suffices to compute $\alpha(mQ)/\beta(mQ)$ and $t(mQ)$ modulo $p^{M'}$. For this, we have the following result, which is proved in §5.1 below.

PROPOSITION 9. *Suppose that $Q \in E(\mathbf{Q})$ reduces to a non-singular point of $E(\mathbf{F}_\ell)$ for every prime ℓ at which E has bad reduction. Let $L \geqslant 1$ be odd, and $m \geqslant 2$. Given the values of $\alpha(Q)$, $\beta(Q)$ and $d(Q)$ modulo L, one may compute $\pm\alpha(mQ)$, $\beta(mQ)$ and $\pm d(mQ)$ modulo L in time $\widetilde{O}(\log L \log m)$.*

(The $\pm$ symbols indicate that $\alpha(mQ)$ and $d(mQ)$ will be correct only up to sign, and that the signs will agree.)

Proposition 9 completely avoids the problem of the coordinates of mQ growing out of control, since the dependence on m is only logarithmic.

We apply Proposition 9 with $L = p^{M'}$, to determine $\alpha(mQ)/\beta(mQ)$ and $t(mQ)$ modulo $p^{M'}$. The sign ambiguity is irrelevant, since the signs cancel out in $t(mQ) = -d(mQ)\alpha(mQ)/\beta(mQ)$, and the sign of $\alpha(mQ)/\beta(mQ)$ is not needed because the p-adic logarithm is insensitive to the sign of its input.

Now we analyse the running time. Recall n_2 is assumed to be $O(1)$, so the time for computing $Q = n_2 P$ is $\widetilde{O}(C_P)$. Applying Proposition 9 costs $\widetilde{O}(\log(p^{M'}) \log m)$. We have $M' = O(M)$ and $m = O(p)$, so this is $\widetilde{O}(M \log^2 p)$. We must substitute $t(mQ)$ into (6), which requires $O(M')$ ring operations in $\mathbf{Z}/p^{M'}\mathbf{Z}$, costing time $\widetilde{O}(M' \log(p^{M'})) = \widetilde{O}(M^2 \log p)$. Finally we must compute the p-adic logarithm in (5). Using the series expansion of $\log_p(1 + x)$, this requires $O(M')$ ring operations, for a cost of $\widetilde{O}(M^2 \log p)$; with more effort one can obtain $\widetilde{O}(M \log p)$ [**1**, §16].

5.1. *Proof of Proposition 9*

Our main tools in the proof are the division polynomials ψ_m associated to our choice of Weierstrass equation for E, and a non-cancellation result of Wuthrich [**11**, Prop. IV.2] that in effect controls the amount of cancellation that can occur while computing mQ from Q. For further background on division polynomials, including proofs of the assertions we use in this section, we refer the reader to [**8**, Appendix I] and [**6**, Ch. II].

The relevance of division polynomials is that they appear in a simple formula for the coordinates of mQ in terms of the coordinates of Q. For an integer $m \geqslant 1$, and a non-torsion point Q, we have

$$x(mQ) = \frac{\theta_m(Q)}{\psi_m(Q)^2}, \qquad y(mQ) = \frac{\omega_m(Q)}{\psi_m(Q)^3}, \tag{7}$$

where $\theta_m, \omega_m \in \mathbf{Q}(E)$ are certain auxiliary functions defined in terms of the ψ_m.

The quantities $\psi_m(Q), \theta_m(Q), \omega_m(Q) \in \mathbf{Q}$ will generally not be integers, due to the coordinates of Q themselves having denominators. It is convenient to introduce a normalising factor that absorbs these denominators. Accordingly we set

$$\hat{\psi}_m(Q) = \psi_m(Q)d(Q)^{m^2-1},$$
$$\hat{\theta}_m(Q) = \theta_m(Q)d(Q)^{2m^2},$$
$$\hat{\omega}_m(Q) = \omega_m(Q)d(Q)^{3m^2}.$$

Note that $\hat{\psi}_m$, $\hat{\theta}_m$ and $\hat{\omega}_m$ are defined only on $E(\mathbf{Q})$ — they are most definitely *not* rational functions on E. One checks, by examining the degrees of ψ_m, θ_m and ω_m, that $\hat{\psi}_m(Q)$, $\hat{\theta}_m(Q)$ and $\hat{\omega}_m(Q)$ are all integers. Therefore we now have *two* representations of $x(mQ)$ and $y(mQ)$ as ratios of integers,

$$x(mQ) = \frac{\alpha(mQ)}{d(mQ)^2} = \frac{\hat{\theta}_m(Q)}{\hat{\psi}_m(Q)^2 d(Q)^2},$$
$$y(mQ) = \frac{\beta(mQ)}{d(mQ)^3} = \frac{\hat{\omega}_m(Q)}{\hat{\psi}_m(Q)^3 d(Q)^3}. \tag{8}$$

The point of (8) is twofold. First, Proposition IV.2 of [**11**] guarantees that, under the hypothesis of Proposition 9, the fractions on the right are in fact *reduced* fractions; in other words, that $d(mQ) = \pm\hat{\psi}_m(Q)d(Q)$. Therefore we may conclude from (8) that

$$d(mQ) = \pm\hat{\psi}_m(Q)d(Q),$$
$$\alpha(mQ) = \hat{\theta}_m(Q), \tag{9}$$
$$\beta(mQ) = \pm\hat{\omega}_m(Q),$$

where the choices of signs in the first and third equations agree.

Second, we will show that $\hat{\psi}_m(Q)$, $\hat{\theta}_m(Q)$ and $\hat{\omega}_m(Q)$ can be efficiently computed modulo L using the usual recursion formulae for the division polynomials. For our application, it is not necessary to compute the division polynomials themselves — in fact, to do so would completely miss the point, since their degree grows like m^2, which is precisely the rate of growth that we are trying to avoid. Rather, we need only compute their *values at Q*, and only modulo L. Fortunately the standard recursive formulae for division polynomials, with minor modifications, are perfectly suited to this task.

To avoid losing p-adic precision, we give a version of the formulae that involve no divisions (apart from one division by 2, which is permitted since we have assumed that 2 is invertible modulo L). Also, our formulae are tailored to computing the normalised versions $\hat{\psi}_m(Q)$, $\hat{\theta}_m(Q)$ and $\hat{\omega}_m(Q)$ directly, instead of $\psi_m(Q)$, $\theta_m(Q)$ and $\omega_m(Q)$, so that we can work with integral quantities throughout.

In the formulae below, we abbreviate $\alpha(Q)$ by α, and similarly with the other variables. We start by defining normalised versions of the coefficients a_k of the elliptic curve, setting

$$\hat{a}_k = d^k a_k, \quad k = 1, 2, 3, 4, 6.$$

We next define variables $\hat{b}_k$ and $\hat{B}_k$, which are normalised versions of the b_k and B_k appearing in [8]:

$$\hat{b}_2 = \hat{a}_1^2 + 4\hat{a}_2,$$
$$\hat{b}_4 = \hat{a}_1\hat{a}_3 + 2\hat{a}_4,$$
$$\hat{b}_6 = \hat{a}_3^2 + 4\hat{a}_6,$$
$$\hat{b}_8 = \hat{a}_1^2\hat{a}_6 + 4\hat{a}_2\hat{a}_6 - \hat{a}_1\hat{a}_3\hat{a}_4 + \hat{a}_2\hat{a}_3^2 - \hat{a}_4^2,$$
$$\hat{B}_4 = 6\alpha^2 + \hat{b}_2\alpha + \hat{b}_4,$$
$$\hat{B}_6 = 4\alpha^3 + \hat{b}_2\alpha^2 + 2\hat{b}_4\alpha + \hat{b}_6,$$
$$\hat{B}_8 = 3\alpha^4 + \hat{b}_2\alpha^3 + 3\hat{b}_4\alpha^2 + 3\hat{b}_6\alpha + \hat{b}_8.$$

Similarly, we need normalised versions of the g_m from [8], defined by

$$\hat{g}_0 = 0, \quad \hat{g}_1 = 1, \quad \hat{g}_2 = -1, \quad \hat{g}_3 = \hat{B}_8, \quad \hat{g}_4 = \hat{B}_6^2 - \hat{B}_4\hat{B}_8,$$

and then recursively for $m \geqslant 5$ by

$$\hat{g}_{2n+1} = \begin{cases} \hat{B}_6^2\hat{g}_{n+2}\hat{g}_n^3 - \hat{g}_{n-1}\hat{g}_{n+1}^3, & n \text{ even}, \\ \hat{g}_{n+2}\hat{g}_n^3 - \hat{B}_6^2\hat{g}_{n-1}\hat{g}_{n+1}^3, & n \text{ odd}, \end{cases} \tag{10}$$
$$\hat{g}_{2n} = \hat{g}_n(\hat{g}_{n-2}\hat{g}_{n+1}^2 - \hat{g}_{n+2}\hat{g}_{n-1}^2).$$

Finally, the values of $\hat{\psi}_m$, $\hat{\theta}_m$ and $\hat{\omega}_m$ for $m \geqslant 2$ are given in terms of the $\hat{g}_m$ by

$$\hat{T} = 2\beta + \hat{a}_1\alpha + \hat{a}_3,$$
$$\hat{\psi}_m = \hat{T}^{\sigma(m+1)}\hat{g}_m,$$
$$\hat{\theta}_m = \alpha\hat{\psi}_m^2 - \hat{\psi}_{m+1}\hat{\psi}_{m-1}, \tag{11}$$
$$\hat{\omega}_m = \frac{-1}{2}\left(\hat{T}^{\sigma(m)}(\hat{g}_{m-2}\hat{g}_{m+1}^2 - \hat{g}_{m+2}\hat{g}_{m-1}^2) + \hat{\psi}_m(\hat{a}_1\hat{\theta}_m + \hat{a}_3\hat{\psi}_m^2)\right),$$

where $\sigma(k)$ is 0 or 1 accordingly as k is even or odd.

The algorithm implementing Proposition 9 now runs as follows. All computations are performed modulo L. We are given as input $\alpha(Q)$, $\beta(Q)$ and $d(Q)$, and the constants a_k. From (9) it suffices to compute $\hat{\psi}_m$, $\hat{\theta}_m$ and $\hat{\omega}_m$.

We start by computing all of the $\hat{a}_k$, $\hat{b}_k$ and $\hat{B}_k$, and $\hat{g}_0$ through $\hat{g}_4$, using the formulae given above. Then, using (10), recursively compute $\hat{g}_{m-2}$ through $\hat{g}_{m+2}$. During this recursive step, it is important to retain the values of $\hat{g}_j$ as they are computed, since many of them will be reused. Finally, the equations (11) then determine $\hat{\psi}_m$, $\hat{\theta}_m$ and $\hat{\omega}_m$.

Now we analyse the complexity. Arithmetic operations in $\mathbf{Z}/L\mathbf{Z}$ may be performed in time $\widetilde{O}(\log L)$, so the crux of the matter is to show that the recursive formulae (10) are evaluated at most $O(\log m)$ times.

Let $k \geqslant 4$. To determine $\hat{g}_j$ for all j in the range $k \leqslant j \leqslant k+7$, using (10) it suffices to know $\hat{g}_j$ for $(k-3)/2 \leqslant j \leqslant (k+11)/2$ if k is odd, or $(k-4)/2 \leqslant j \leqslant (k+10)/2$ if k is even. In other words, to determine 8 consecutive values of $\hat{g}_j$ near $j = k$, it suffices to know 8 consecutive values of $\hat{g}_j$ near $j = k/2$. Iterating this process, to compute $\hat{g}_k$ one must evaluate (10) at most $8\log_2(k) = O(\log k)$ times.

EXAMPLE 10. Let $P = (5/4, -3/8)$ be a generator of $E(\mathbf{Q})$ where E is the elliptic curve $y^2 + y = x^3 + x^2 - 7x + 5$. This curve is '91b1' in Cremona's database, and has conductor $91 = 7 \cdot 13$. One checks directly that P reduces to a non-singular point at the bad primes 7 and 13.

We will illustrate Proposition 9 for the case $L = 99$, $m = 101$. The starting point P is specified by

$$\alpha = 5, \quad \beta = 96, \quad d = 2.$$

The various constants are initialised as

$$\hat{a}_1 = 0, \quad \hat{b}_2 = 16, \quad \hat{B}_4 = 6,$$
$$\hat{a}_2 = 4, \quad \hat{b}_4 = 73, \quad \hat{B}_6 = 4,$$
$$\hat{a}_3 = 8, \quad \hat{b}_6 = 57, \quad \hat{B}_8 = 67,$$
$$\hat{a}_4 = 86, \quad \hat{b}_8 = 59, \quad \hat{T} = 2.$$
$$\hat{a}_6 = 23,$$

We now list the intermediate results in the computation of $\alpha(mP)$, $\beta(mP)$ and $d(mP)$ modulo L. The required values of $\hat{g}_m$ are

$\hat{g}_0 = 0$,	$\hat{g}_6 = 63$,	$\hat{g}_{12} = 0$,	$\hat{g}_{23} = 35$,	$\hat{g}_{48} = 0$,	$\hat{g}_{99} = 49$,
$\hat{g}_1 = 1$,	$\hat{g}_7 = 98$,	$\hat{g}_{13} = 64$,	$\hat{g}_{24} = 0$,	$\hat{g}_{49} = 1$,	$\hat{g}_{100} = 19$,
$\hat{g}_2 = 98$,	$\hat{g}_8 = 35$,	$\hat{g}_{14} = 71$,	$\hat{g}_{25} = 91$,	$\hat{g}_{50} = 62$,	$\hat{g}_{101} = 82$,
$\hat{g}_3 = 67$,	$\hat{g}_9 = 50$,	$\hat{g}_{15} = 4$,	$\hat{g}_{26} = 17$,	$\hat{g}_{51} = 49$,	$\hat{g}_{102} = 72$,
$\hat{g}_4 = 10$,	$\hat{g}_{10} = 73$,	$\hat{g}_{16} = 1$,	$\hat{g}_{27} = 67$,	$\hat{g}_{52} = 46$,	$\hat{g}_{103} = 98$.
$\hat{g}_5 = 37$,	$\hat{g}_{11} = 98$,	$\hat{g}_{22} = 1$,	$\hat{g}_{28} = 46$,	$\hat{g}_{53} = 1$,	

From these it follows that

$$\hat{\psi}_{100} = 38, \quad \hat{\psi}_{101} = 82, \quad \hat{\psi}_{102} = 45,$$

and

$$\hat{\theta}_{101} = 32, \quad \hat{\omega}_{101} = 4.$$

Therefore we obtain

$$\alpha(101P) = 32,$$
$$\beta(101P) = \pm 4,$$
$$d(101P) = \pm 2 \cdot 82 = \pm 65.$$

The result may be verified by using a machine to explicitly compute the coordinates of $101P$; it turns out that the negative sign was the correct one.

6. A complete example

We will illustrate the main algorithm by stepping through a computation of the p-adic height of a rational point on the curve

$$E \colon y^2 + xy = x^3 - 12x + 16,$$

which is '214a1' from Cremona's database. It has conductor $214 = 2 \cdot 107$, and its rank over $\mathbf{Q}$ is one, with generator

$$P = (0, 4).$$

To make life interesting, we will take $p = 43$, which is anomalous for E since $\#E(\mathbf{F}_{43}) = 43$. We will aim to compute $h_p(P)$ to precision p^6, so set $M = 6$. The Tamagawa numbers of E at 2 and 107 are respectively 7 and 1; therefore we set

$$n_1 = \#E(\mathbf{F}_{43}) = 43,$$
$$n_2 = \mathrm{LCM}(7, 1) = 7,$$
$$n = \mathrm{LCM}(n_1, n_2) = 301,$$
$$m = n/n_2 = 43,$$
$$M' = M + 2v_{43}(n) = 8.$$

To apply Theorem 3 we need $\sigma_p(t)$ modulo I_9; from Theorem 2 this means we require $\mathbf{E}_2$ modulo p^6. To be able to apply Kedlaya's algorithm, it is convenient to change coordinates to put the curve into the Weierstrass form $y^2 = x^3 + ax + b$; for our curve this turns out to be

$$E' \colon y^2 = x^3 - \frac{577}{48}x + \frac{14689}{864}.$$

(Note that the equation E' has the same discriminant $\Delta = -13696$ as our original equation, so the value of $\mathbf{E}_2$ we compute for E' will be the correct value also for E. If instead we chose an equation whose discriminant differed by a factor of u^{12}, then we would need to adjust the computed $\mathbf{E}_2$ by u^2, since $\mathbf{E}_2$ is of weight two. In other words, we need to take into account the choice of invariant differential implied by the choice of equation, as $\mathbf{E}_2(E, \omega)$ is a function of both E and ω.)

We apply Kedlaya's algorithm — omitting the details — to find that the Frobenius matrix is

$$F = \begin{pmatrix} 4996923274 & 3651910366 \\ 1002107518 & 1324439776 \end{pmatrix} \pmod{p^6}.$$

Then

$$F^6 = \begin{pmatrix} 3987851820 & 4837860471 \\ 1528699020 & 2333368599 \end{pmatrix} \pmod{p^6},$$

so

$$\mathbf{E}_2 = -12 \cdot \frac{4837860471}{2333368599} = 5899790810 \pmod{p^6}.$$

Using $\mathbf{E}_2$ as input, Theorem 2 then yields the sigma function,

$$\sigma_p(t) = t + (135909305554 + O(p^7))t^2 + (3933286396 + O(p^6))t^3$$
$$+ (129848206 + O(p^5))t^4 + (2650487 + O(p^4))t^5$$
$$+ (77893 + O(p^3))t^6 + (1561 + O(p^2))t^7 + (8 + O(p))t^8 + O(t^9).$$

(See Example 7 for a more detailed example of computing $\sigma_p(t)$.)

Now we bring into the picture the particular P whose height is sought. We compute

$$Q = n_2 P = \left(\frac{3}{4}, \frac{-25}{8}\right),$$

so that $\alpha(Q) = 3$, $\beta(Q) = -25$, and $d(Q) = 2$. Using Proposition 9, we then find that, modulo p^8,

$$\alpha(mQ) = 9491762277279,$$
$$\beta(mQ) = \pm 10171094217691,$$
$$d(mQ) = \pm 3360349669562.$$

(See Example 10 for a more detailed example of this step.) Substituting everything into (5) and (6), we find that

$$h_p(P) = \frac{2}{301^2} \log_p(1430987165464 + O(43^8))$$
$$= \frac{2}{7^2 \cdot 43^2}(43 \cdot 44668563676 + O(43^8))$$
$$= 43^{-1} \cdot 96127622779 + O(43^6),$$

to precision p^6 as desired.

7. Sample computations

The author implemented the above algorithms in the computer algebra system SAGE [10], building on an implementation of the algorithm of [7] by Robert Bradshaw, Jennifer Balakrishnan, Liang Xiao, and the author. The code is freely available under a GPL license, and is distributed as a standard component of SAGE (version 2.2 and later). An example session:

```
sage: E = EllipticCurve("37a"); E
 Elliptic Curve defined by y^2 + y = x^3 - x over Rational Field
sage: P = E.gens()[0]; P     # a generator of E(Q)
 (0 : 0 : 1)
sage: h = E.padic_height(p=5, prec=5)
sage: h(P)
```

```
4*5 + 3*5^2 + 3*5^3 + 4*5^4 + O(5^5)
```

The examples below were run on an AMD Opteron running at 1.8 GHz, kindly provided by William Stein, funded by his NSF grant #0555776. For these examples, we did *not* use the methods of §3.1, opting instead to use the trace and determinant as a correctness check.

7.1. *Large prime case*

We will take the elliptic curve '92b1' from Cremona's database, which has equation $y^2 = x^3 - x + 1$ and conductor $92 = 2^2 \cdot 23$. A generator of the group of rational points is $P = (x, y) = (1, 1)$.

Let $p = 10^{11} + 3$ and $M = 6$. We need to compute $\mathbf{E}_2$ modulo p^4. Using the method of [4], SAGE finds that the matrix of Frobenius on the standard basis for Monsky–Washnitzer cohomology is given by

6430458576087611569817568034419831813001383   42503972380936025561124602310186734870477220

9712855838536821054056814178945773533527354   35695414251123884302364319655812481869943749

modulo p^4. The computation time was 42 minutes. As a consistency check, one may verify that the determinant of this matrix is $p \pmod{p^4}$, and that the trace is

$$1000000000120000000005400000000107999999956832 \equiv -43249 \pmod{p^4},$$

which agrees with other point-counting algorithms.

From the matrix SAGE finds immediately that

$$\mathbf{E}_2 = 744701682804855332135084234707411222845601 52 + O(p^4),$$

and that the p-adic sigma function is

$$\sigma_p(t) = t + (69769590353020230550922850977954746761856727 + O(p^4))t^3$$
$$+ (5214659177355434704657 + O(p^2))t^5 + O(t^7).$$

Finally, SAGE uses Theorem 3 to find that $h_p(P)$ is

$$p \cdot 9226324270539878944369124959203473806055293044599072658 + O(p^6).$$

This last step is virtually instantaneous. Observe that the original algorithm of [7] would have required computing the coordinates of the point nP where $n \approx 10^{11}$, which would occupy around 10^{22} bits of storage — not to mention the computation time. Clearly, Proposition 9 is essential for handling such large p.

7.2. *High precision case*

Continuing with the same curve, we now consider the case $p = 5$ and $M = 3000$. Therefore we require $\mathbf{E}_2$ modulo 5^{2998}. Since p is small relative to M, SAGE selects Kedlaya's original algorithm [5] for the Frobenius matrix computation. The first and last few digits of $\mathbf{E}_2$ are

$$\mathbf{E}_2 = 3 + 2 \cdot 5 + 2 \cdot 5^3 + 3 \cdot 5^5 + 2 \cdot 5^7 + \cdots + 3 \cdot 5^{2995} + 3 \cdot 5^{2996} + O(5^{2998}).$$

The computation time to obtain $\mathbf{E}_2$ was 229 seconds.

Obtaining the p-adic sigma function took 158 seconds. There are about 3000 coefficients, each with a similar number of 5-adic digits, so we dare not write it down here. For such high precision computations, the original algorithm of [7]

would have been utterly impractical; the M^4 contribution from the initial power series expansions would multiply the above running time by a factor of perhaps 10^7.

Finally, computing $h_p(P)$ itself from the sigma function took about 6 seconds. It turns out to be

$$h_p(P) = 3 \cdot 5 + 3 \cdot 5^2 + 2 \cdot 5^3 + 5^4 + \cdots + 4 \cdot 5^{2998} + 2 \cdot 5^{2999} + O(5^{3000}).$$

Acknowledgements

References

1. D. BERNSTEIN, 'Fast multiplication and its applications'. 52, 58
http://cr.yp.to/lineartime/multapps-20041007.pdf

2. R. P. BRENT, 'Multiple-precision zero-finding methods and the complexity of elementary function evaluation', *Analytic computational complexity*, Pittsburgh, 1975 (Academic Press, New York, 1976) 151–176. 47

3. R. P. BRENT and H. T. KUNG, 'Fast algorithms for manipulating formal power series', *J. Assoc. Comput. Mach.* 25 (1978) 581–595. 42

4. D. HARVEY, 'Kedlaya's algorithm in larger characteristic', *Int. Math. Res. Notices* (2007), no. rnm095, rnm095–29. 43, 44, 57

5. K. S. KEDLAYA, 'Counting points on hyperelliptic curves using Monsky-Washnitzer cohomology', *J. Ramanujan Math. Soc.* 16 (2001) 323–338. 42, 43, 44, 57

6. S. LANG, *Elliptic curves: Diophantine analysis*, vol. 231 of *Grundlehren der Mathematischen Wissenschaften* [*Fundamental Principles of Mathematical Sciences*] (Springer-Verlag, Berlin, 1978). 52

7. B. MAZUR, W. STEIN and J. TATE, 'Computation of p-adic heights and log convergence', *Documenta Math.* (*Extra Volume: John H. Coates' Sixtieth Birthday*) (2006) 577–614. 40, 41, 42, 43, 46, 56, 57

8. B. MAZUR and J. TATE, 'The p-adic sigma function', *Duke Math. J.* 62 (1991) 663–688. 41, 52, 53

9. J. H. SILVERMAN, *The arithmetic of elliptic curves*, vol. 106 of *Graduate Texts in Mathematics* (Springer-Verlag, New York, 1992). Corrected reprint of the 1986 original. 45, 46

10. W. STEIN and D. JOYNER, 'Sage: System for algebra and geometry experimentation', *Communications in Computer Algebra (ACM SIGSAM Bulletin)* 39 (2005) 61–64. 56, 58
http://www.sagemath.org/

11. C. WUTHRICH, 'The fine Selmer group and height pairings', Ph.D. thesis, Cambridge, (2004). 52
http://www.maths.nottingham.ac.uk/personal/cw/thesisandessays.html

David Harvey   dmharvey@math.harvard.edu

Department of Mathematics
Harvard University
1 Oxford St, Cambridge, MA 02138, USA