Radical isogenies on every model

Thomas Decru
COSIC, KU Leuven, Leuven, Belgium
thomas.decru@kuleuven.be
ORCID: 0000-0003-0253-4180

Abstract

Radical isogeny formulae compute a cyclic chain of N-isogenies with one Nth root per step and without sampling torsion points, but for general N they were only known in Tate normal form. We give radical steps of every odd degree N directly on short Weierstrass, Montgomery, twisted Edwards, Huff, extended Jacobi quartic, twisted Hessian, Legendre and Jacobi intersection curves, on Kummer lines, and on all curves with a rational point of order 2. Our approach is model-independent. The Tate normal form data is obtained from the x-coordinate of the kernel generator by a shifted recursion, and the codomain is recovered in the original model by locating the image of a distinguished torsion point, without square roots. In Tate normal form we lower the cost of a step from $12.5N$ to $10N$ multiplications plus one exponentiation, and short Weierstrass curves come close at $10N + 35$. Montgomery and twisted Edwards curves cost $10.5N + O(1)$. However, pushing a point through their steps costs only $2N$ multiplications without precomputation, against $2N + 16$ plus a shared $2N - 7$ in Tate normal form, which makes them the cheapest models for chains that transport torsion points, e.g. for all $N \geq 19$ when two points are pushed. The other models cost between $11.5N + O(\log N)$ and $15.5N + O(\log N)$ per step. All algorithms are inversion-free, with exact operation counts.

Keywords: radical isogenies, isogeny-based cryptography, elliptic curve models, Vélu's formulae, Montgomery curves, CSIDH
Mathematics Subject Classification (2020): 14H52, 14K02, 11G20, 11Y16, 94A60

1 Introduction

Models of elliptic curves. Every elliptic curve over a field K of characteristic different from 2 and 3 has a short Weierstrass model, but for computations other models are often preferred. In 1987, Montgomery introduced the curves $By^2 = x^3 + Ax^2 + x$ [30], whose fast x-only ladder later made them the standard choice for CSIDH [10]; a curve has a Montgomery model if and only if it has a rational point T of order 2 satisfying a quadratic residuosity condition [33]. Twisted Edwards curves $ax^2 + y^2 = 1 + dx^2y^2$ [1], which generalize the curves introduced by Edwards in 2007 [19] and brought to cryptography by Bernstein and Lange [4], are birationally equivalent to Montgomery curves and have complete addition laws. Hessian curves $X^3 + Y^3 + Z^3 = dXYZ$ [22, 39] require a rational point of order 3 but do not cover all such curves, whereas twisted Hessian curves $aX^3 + Y^3 + Z^3 = dXYZ$ [2] exist exactly for curves with a rational point of order 3. Huff curves $ax(y^2 - 1) = by(x^2 - 1)$ [23] require a rational subgroup $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/4\mathbb{Z}$. Jacobi quartics $y^2 = dx^4 + 2ax^2 + 1$ [5, 21] require a rational point of order 2, and Jacobi intersections $s^2 + c^2 = 1$, $as^2 + d^2 = 1$ [12, 28] require full rational 2-torsion. Legendre curves $y^2 = x(x-1)(x-\lambda)$ also require full rational 2-torsion, and Kummer lines in squared theta coordinates [20, 24] give an x-only model. Finally, the Tate normal form $y^2 + (1 - c)xy - by = x^3 - bx^2$ parametrizes pairs (E, P) with P of order at least 4, and is the model in which radical isogenies were originally formulated [9].

Isogenies between models. Vélu's formulae [40] from 1971, in the x-only form of Kohel [27], compute isogenies between Weierstrass curves. With the rise of isogeny-based cryptography, analogues for the other models followed. In 2016, Moody and Shumow [31] gave formulae for Edwards and Huff curves, and in 2017 Costello and Hisil [14] gave odd-degree formulae for Montgomery curves that keep the 2-torsion point $(0, 0)$ fixed, which Renes [36] explained and generalized through the action of $(0, 0)$. Meyer and Reith [29] then observed that the codomain of such an isogeny is cheapest to compute in twisted Edwards form, Kim, Yoon, Park and Hong [26] optimized Edwards isogenies further with w-coordinates, and Moriya, Onuki and Takagi [32] constructed CSIDH directly on Edwards curves. Isogeny formulae for twisted Hessian curves were given by Dang and Moody [15] and by Perez Broon, Dang, Fouotsa and Moody [35]. For Huff curves, Kim [25] analyzed Huff-based SIDH and CSIDH, and Wroński [41] adapted the square-root Vélu formulae of [3]. More recently, Dzierzkowski and Wroński [18] treated extended Jacobi quartics, and Robert and Sarkis [37] computed 2-isogenies between Kummer lines using theta groups. However, all of these formulae take the kernel as input, i.e. they need a point of order N, which is typically obtained by sampling.

Radical isogenies. In 2020, Castryck, Decru and Vercauteren introduced radical isogenies [9] to avoid this sampling. Starting from a curve in Tate normal form with a point P of order N, they compute the next Tate normal form, in which $(0, 0)$ generates the kernel of the next step, using one Nth root. Explicit formulae were first found for small N [8, 9], conjectured for all odd N in [16], and proved for every $N \geq 4$ in [17], where an inversion-free step of about $12.5N$ multiplications is also given. For the small degrees of [9], Chi-Domínguez and Reijnders [11] had already given fully projective, hence inversion-free, constant-time versions of the formulae for use in CSIDH and CSURF. On Montgomery curves, radical isogenies of degree 3 and 4 were given by Onuki and Moriya [34]. For general N, however, a chain on a practical model so far had to convert to Tate normal form, run the chain, and convert back. Converting back requires locating a 2-torsion point (Montgomery, Edwards) or a 3-torsion point (Hessian) of the codomain, and in general a square root, which costs an additional exponentiation per step.

Notation and cost model. Throughout, K is a field of characteristic different from 2, 3 and from the prime divisors of N, and $N \geq 5$ is odd. For a point P we write $x_i = x([i]P)$, and primes denote quantities on the codomain. We count multiplications and squarings in K as $\mathbf{M}$, and ignore additions and multiplications by fixed constants, i.e. by integers and by quotients of integers such as $\frac{1}{2}$ or $\frac{16}{9}$, which are precomputed once. All radical step counts exclude the Nth root, i.e. one exponentiation by $N^{-1} \bmod p - 1$ over $\mathbb{F}_p$. The remaining powers have public exponents of size about N, and we write

$$\mathbf{L}_k = \lfloor \log_2 k \rfloor, \qquad \mathbf{E}_k = \mathbf{L}_k + \mathrm{wt}(k) - 1 \leq 2 \log_2 k$$

for the cost of $x \mapsto x^k$ by left-to-right square-and-multiply, where $\mathrm{wt}(k)$ is the binary weight of k. For odd N we have $\mathbf{E}_{N-1} = \mathbf{E}_N - 1$ and $\mathbf{E}_{(N-1)/2} = \mathbf{E}_N - 2$. All exact counts hold for $N \geq 7$, except in Tate normal form, where they hold for $N \geq 11$, and on short Weierstrass and twisted Hessian curves, where they hold for $N \geq 9$ (i.e. $N \geq 11$ for twisted Hessian curves, as $3 \nmid N$); for smaller N some loops are empty and the counts differ by a few $\mathbf{M}$.

Our contributions. Let $N \geq 5$ be odd. For each of the models above, we provide a radical N-isogeny step whose input is the curve and $x(P)$ for a point P of order N. It outputs the codomain in the same model, together with the x-coordinate of a point P' that generates the kernel of the next step. More precisely:

- the Tate normal form data of (E, P) is obtained from $x(P)$ alone, and the Vélu codomain with the next kernel generator is read off from the radical output at constant cost, which gives a model-independent reduction (Section 3.2);

- a reflection symmetry (Lemma 3.1), telescoping denominators (Lemma 3.6) and the reuse of intermediate products lower the step of [17] from $12.5N$ to $10N$ multiplications, plus the Nth root;
- all steps are inversion-free and come with exact operation counts (Sections 4.1–4.10), and the codomain is normalized without square roots;
- for models with a distinguished 2-torsion point, the normalization is a common root of two polynomials, with a characterization of the exceptional case (Proposition 4.10), whereas Huff curves, Kummer lines, Legendre curves and Jacobi intersections use norms over the kernel;
- we give a closed formula for the codomain of twisted Hessian curves in terms of the x-only quantity $q = X^2/(YZ)$ over half of the kernel, namely $a' = a^N$ and $d' = (Nd - 6a\sum q_i)/\prod q_i$ (Theorem 4.26), of which the radical step only needs the product $\prod q_i$ (Theorem 4.28);
- points can be pushed on every model, at a cost linear in N per point; on Montgomery and twisted Edwards curves this costs only $2N$ multiplications per point, without precomputation, which makes these the cheapest models for radical chains that transport torsion points, as quantified below.

Overview of costs. Table 1 summarizes the costs. "Radical isogeny step" is the cost of computing the codomain and the next kernel generator, excluding the Nth root. The cheaper "codomain only" variants for the last step of a chain, where no next generator is needed and hence no root is extracted, are given in Section 4. The powers B^N, a^N and so on in the various steps contribute multiples of $\mathbf{E}_N$.

			Point pushing	
Model	Condition on $E(K)$	Radical isogeny step	Precomp.	Per point
Tate normal form	none	$10N$	$2N - 7$	$2N + 16$
Short Weierstrass	none	$10N + 35$	$2.5N - 12.5$	$2N + 13$
Montgomery	$\mathbb{Z}/2\mathbb{Z}$	$10.5N + 52.5$	0	$2N$
Twisted Edwards	$\mathbb{Z}/2\mathbb{Z}$	$10.5N + 52.5$	0	$2N$
$E_{A,B}$, $B \neq 1$	$\mathbb{Z}/2\mathbb{Z}$	$11.5N + 2\mathbf{E}_N + 56.5$	0	$2.5N + 2.5$
Huff	$\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/4\mathbb{Z}$	$11.5N + 2\mathbf{E}_N + 62.5$	0	$2N + 4$
Kummer line	$\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/4\mathbb{Z}^\dagger$	$11.5N + 2\mathbf{E}_N + 62.5$	0	$2N + 4$
Jacobi quartic	$\mathbb{Z}/2\mathbb{Z}$	$11.5N + 2\mathbf{E}_N + 67.5$	0	$2.5N + 8.5$
Legendre	$\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$	$13.5N + 2\mathbf{E}_N + 64.5$	0	$2.5N + 4.5$
Jacobi intersection	$\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$	$13.5N + 2\mathbf{E}_N + 64.5$	0	$2.5N + 4.5$
Twisted Hessian	$\mathbb{Z}/3\mathbb{Z}$ ($3 \nmid N$)	$15.5N + 2\mathbf{E}_N + 60.5$	$N + \mathbf{E}_N + 1$	$3.5N + 6.5$

Table 1: Cost in $\mathbf{M}$ of one radical step of odd degree $N \geq 7$, excluding the Nth root. The condition is the rational subgroup that the model requires, besides the kernel $\langle P \rangle \cong \mathbb{Z}/N\mathbb{Z}$ (for †, on E or on its quadratic twist); $E_{A,B}\colon y^2 = x(x^2 + Ax + B)$, and $\mathbf{E}_N$ is as in the notation paragraph. In Tate normal form the counts hold for $N \geq 11$, on short Weierstrass and twisted Hessian curves for $N \geq 9$ (for the latter, $3 \nmid N$, so $N \geq 11$). "Precomp." is the shared precomputation per step for pushing points, "per point" the cost of pushing one x-only point. All counts are those of the algorithms in Sections 3 and 4; each model links to its subsection.

We draw the following conclusions from Table 1. The bulk of the work, about $6.5N$ $\mathbf{M}$, happens on the data of the Tate normal form and is the same for every model. The models differ

in two places. The first is the cost of the multiples of the kernel generator, which is $3N$ **M** with the Montgomery ladder, $3.5N$ **M** with the Tate recursion and $4N$ **M** for $y^2 = x(x^2 + Ax + B)$ with $B \neq 1$. The second is the cost of normalizing the codomain. For all models with a distinguished 2-torsion point, this is a product of $\frac{1}{2}(N-1)$ linear factors, i.e. $0.5N$ **M**. Huff curves and Kummer lines need an extra N **M**, and Legendre curves and Jacobi intersections an extra $2N$ **M**, to place the other 2-torsion points, whereas twisted Hessian curves need a product of a cubic rational function, i.e. $5N$ **M**. Short Weierstrass curves have the cheapest step after Tate normal form, since the Vélu codomain needs no normalization and the recursion gives Π_Δ^2 for free, and Montgomery and Edwards curves follow at about $0.5N$ **M** more. The latter become the best choice as soon as points have to be pushed along the walk, as for oriented curves or level structures, since they push a point at $2N$ multiplications without precomputation. With k pushed points per step, Table 1 gives a difference of $2N - 30 + 13k$ multiplications per step in their favor against short Weierstrass curves, which is positive for all $N \geq 9$ as soon as $k \geq 1$. Against Tate normal form the difference is $1.5N - 59.5 + 16k$, which is positive for $N \geq 31$ if $k = 1$, for $N \geq 19$ if $k = 2$, and for all $N \geq 11$ if $k \geq 3$.

Outline. Section 2 recalls the necessary background, including the radical isogeny formulae in Tate normal form. Section 3 proves the reflection symmetry and gives the model-independent reduction, and Section 4 derives the formulae for the different models. Section 5 describes the implementation.

2 Preliminaries

2.1 Weierstrass equations and changes of coordinates

We use general Weierstrass equations $E\colon y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$ with the usual quantities $b_2 = a_1^2 + 4a_2$, $b_4 = 2a_4 + a_1a_3$, $b_6 = a_3^2 + 4a_6$, and we put

$$F(x) = 4x^3 + b_2x^2 + 2b_4x + b_6, \qquad G(x) = 6x^2 + b_2x + b_4 = \tfrac{1}{2}F'(x).$$

Let ψ_k denote the kth division polynomial of E [38, Exercise 3.7]. Then F is the square of the 2-division polynomial, written as a polynomial in x: for $Q \in E$ we have $\psi_2(Q)^2 = (2y(Q) + a_1x(Q) + a_3)^2 = F(x(Q))$. For $Q \notin E[2]$, the values $F(x(Q))$ and $G(x(Q))$ are the quantities u_Q and v_Q of Vélu's formulae [40], a notation we avoid since u_Q denotes a scaling factor in Section 2.3. A change of coordinates is written $x = u^2x^* + r$, $y = u^3y^* + su^2x^* + t$, and $E^{(u,r,s,t)}$ denotes the resulting curve [38, Table 3.1]. Completing the square gives $\big(y + \frac{1}{2}(a_1x + a_3)\big)^2 = \frac{1}{4}F(x)$, so the x-coordinates of points of E are described by the monic cubic $C_E(x) = \frac{1}{4}F(x) = x^3 + \frac{b_2}{4}x^2 + \frac{b_4}{2}x + \frac{b_6}{4}$. Finally, for $k \geq 1$ with $[k]Q \neq \mathcal{O}$ we have $x([k]Q) = x(Q) - \psi_{k-1}\psi_{k+1}(Q)/\psi_k(Q)^2$, again by [38, Exercise 3.7]; in particular $x([2]Q) - x(Q) = -\psi_3(x(Q))/F(x(Q))$.

2.2 Vélu's formulae

For a finite subgroup $\mathcal{G} \subset E$ of odd order, Vélu's isogeny $\varphi\colon E \to E'$ with kernel $\mathcal{G}$ satisfies $x(\varphi(R)) = x(R) + \sum_{Q \in \mathcal{G}\setminus\{\mathcal{O}\}}\big(x(R+Q) - x(Q)\big)$, and E' has the same a_1, a_2, a_3 as E [40]. In Kohel's form [27],

$$x(\varphi(R)) = x + \sum_{Q \in \mathcal{Q}}\Big(\frac{G(x(Q))}{x - x(Q)} + \frac{F(x(Q))}{(x - x(Q))^2}\Big), \tag{1}$$

where $x = x(R)$ and $\mathcal{Q}$ contains one point of each pair $\{Q, -Q\} \subset \mathcal{G} \setminus \{\mathcal{O}\}$. Since F is a cubic and $G = \frac{1}{2}F'$, Taylor expansion at x gives $\frac{G(x(Q))}{x-x(Q)} + \frac{F(x(Q))}{(x-x(Q))^2} = \frac{F(x)}{(x-x(Q))^2} - \frac{G(x)}{x-x(Q)} + 2(x - x(Q))$,

hence, for $|\mathcal{G}| = N$,

$$x(\varphi(R)) = Nx - 2\sum_{Q\in\mathcal{Q}} x(Q) + F(x)\sum_{Q\in\mathcal{Q}} \frac{1}{(x-x(Q))^2} - G(x)\sum_{Q\in\mathcal{Q}} \frac{1}{x-x(Q)}, \tag{2}$$

in which F and G are only evaluated at x. We will use the following simple observation repeatedly.

Lemma 2.1 (Covariance)**.** *Let $E^* = E^{(u,r,s,t)}$, let $\iota\colon E^* \to E$ be the attached isomorphism, and let $\mathcal{G} \subset E$ be a finite subgroup. Let $\varphi\colon E \to E'$ and $\varphi^*\colon E^* \to E'^*$ be the Vélu isogenies with kernels $\mathcal{G}$ and $\iota^{-1}(\mathcal{G})$. Then $E'^* = E'^{(u,r,s,t)}$, and $\varphi\circ\iota = \iota'\circ\varphi^*$ for the isomorphism ι' attached to (u,r,s,t).*

Proof. In Vélu's formula, $x(R+Q) - x(Q)$ transforms as $u^2(x^*(R^*+Q^*) - x^*(Q^*))$, so $x(\varphi(\iota R^*)) = u^2 x(\varphi^*(R^*)) + r$. The same holds for y. Hence both maps agree on x and y, and the codomains are related by (u,r,s,t). □

2.3 Radical isogenies in Tate normal form

For $b, c \in K$ let $E_{b,c}\colon y^2 + (1-c)xy - by = x^3 - bx^2$. Every pair (E,Q) with Q of order at least 4 is isomorphic to $(E_{b,c},(0,0))$ for a unique pair (b,c), via the isomorphism attached to $(u_Q, x(Q), s_Q, y(Q))$, where s_Q is the slope of the tangent at Q and $u_Q = \psi_2(Q)/(x(Q) - x([2]Q))$ [17, Lemma 2.2]. Its b-invariants are

$$b_2 = (1-c)^2 - 4b, \qquad b_4 = -b(1-c), \qquad b_6 = b^2. \tag{3}$$

We recall the main result of [17] for odd N.

Theorem 2.2 ([17, Theorem 3.1, Proposition 3.2])**.** *Let $P = (0,0) \in E_{b,c}$ have order N, let $x_k = x([k]P)$ for $2 \le k \le N-2$ (so $x_{N-k} = x_k$), and let β be an Nth root of $\rho_N = \big(b^2\prod_{k=2}^{N-2} x_k\big)^{-1}$. Put $\alpha = -\beta$ and $h_i = \alpha^i \prod_{k=2}^{i} x_k$ for $1 \le i \le N-2$, and*

$$\mathcal{S}_1 = b\sum_{i=1}^{N-2} h_i, \qquad \mathcal{S}_{1,1} = b^2\sum_{i=1}^{N-2} h_i^2,$$

$$\mathcal{S}_2 = \frac{b}{\alpha}\sum_{i=1}^{N-3} h_i h_{i+1} - b\,h_{N-2}, \qquad \mathcal{S}_{1,2} = \frac{b^2}{\alpha}\sum_{i=1}^{N-3} h_i^2 h_{i+1},$$

$$\mathcal{S}_3 = \frac{1}{\alpha^3}\sum_{i=1}^{N-4} h_i h_{i+1} h_{i+2} - \frac{h_{N-2}^2}{\alpha^2} - \frac{h_{N-2}}{\alpha}.$$

Let $\varphi\colon E_{b,c} \to E'$ be the Vélu isogeny with kernel $\langle P\rangle$ and put

$$u = 1 + 3\mathcal{S}_1 - \mathcal{S}_3, \quad s = 2\mathcal{S}_1 - \mathcal{S}_2, \quad r = (\mathcal{S}_1 + \mathcal{S}_2)(1 - \mathcal{S}_3) + \tfrac{3}{2}(\mathcal{S}_1^2 - \mathcal{S}_{1,1}) - \mathcal{S}_{1,2},$$

and $t = \frac{1}{2}\big(u(3r - b - (1-c)s - s^2) + b - (1-c)r\big)$. Then $u \neq 0$ and $E'^{(u,r,s,t)} = E_{b',c'}$ with

$$b' = \frac{b + (1-c)s + s^2 - 3r}{u^2}, \qquad c' = 1 - \frac{1 - c + 2s}{u}.$$

Moreover $P' = (r,t) \in E'$ is P-distinguished, i.e. $\hat{\varphi}(P') = P$, and has order N, and the composition of φ with the next step is cyclic.

3 A model-independent radical step

This section contains the two ingredients shared by all models: a reflection symmetry of the radical formulae, and a reduction that runs them on the data of an arbitrary Weierstrass model and returns the codomain with the next generator. Algorithm 1 combines both.

3.1 A reflection symmetry

We will first show that the terms h_i of Theorem 2.2 come in pairs with a constant product, which halves the cost of several of its sums.

Lemma 3.1 (Reflection)**.** *In the setting of Theorem 2.2, $h_i h_{N-1-i} = -1/(\alpha b^2)$ for all $1 \le i \le N-2$. Consequently, any representation $h_i = e_i/\mathcal{D}$ with a common denominator satisfies $e_a e_b = \Lambda := e_1 e_{N-2}$ whenever $a+b = N-1$. Let $h = \frac{1}{2}(N-1)$, $p_i = e_i e_{i+1}$, and*

$$s_i = e_i + e_{N-1-i}, \qquad u_i = e_i + e_{N-2-i}, \qquad w_i = e_i + e_{N-3-i}.$$

Then

$$\sum_{i=1}^{N-2} e_i = \sum_{i=1}^{h-1} s_i + e_h, \qquad \sum_{i=1}^{N-2} e_i^2 = \sum_{i=1}^{h-1} s_i^2 - (N-4)\Lambda,$$
$$\sum_{i=1}^{N-3} e_i^2 e_{i+1} = \sum_{i=1}^{h-1} u_i\,(p_i + p_{N-2-i}) - \Lambda \sum_{i=1}^{h-1} u_i,$$
$$\sum_{i=1}^{N-4} e_i e_{i+1} e_{i+2} = \sum_{i=1}^{h-2} w_i\,(p_{i+1} + p_{N-2-i}) - \Lambda\Big(\sum_{i=2}^{h-1} s_i - e_h\Big).$$

Proof. Using $x_{N-k} = x_k$ we get $\prod_{k=2}^{N-1-i} x_k = \prod_{k=i+1}^{N-2} x_k$, hence $h_i h_{N-1-i} = \alpha^{N-1} \prod_{k=2}^{N-2} x_k = -1/(\alpha b^2)$, because $\alpha^N = -\beta^N = -\rho_N$ for odd N. Multiplying by $\mathcal{D}^2$ gives $e_i e_{N-1-i} = e_1 e_{N-2}$. For the sums, we split the terms into pairs. In the first two sums, the pairs are $\{i, N-1-i\}$ for $1 \le i \le h-1$, and the middle term $i = h$ satisfies $e_h^2 = \Lambda$; each pair contributes s_i, respectively $s_i^2 - 2\Lambda$. In the third sum, the terms with indices i and $N-2-i$ form a pair for $1 \le i \le h-1$, and these pairs cover all $N-3$ terms. Expanding,

$$\begin{aligned} u_i(p_i + p_{N-2-i}) &= e_i^2 e_{i+1} + e_{N-2-i}^2 e_{N-1-i} + e_{N-2-i} \cdot e_i e_{N-1-i} + e_i \cdot e_{i+1} e_{N-2-i} \\ &= e_i^2 e_{i+1} + e_{N-2-i}^2 e_{N-1-i} + \Lambda u_i, \end{aligned}$$

since the two cross terms contain $e_i e_{N-1-i} = \Lambda$ and $e_{i+1} e_{N-2-i} = \Lambda$. In the fourth sum, the terms with indices i and $N-3-i$ form a pair for $1 \le i \le h-2$, and the remaining term $i = h-1$ equals $e_h \cdot e_{h-1} e_{h+1} = \Lambda e_h$. Expanding in the same way,

$$\begin{aligned} w_i(p_{i+1} + p_{N-2-i}) &= e_i e_{i+1} e_{i+2} + e_{N-3-i} e_{N-2-i} e_{N-1-i} + e_{N-2-i} \cdot e_i e_{N-1-i} \\ &\quad + e_{i+1} \cdot e_{i+2} e_{N-3-i} \\ &= e_i e_{i+1} e_{i+2} + e_{N-3-i} e_{N-2-i} e_{N-1-i} + \Lambda s_{i+1}, \end{aligned}$$

and summing over the pairs gives the claim. □

The reflection thus lets us group the terms of each sum in pairs whose sum is, up to a multiple of Λ, a single product of a sum of two e_i with a sum of two products p_i. In particular, once the p_i are known, Σe_i^2, $\Sigma e_i^2 e_{i+1}$ and $\Sigma e_i e_{i+1} e_{i+2}$ cost one multiplication per pair instead of one per term.

3.2 A model-independent reduction

Let E be given by a Weierstrass equation and let $P = (x_1, y_1) \in E(K)$ have order N. We write $x_k = x([k]P)$ and $\hat{x}_k = x_k - x_1$, so that $\hat{x}_1 = 0$ and $\hat{x}_{N-k} = \hat{x}_k$.

Lemma 3.2 (Tate data from $x(P)$)**.** *The Tate normal form (b, c) of (E, P) and the attached isomorphism satisfy*

$$u_P^2 = \frac{F(x_1)}{\hat{x}_2^2}, \qquad x^{\mathrm{Tate}}(Q) = \frac{x(Q) - x_1}{u_P^2}, \qquad b = \frac{\hat{x}_2^3}{F(x_1)}, \qquad 1 - c = -\frac{G(x_1)\,\hat{x}_2}{F(x_1)}.$$

In particular all of them only depend on x_1, and the Tate coordinates of $[k]P$ are $\hat{x}_k/u_P^2$.

Proof. By [17, Lemma 2.2], $u_P = \psi_2(P)/(x_1 - x_2)$ and $x = u_P^2 x^{\mathrm{Tate}} + x_1$, which gives the first two identities since $\psi_2(P)^2 = F(x_1)$. Then $b = x^{\mathrm{Tate}}([2]P) = \hat{x}_2/u_P^2$. Finally $1 - c = (a_1 + 2s_P)/u_P$ with $a_1 + 2s_P = G(x_1)/\psi_2(P)$, as in the proof of [17, Proposition 5.5]. □

Lemma 3.2 makes the approach x-only and model-independent. Indeed, only u_P^2 occurs, never an odd power of u_P, which would involve $\psi_2(P)$ and hence y_1. Therefore any model in which $x(P)$ and its multiples can be computed can call the radical formulae of Theorem 2.2 without ever constructing the Tate normal form, or a y-coordinate, explicitly.

Lemma 3.3 (Shifted recursion)**.** *For $2 \le k \le N - 3$,*

$$\hat{x}_{k+1} + \hat{x}_{k-1} = \frac{F(x_1) + G(x_1)\,\hat{x}_k}{\hat{x}_k^2}.$$

Proof. In the model $E^{(1, x_1, s_P, y_1)}$ the point P becomes $(0, 0)$, the coefficients a_4 and a_6 vanish, and x-coordinates become $\hat{x}$. By [17, Lemma 5.1], $\hat{x}(Q+P) + \hat{x}(Q-P) = \hat{a}_3(\hat{a}_3 + \hat{a}_1\hat{x}(Q))/\hat{x}(Q)^2$ there, with $\hat{a}_3 = \psi_2(P)$ and $\hat{a}_1 = a_1 + 2s_P$. Hence $\hat{a}_3^2 = F(x_1)$ and $\hat{a}_1\hat{a}_3 = G(x_1)$. □

Projectively, with $\hat{x}_k = X_k/D_k$, the recursion costs $7\mathbf{M}$ per term as in [17, Algorithm 5.2]: $X_{k+1} = D_k\big(F(x_1)D_k + G(x_1)X_k\big)D_{k-1} - X_{k-1}X_k^2$ and $D_{k+1} = X_k^2 D_{k-1}$. Together with Lemma 3.2, it feeds the radical formulae for any model, given only $x(P)$. It remains to describe the codomain in the original model.

Proposition 3.4 (Vélu codomain from the Tate output)**.** *Let $E\colon y^2 = x^3 + a_2x^2 + a_4x + a_6$ and let $\varphi\colon E \to E'$ be the Vélu isogeny with kernel $\langle P\rangle$. Let (u, r, s, t) and (b', c') be as in Theorem 2.2 for the Tate normal form of (E, P), let b_2', b_4', b_6' be the invariants (3) of $E_{b',c'}$, and put*

$$x_1' = x_1 + u_P^2\, r, \qquad \Omega = u_P^2 u^2.$$

Then $x(P') = x_1'$ for the P-distinguished point P' of Theorem 2.2, the map $x = \Omega x^* + x_1'$ identifies x-coordinates on $E_{b',c'}$ and on E', and

$$E'\colon\ y^2 = (x - x_1')^3 + \tfrac{b_2'}{4}\,\Omega\,(x - x_1')^2 + \tfrac{b_4'}{2}\,\Omega^2(x - x_1') + \tfrac{b_6'}{4}\,\Omega^3.$$

In particular $a_2 = -3x_1' + \frac{b_2'}{4}\Omega$.

Proof. By Lemma 2.1, the Vélu codomain of the Tate normal form is $E'^{(u_P, x_1, s_P, y_1)}$, and by Theorem 2.2 its transform by (u, r, s, t) is $E_{b',c'}$. Composing the two changes of coordinates, $x = u_P^2(u^2x^* + r) + x_1 = \Omega x^* + x_1'$, and $(0, 0) \in E_{b',c'}$ maps to $x = x_1'$. Since E' has $a_1 = a_3 = 0$, its equation is the completed square $y^2 = C_{E_{b',c'}}(x^*)$ rescaled by $x = \Omega x^* + x_1'$. □

Proposition 3.4 turns the output of the radical formulae, a new Tate normal form, into concrete data on E': the affine map $x = \Omega x^* + x'_1$, the next generator $x(P') = x'_1$, and the equation of E'. Since Ω and x'_1 only involve u_P^2, all of this is a rational function of x_1 and the radical β, without square roots or y-coordinates. Each model then only needs its own normalization of the codomain, i.e. the change of coordinates betwixt the Vélu codomain and the target model.

Theorem 2.2 is stated in Tate coordinates, in which b, c, the x_k^{T}, the root β and the sums $\mathcal{S}_1, \ldots, \mathcal{S}_{1,2}$ are all fractions. Evaluating them naively would therefore require the Tate normal form and many inversions. Algorithm 1 instead takes projective data in the coordinates of E, which every model obtains from its own arithmetic, and returns numerators $g, \mathcal{R}, \mathcal{U}, \mathcal{S}, \mathcal{R}_d, \bar{a}, \bar{b}, \bar{c}$; every quantity of Theorem 2.2 is one of them divided by an explicit product of them. It is the part of [17, Algorithm 5.2] that only uses the Tate data, and every algorithm in Section 4 calls it. Its input essentially consists of the x-coordinates of half of the kernel: the differences $x([k]P) - x(P) = \Delta_k/(D_k Z_1)$ for $2 \le k \le \frac{1}{2}(N-1)$, in projective form, together with the values F_n and G_n of F and G at $x(P)$ and the common denominator C of the curve coefficients. We write $h = \frac{1}{2}(N-1)$ and mirror indices by $k \mapsto N-k$. Each numbered line groups related assignments, one per row, with the cost of each assignment in $\mathbf{M}$ on the right; a product that occurs in several assignments is computed once and counted where it first occurs. Throughout the algorithms, we will use the following conventions. A running product carries the name of its factors, e.g. $\Pi_{\eta,k} = \eta_2 \cdots \eta_k$ or $\Pi_{X,k} = X_1 \cdots X_k$, and drops the second index for the full product over half of the kernel, e.g. $\Pi_\Delta = \Pi_{\Delta,h}$. A curve coefficient given projectively carries the subscript n over a common denominator C (e.g. $A = A_n/C$), primes denote the codomain, $(X_i : Z_i)$ are the multiples $x([i]P)$, $(X_R : Z_R)$ is a point to be pushed, and $t_0, t_1, \ldots$ (and s_i, t_i, u_i inside a product over i) are temporary values. Proofs freely use fractions, such as the expressions for r, s, u and Ω in Proposition 3.5, to derive the formulae; the algorithms only compute their numerators and denominators, and every output is a projective tuple, so no field element computed during a step is ever inverted (only the fixed constants mentioned in the notation paragraph are).

We use the following naming convention. Calligraphic letters denote numerators and denominators of quantities of Theorem 2.2. The numerators are $\mathcal{R}$, $\mathcal{S}$ and $\mathcal{U}$ for r, s and u, $\mathcal{B}$ for b', and $\mathcal{W}$ for $1 - c + 2s$, which gives $1 - c' = 2g\mathcal{W}/(F_n\mathcal{U})$. The denominators are $\mathcal{D}$, common to all terms h_i, and $\mathcal{R}_d$, in $x(P') - x_1 = F_n\mathcal{R}/(CZ_1\mathcal{R}_d)$. Finally, $\bar{a}, \bar{b}, \bar{c}$ are the coefficients of the codomain cubic, scaled so that they are polynomial. The exact relations are given in Proposition 3.5, and Table 2 collects the notation.

Proposition 3.5. *Let the input of Algorithm 1 come from (E, P) as stated, with $x_1 = X_1/Z_1$. Then the point P' of Theorem 2.2 has x-coordinate*

$$x'_1 = x_1 + u_P^2 r = x_1 + \frac{F_n \mathcal{R}}{C Z_1 \mathcal{R}_d}$$

on the Vélu codomain (Proposition 3.4), and $(\bar{a}, \bar{b}, \bar{c}) = (\vartheta\alpha_2, \vartheta^2\alpha_1, \vartheta^3\alpha_0)$ for the cubic $x^3 + \alpha_2 x^2 + \alpha_1 x + \alpha_0$ of $E_{b',c'}$ and $\vartheta = 2D_2 F_n^2 \mathcal{U}^2$. Here

$$r = \frac{\mathcal{R}}{2D_2^2 g^5}, \qquad s = \frac{\mathcal{S}}{D_2 g^2}, \qquad u = \frac{\mathcal{U}}{D_2 g^3}, \qquad \Omega = u_P^2 u^2 = \frac{F_n \mathcal{U}^2}{C Z_1 \Delta_2^2 g^6}$$

are the quantities of Theorem 2.2 and Proposition 3.4, for the Nth root $\beta = -\gamma\chi/\mu$ of ρ_N, where $\chi = F_n D_2^2/(C\Delta_2^2)$. Given Π_Δ^2, lines 1–8 cost $\frac{1}{2}(13N + 21)$ $\mathbf{M}$ and lines 9–10 cost 28 $\mathbf{M}$. If $C = 1$, both counts drop by one, and if $D_2 = 1$, they drop by two and three, respectively. As a product, Π_Δ^2 costs $\frac{1}{2}(N-3)$ $\mathbf{M}$; Lemma 3.6 gives it more cheaply for some models.

Proof. Throughout, let $h = \frac{1}{2}(N-1)$, and extend Δ_k, D_k, Γ_k and η_k to $h < k \le N-2$ by $k \mapsto N-k$. This is consistent with $x_{N-k} = x_k$, and it gives $\prod_{k=2}^{N-2} \Delta_k = \Pi_\Delta^2$ and $\prod_{k=2}^{N-2} D_k = \Pi_D^2$.

Algorithm 1 Radical core

Input: $F_n, G_n, C \in K$, Π_Δ^2 and $(\Delta_k, D_k)_{2\le k\le h}$ such that $F(x_1) = F_n/(CZ_1^3)$, $G(x_1) = 2G_n/(CZ_1^2)$ and $x_k - x_1 = \Delta_k/(D_k Z_1)$ for some $Z_1 \in K^*$, and $\Pi_\Delta = \prod_{k=2}^h \Delta_k$

Output: $g, \mathcal{R}, \mathcal{U}, \mathcal{S}, \mathcal{R}_d$ and $(\bar{a}, \bar{b}, \bar{c})$ as in Proposition 3.5

$\mu \leftarrow F_n \Pi_\Delta^2$	1
$\eta_k \leftarrow \mu D_k$ for $2 \le k \le h$	$\frac{1}{2}(N-3)$
$\Pi_{\eta,2} \leftarrow \eta_2$, $\Pi_{\eta,k} \leftarrow \Pi_{\eta,k-1}\eta_k$ for $3 \le k \le h$	$\frac{1}{2}(N-5)$
$\gamma \leftarrow \big(-C(\mu\Pi_{\eta,h})^2\big)^{1/N}$	3, 1 root
$\Gamma_k \leftarrow \gamma\Delta_k$ for $2 \le k \le h$	$\frac{1}{2}(N-3)$
$e_i \leftarrow \gamma \prod_{k=2}^{i} \Gamma_k \cdot \prod_{k=2}^{N-1-i} \eta_k$ for $1 \le i \le N-2$, continuing the $\Pi_{\eta,k}$	$\frac{1}{2}(5N-15)$
$\Lambda \leftarrow e_1 e_{N-2}$, $p_i \leftarrow e_i e_{i+1}$ for $1 \le i \le N-3$	$N-2$
$s_i, u_i \leftarrow e_i + e_{N-1-i}$, $e_i + e_{N-2-i}$ for $1 \le i \le h-1$,	0
$w_i \leftarrow e_i + e_{N-3-i}$ for $1 \le i \le h-2$	
$\Sigma_1 \leftarrow \sum_{i=1}^{h-1} s_i + e_h$, $\Sigma_2 \leftarrow \sum_{i=1}^{N-3} p_i$	0
$\Sigma_{11} \leftarrow \sum_{i=1}^{h-1} s_i^2 - (N-4)\Lambda$	$\frac{1}{2}(N-3)$
$\Sigma_{12} \leftarrow \sum_{i=1}^{h-1} u_i(p_i + p_{N-2-i}) - \Lambda \sum_{i=1}^{h-1} u_i$	$\frac{1}{2}(N-1)$
$\Sigma_3 \leftarrow \sum_{i=1}^{h-2} w_i(p_{i+1} + p_{N-2-i}) - \Lambda\big(\sum_{i=2}^{h-1} s_i - e_h\big)$	$\frac{1}{2}(N-3)$
$g \leftarrow \mu e_1$	1
$V \leftarrow \mu^2\Sigma_3 - \mu g e_{N-2}^2 - g^2 e_{N-2}$	7
$T_1 \leftarrow g(\Sigma_1 - e_{N-2}) + \mu\Sigma_2$	2
$T_2 \leftarrow g^3 - \mu V$	2
$\mathcal{U} \leftarrow D_2 T_2 + 3\Gamma_2\Sigma_1 g^2$	3
$\mathcal{S} \leftarrow \Gamma_2\big(g(2\Sigma_1 + e_{N-2}) - \mu\Sigma_2\big)$	2
$\mathcal{R} \leftarrow 2\Gamma_2 T_1 \cdot D_2 T_2 + \Gamma_2^2 g^2\big(3(\Sigma_1^2 - \Sigma_{11})g - 2\mu\Sigma_{12}\big)$	8
$\mathcal{W} \leftarrow F_n\mathcal{S} - G_n\Delta_2 g^2$	3
$\mathcal{R}_d \leftarrow 2\Delta_2^2 g^5$	3
$\mathcal{B} \leftarrow C\Delta_2\mathcal{R}_d + D_2\big(2F_n\mathcal{S}^2 g - 4G_n\Delta_2\mathcal{S}g^3 - 3F_n\mathcal{R}\big)$	8
$\bar{a} \leftarrow g\big(2D_2 g\mathcal{W}^2 - F_n\mathcal{B}\big)$	5
$\bar{b} \leftarrow -2D_2F_n^2g^2 \cdot \mathcal{B}\mathcal{U} \cdot \mathcal{W}$	6
$\bar{c} \leftarrow \frac{1}{2}D_2F_n^2g^2 \cdot F_n^2(\mathcal{B}\mathcal{U})^2$	3

Symbol	Meaning	Defined in
h	$\frac{1}{2}(N-1)$, the number of multiples used	
x_k, $\hat{x}_k$	$x([k]P)$ and $x_k - x_1$, so that $\hat{x}_{N-k} = \hat{x}_k$	Section 3.2
u_P	scaling factor of the isomorphism from (E, P) to its Tate normal form	Lemma 3.2
F_n, G_n, C	$F(x_1) = F_n/(CZ_1^3)$ and $G(x_1) = 2G_n/(CZ_1^2)$	Algorithm 1
Δ_k, D_k	$\hat{x}_k = \Delta_k/(D_k Z_1)$ for $2 \le k \le h$	Algorithm 1
Π_Δ, Π_D	$\prod_{k=2}^{h} \Delta_k$ and $\prod_{k=2}^{h} D_k$	Lemma 3.6
μ, η_k	$F_n \Pi_\Delta^2$ and μD_k	line 1
γ, Γ_k	the Nth root, and $\gamma \Delta_k$	lines 2–3
e_i, $\mathcal{D}$	numerators and common denominator of the h_i of Theorem 2.2: $h_i = e_i/\mathcal{D}$	line 4
Λ, p_i	$e_1 e_{N-2}$ and $e_i e_{i+1}$	Lemma 3.1
s_i, u_i, w_i	pair sums $e_i + e_{N-1-i}$, $e_i + e_{N-2-i}$, $e_i + e_{N-3-i}$	Lemma 3.1
$\Sigma_1, \Sigma_2, \Sigma_3, \Sigma_{11}, \Sigma_{12}$	$\sum e_i$, $\sum e_i e_{i+1}$, $\sum e_i e_{i+1} e_{i+2}$, $\sum e_i^2$, $\sum e_i^2 e_{i+1}$	line 5
g	μe_1; the outputs are expressed over powers of g and D_2	line 6
$\mathcal{U}$, $\mathcal{S}$, $\mathcal{R}$	numerators of u, s, r of Theorem 2.2	line 8
$\mathcal{W}$, $\mathcal{B}$	numerators of $1 - c + 2s$ and of b'	line 9
$\mathcal{R}_d$	denominator in $x_1' - x_1 = F_n \mathcal{R}/(CZ_1 \mathcal{R}_d)$	line 9
$\bar{a}$, $\bar{b}$, $\bar{c}$	coefficients of the cubic of $E_{b',c'}$, with x scaled by ϑ	line 10
ϑ	$2D_2 F_n^2 \mathcal{U}^2$	Proposition 3.5
Ω, x_1'	affine map $x = \Omega x^* + x_1'$ from $E_{b',c'}$ to the Vélu codomain; $x_1' = x(P')$	Proposition 3.4

Table 2: Notation of Section 3.2 and Algorithm 1; "line" refers to Algorithm 1. A running product $\Pi_{\cdot,k}$ is the product of the first factors up to index k, and primes denote the codomain.

The proof follows three layers of scaling: the passage from E to Tate coordinates (a factor χ), a common denominator $\mathcal{D}$ for the terms h_i of Theorem 2.2, and the normalization by powers of g and D_2 in which all outputs are expressed. The first two layers cancel completely, which is why the algorithm never needs them.

Step 0: Tate data in the input format. By Lemma 3.2, $u_P^2 = F(x_1)/\hat{x}_2^2$. The input format gives $F(x_1) = F_n/(CZ_1^3)$ and $\hat{x}_2 = \Delta_2/(D_2 Z_1)$, hence

$$u_P^2 = \frac{F_n D_2^2}{CZ_1 \Delta_2^2} = \frac{\chi}{Z_1}, \qquad \chi = \frac{F_n D_2^2}{C\Delta_2^2}.$$

Again by Lemma 3.2, the Tate coordinates of the multiples are $x_k^{\mathrm{T}} = \hat{x}_k/u_P^2 = \Delta_k/(D_k \chi)$, i.e. the input fractions Δ_k/D_k divided by one common factor χ. In the same way,

$$b = x_2^{\mathrm{T}} = \frac{\Delta_2}{D_2 \chi} = \frac{C\Delta_2^3}{D_2^3 F_n}, \qquad 1 - c = -\frac{G(x_1)\,\hat{x}_2}{F(x_1)} = -\frac{2G_n \Delta_2}{D_2 F_n}.$$

The factor χ is a fraction, and the algorithm never computes it; the next steps show that it cancels.

Step 1: the radical (lines 1–2). Using the mirror symmetry, the product of all Tate coordinates is a square,

$$\prod_{k=2}^{N-2} x_k^{\mathrm{T}} = \frac{\Pi_\Delta^2}{\Pi_D^2 \chi^{N-3}}, \qquad \text{so} \qquad \rho_N = \Big(b^2 \prod_{k=2}^{N-2} x_k^{\mathrm{T}}\Big)^{-1} = \frac{D_2^2 \Pi_D^2 \chi^{N-1}}{\Delta_2^2 \Pi_\Delta^2}.$$

Since $D_2^2/\Delta_2^2 = C\chi/F_n$, this simplifies to $\rho_N = C\Pi_D^2 \chi^N/\mu$ with $\mu = F_n \Pi_\Delta^2$. The root can be computed cheaply for two reasons. First, the factor χ^N is an Nth power, so it can be taken

out of the root. Second, dividing by μ is the same as multiplying by μ^{N-1}, up to the Nth power μ^N. We therefore extract the root of the polynomial expression $-C\Pi_D^2\mu^{N-1}$ instead of ρ_N. Furthermore, line 1 computes $\eta_k = \mu D_k$, so the running product $\Pi_{\eta,h} = \eta_2 \cdots \eta_h$ of line 2 equals $\mu^{h-1}\Pi_D$, and since $N-1 = 2h$,

$$-C\Pi_D^2\mu^{N-1} = -C\big(\mu^{h-1}\Pi_D\big)^2\mu^2 = -C(\mu\Pi_{\eta,h})^2.$$

Consequently, line 2 needs neither Π_D nor the power μ^{N-1}, only the running products $\Pi_{\eta,k}$, which line 4 needs in any case. Explicitly, $\alpha := \gamma\chi/\mu$ satisfies

$$\alpha^N = \frac{\gamma^N\chi^N}{\mu^N} = -\frac{C\Pi_D^2\mu^{N-1}\chi^N}{\mu^N} = -\rho_N,$$

so that $\beta = -\alpha$ is an Nth root of ρ_N, as N is odd. From now on we use this root in Theorem 2.2; other choices of γ correspond to the other Nth roots of ρ_N.

Step 2: a common denominator for the h_i (lines 3–4). Substituting $\alpha = \gamma\chi/\mu$ and $x_k^{\mathrm{T}} = \Delta_k/(D_k\chi)$ into $h_i = \alpha^i \prod_{k=2}^{i} x_k^{\mathrm{T}}$, all powers of χ but one cancel:

$$h_i = \frac{\gamma^i\chi}{\mu^i}\prod_{k=2}^{i}\frac{\Delta_k}{D_k}.$$

The denominator $\mu^i\prod_{k=2}^{i} D_k$ still depends on i. Multiplying numerator and denominator by $\mu^{N-2-i}\prod_{k=i+1}^{N-2} D_k$, and using the mirror symmetry $\prod_{k=i+1}^{N-2} D_k = \prod_{k=2}^{N-1-i} D_k$, we obtain

$$h_i = \frac{e_i}{\mathcal{D}}, \qquad e_i = \gamma^i\mu^{N-2-i}\prod_{k=2}^{i}\Delta_k\prod_{k=2}^{N-1-i}D_k = \gamma\prod_{k=2}^{i}\Gamma_k\cdot\prod_{k=2}^{N-1-i}\eta_k, \qquad \mathcal{D} = \frac{\mu^{N-2}\Pi_D^2}{\chi},$$

where $\mathcal{D}$ no longer depends on i. Thus each e_i is a product of γ, of $i-1$ factors $\Gamma_k = \gamma\Delta_k$ and of $N-2-i$ factors $\eta_k = \mu D_k$, and passing from e_{i-1} to e_i trades one factor η for one factor Γ:

$$\frac{e_i}{e_{i-1}} = \frac{\Gamma_i}{\eta_{N-i}} = \frac{\gamma\Delta_i}{\mu D_i} = \alpha x_i^{\mathrm{T}} = \frac{h_i}{h_{i-1}}.$$

The e_i are thus the h_i with all denominators moved into the single factor $\mathcal{D}$. Line 4 computes them from the running products $\gamma\Gamma_2\cdots\Gamma_i$ and $\eta_2\cdots\eta_j$ (the latter continuing the $\Pi_{\eta,k}$ of line 2), each computed once, so that each e_i costs at most one further multiplication.

Step 3: the sums (line 5). With $h_i = e_i/\mathcal{D}$, the sums of the h_i in Section 2.3 become the five sums

$$\Sigma_1 = \sum e_i, \quad \Sigma_{11} = \sum e_i^2, \quad \Sigma_2 = \sum e_ie_{i+1}, \quad \Sigma_3 = \sum e_ie_{i+1}e_{i+2}, \quad \Sigma_{12} = \sum e_i^2e_{i+1}$$

of the e_i, divided by powers of $\mathcal{D}$ (made explicit in Step 5). Evaluated term by term, they cost about $3.5N$ $\mathbf{M}$: the $N-3$ products $p_i = e_ie_{i+1}$, which give Σ_2, and one further multiplication per term of Σ_{11}, Σ_3 and Σ_{12}. Line 5 halves the last three with Lemma 3.1, which applies because the e_i are the h_i over a common denominator, so that $e_ae_b = \Lambda = e_1e_{N-2}$ whenever $a+b = N-1$. Instead of term by term, the sums are then evaluated pair by pair, at one multiplication per pair, and the correction terms are added up first and multiplied by Λ once.

Step 4: absorbing the denominators (line 6). Line 6 computes $g = \mu e_1 = \gamma\mu\prod_{k=2}^{N-2}\eta_k = \gamma\mu^{N-2}\Pi_D^2$ which reuses $e_1 = \gamma\prod_{k=2}^{N-2}\eta_k$ from line 4. Comparing with Steps 1 and 2,

$$\alpha\mu\mathcal{D} = \frac{\gamma\chi}{\mu}\cdot\mu\cdot\frac{\mu^{N-2}\Pi_D^2}{\chi} = g, \qquad \text{so} \qquad \frac{1}{\mathcal{D}} = \frac{\alpha\mu}{g},$$

and, writing $\tilde\Gamma = \Gamma_2/D_2 = \Delta_2\gamma/D_2$ for the rest of the proof,

$$b\alpha = \frac{\Delta_2}{D_2\chi}\cdot\frac{\gamma\chi}{\mu} = \frac{\tilde\Gamma}{\mu}.$$

These two identities suffice, since they express the fractions $\mathcal{D}$, α and b, which the algorithm does not know individually, in terms of g, μ and $\tilde\Gamma$, and χ disappears.

Step 5: the sums $\mathcal{S}_i$, $\mathcal{S}_{i,j}$ and the quantities u, s, r (lines 7–8). We substitute $h_i = e_i\alpha\mu/g$ into the definitions of Section 2.3 and use $b\alpha = \tilde\Gamma/\mu$; in every term the powers of α combine with b or cancel. For instance $\mathcal{S}_1 = b\sum h_i = b\alpha\mu\Sigma_1/g = \tilde\Gamma\Sigma_1/g$, and

$$\mathcal{S}_3 = \frac{\mu^3\Sigma_3}{g^3} - \frac{\mu^2 e_{N-2}^2}{g^2} - \frac{\mu e_{N-2}}{g} = \frac{\mu V}{g^3}, \qquad V = \mu^2\Sigma_3 - \mu g e_{N-2}^2 - g^2 e_{N-2},$$

The same computation gives

$$\mathcal{S}_1 = \frac{\tilde\Gamma\Sigma_1}{g}, \quad \mathcal{S}_2 = \frac{\tilde\Gamma(\mu\Sigma_2 - g e_{N-2})}{g^2}, \quad \mathcal{S}_{1,1} = \frac{\tilde\Gamma^2\Sigma_{11}}{g^2}, \quad \mathcal{S}_3 = \frac{\mu V}{g^3}, \quad \mathcal{S}_{1,2} = \frac{\tilde\Gamma^2\mu\Sigma_{12}}{g^3}.$$

The combinations that occur in Theorem 2.2 are $\mathcal{S}_1 + \mathcal{S}_2 = \tilde\Gamma T_1/g^2$ and $1 - \mathcal{S}_3 = T_2/g^3$, with $T_1 = g(\Sigma_1 - e_{N-2}) + \mu\Sigma_2$ and $T_2 = g^3 - \mu V$ as in line 7. Hence

$$u = 1 + 3\mathcal{S}_1 - \mathcal{S}_3 = \frac{T_2 + 3\tilde\Gamma\Sigma_1 g^2}{g^3} = \frac{\mathcal{U}}{D_2 g^3}, \quad s = 2\mathcal{S}_1 - \mathcal{S}_2 = \frac{\tilde\Gamma\big(g(2\Sigma_1 + e_{N-2}) - \mu\Sigma_2\big)}{g^2} = \frac{\mathcal{S}}{D_2 g^2},$$

with $\mathcal{U}$ and $\mathcal{S}$ as in line 8, and, over the common denominator $2D_2^2 g^5$,

$$r = (\mathcal{S}_1 + \mathcal{S}_2)(1 - \mathcal{S}_3) + \tfrac{3}{2}(\mathcal{S}_1^2 - \mathcal{S}_{1,1}) - \mathcal{S}_{1,2} = \frac{\tilde\Gamma T_1 T_2}{g^5} + \frac{3\tilde\Gamma^2(\Sigma_1^2 - \Sigma_{11})}{2g^2} - \frac{\tilde\Gamma^2\mu\Sigma_{12}}{g^3} = \frac{\mathcal{R}}{2D_2^2 g^5},$$

with $\mathcal{R}$ as in line 8.

Step 6: the next generator. By Proposition 3.4, $x(P') = x_1 + u_P^2 r$ on the Vélu codomain. With $u_P^2 = \chi/Z_1$ from Step 0 and r from Step 5, the factor D_2^2 cancels:

$$x_1' = x_1 + \frac{F_n D_2^2}{C Z_1 \Delta_2^2}\cdot\frac{\mathcal{R}}{2D_2^2 g^5} = x_1 + \frac{F_n\mathcal{R}}{C Z_1 \mathcal{R}_d}, \qquad \mathcal{R}_d = 2\Delta_2^2 g^5.$$

In the same way $\Omega = u_P^2 u^2 = F_n\mathcal{U}^2/(C Z_1 \Delta_2^2 g^6)$.

Step 7: the codomain cubic (lines 9–10). By Theorem 2.2, $1 - c' = (1 - c + 2s)/u$ and $b' = (b + (1-c)s + s^2 - 3r)/u^2$. With Step 0 and Step 5,

$$1 - c + 2s = -\frac{2G_n\Delta_2}{D_2 F_n} + \frac{2\mathcal{S}}{D_2 g^2} = \frac{2\mathcal{W}}{D_2 F_n g^2}, \qquad \text{so} \qquad 1 - c' = \frac{2g\mathcal{W}}{F_n\mathcal{U}},$$

with $\mathcal{W} = F_n\mathcal{S} - G_n\Delta_2 g^2$. Multiplying $b + (1-c)s + s^2 - 3r$ by $2D_2^3 F_n g^5$ turns its four terms into $2C\Delta_2^3 g^5 = C\Delta_2\mathcal{R}_d$, $-4G_n\Delta_2\mathcal{S}D_2 g^3$, $2F_n\mathcal{S}^2 D_2 g$ and $-3F_n\mathcal{R}D_2$, whose sum is $\mathcal{B}$; hence $b' = \mathcal{B}g/(2D_2 F_n\mathcal{U}^2)$. By (3), the cubic of $E_{b',c'}$ has

$$\alpha_2 = \tfrac{1}{4}\big((1-c')^2 - 4b'\big), \qquad \alpha_1 = -\tfrac{1}{2}b'(1-c'), \qquad \alpha_0 = \tfrac{1}{4}b'^2,$$

which still have the denominators $F_n^2\mathcal{U}^2$ and powers of D_2. Scaling x by $\vartheta = 2D_2 F_n^2\mathcal{U}^2$ clears them:

$$\vartheta\alpha_2 = \tfrac{1}{2}D_2 F_n^2\mathcal{U}^2\Big(\frac{4g^2\mathcal{W}^2}{F_n^2\mathcal{U}^2} - \frac{2\mathcal{B}g}{D_2 F_n\mathcal{U}^2}\Big) = g\big(2D_2 g\mathcal{W}^2 - F_n\mathcal{B}\big) = \bar a,$$

$$\vartheta^2\alpha_1 = -2D_2^2 F_n^4\mathcal{U}^4\cdot\frac{\mathcal{B}g}{2D_2 F_n\mathcal{U}^2}\cdot\frac{2g\mathcal{W}}{F_n\mathcal{U}} = -2D_2 F_n^2 g^2\,\mathcal{B}\mathcal{U}\,\mathcal{W} = \bar b,$$

$$\vartheta^3\alpha_0 = 2D_2^3 F_n^6\mathcal{U}^6\cdot\frac{\mathcal{B}^2 g^2}{4D_2^2 F_n^2\mathcal{U}^4} = \tfrac{1}{2}D_2 F_n^4 g^2(\mathcal{B}\mathcal{U})^2 = \bar c.$$

The rescaled cubic $x^3 + \bar{a}x^2 + \bar{b}x + \bar{c}$ has the roots of the cubic of $E_{b',c'}$ multiplied by ϑ, and ratios such as $\bar{b}/\bar{a}^2 = \alpha_1/\alpha_2^2$ are unchanged. Since ϑ is known exactly, every model can work with $(\bar{a}, \bar{b}, \bar{c})$ directly.

Step 8: cost. The costs in Algorithm 1 are per assignment, and a product that occurs in several assignments is computed once and counted where it first occurs. Line 1 costs one product for μ and $h-1$ for the η_k. In line 2 the running products cost $h-2$, and γ costs the product $\mu\Pi_{\eta,h}$, a squaring and the product by C. Line 3 costs $h-1$. In line 4 the running products cost $N-3$ for the Γ_k and $h-1$ to continue the $\Pi_{\eta,k}$, and the e_i another $N-3$, as e_{N-2} needs no product. In line 5, Λ and the $N-3$ products p_i are computed once, and the sums s_i, u_i, w_i, Σ_1 and Σ_2 are free; Σ_{11} costs $h-1$ squarings, and Σ_{12} and Σ_3 cost one product per pair and one product by Λ, i.e. h and $h-1$. In line 6, $g = \mu e_1$ costs one product. From line 7 on, the following products are reused:

- g^2, computed for V, gives $g^3 = g^2 \cdot g$ in T_2 and is reused in $\mathcal{U}$, $\mathcal{R}$, $\mathcal{W}$ and $\mathcal{R}_d$, and g^3 is reused in $\mathcal{R}_d$ and $\mathcal{B}$;
- $\mu\Sigma_2$, computed for T_1, is reused in $\mathcal{S}$;
- D_2T_2, computed for $\mathcal{U}$, is reused in $\mathcal{R}$;
- $G_n\Delta_2$ and $F_n\mathcal{S}$, computed for $\mathcal{W}$, are reused in $\mathcal{B}$;
- $D_2 g$, computed for $\bar{a}$, is reused in $\bar{b}$, and F_n^2, $D_2F_n^2g^2$ and $\mathcal{BU}$, computed for $\bar{b}$, are reused in $\bar{c}$.

With these, the costs of lines 7–10 are as stated. If $C = 1$, the products by C in γ and $\mathcal{B}$ are free. If $D_2 = 1$, so are $\eta_2 = \mu D_2$, D_2T_2 in $\mathcal{U}$ and $\mathcal{R}$, the product by D_2 in $\mathcal{B}$, and $D_2 g$ in $\bar{a}$, $\bar{b}$ and $\bar{c}$; moreover $D_2F_n^2g^2 = F_n^2 \cdot g^2$ then reuses g^2, which saves one more product. □

To summarize, Algorithm 1 evaluates Theorem 2.2 over common denominators. The root is taken of the polynomial expression $-C(\mu\eta_2 \cdots \eta_h)^2$, all h_i share the denominator $\mathcal{D}$ so that the sums are polynomials in the e_i, and every Tate quantity becomes a numerator over powers of g and D_2. Proposition 3.5 records the translation back: u, s, r and the codomain cubic in terms of the outputs, $x(P') = x_1'$ in the coordinates of E, and the exact scaling factor ϑ, so that every model can use the output without inversions. Lines 1–5 dominate the cost; the remaining lines cost a constant 53 **M**, so the core accounts for about $6.5N$ **M** of the $10N$ to $15.5N$ **M** of each model.

Algorithm 1 needs Π_Δ^2, and some models also need Π_D. As products, both cost about $\frac{1}{2}N$. The recursions that produce the multiples, however, relate consecutive denominators, and then one of them is almost free.

Lemma 3.6 (Telescoping denominators)**.** *Let $h = \frac{1}{2}(N-1)$, let $(\Delta_k, D_k)_{2 \le k \le h}$ be as in the input of Algorithm 1, i.e. $x_k - x_1 = \Delta_k/(D_k Z_1)$, and put $D_1 = 1$, $\Pi_\Delta = \prod_{k=2}^h \Delta_k$ and $\Pi_D = \prod_{k=2}^h D_k$.*

1. *If $D_2 = 1$ and $D_{k+1} = \Delta_k^2 D_{k-1}$ for $2 \le k < h$, then $\Pi_\Delta^2 = \Delta_h^2 D_{h-1} D_h$, which costs 3 **M**.*
2. *If $D_2 = D_3 = 1$ and $D_{k+1} = \Delta_k^2 D_{k-1}$ for $3 \le k < h$, then $\Pi_\Delta^2 = \Delta_2^2 \Delta_h^2 D_{h-1} D_h$, which costs 5 **M**.*
3. *If $D_k = Z_k$ for projective multiples $x([k]P) = (X_k : Z_k)$, and $D_k = cX_{k-2}\Delta_{k-1}^2$ for $3 \le k \le h$ and a constant $c \in \mathbb{Z}$, then*

$$\Pi_D = c^{h-2} D_2 \, \Pi_{X,h-2} \, \Pi_{\Delta,h-1}^2, \qquad \Pi_{X,k} = \prod_{i=1}^k X_i, \quad \Pi_{\Delta,k} = \prod_{j=2}^k \Delta_j,$$

*which costs 3 **M** once $\Pi_{\Delta,h-1}$ and $\Pi_{X,h-2}$ are known.*

Proof. In the first case, put $D_{h+1} = \Delta_h^2 D_{h-1}$. Then $\Delta_k^2 = D_{k+1}/D_{k-1}$ for $2 \le k \le h$, and the product telescopes to $\Pi_\Delta^2 = D_h D_{h+1}/(D_1 D_2) = \Delta_h^2 D_{h-1} D_h$. The second case is the same for $3 \le k \le h$, with $D_2 D_3 = 1$, and the factor Δ_2^2 added. In the third case the product telescopes directly: $\prod_{k=3}^h D_k = c^{h-2} \prod_{k=3}^h X_{k-2} \prod_{k=3}^h \Delta_{k-1}^2$. □

The first case applies after the rescaling $D_2 = 1$ of the shifted recursion (Algorithms 3 and 11), the second to the recursion in Tate normal form (Algorithm 2), and the third to the x-only ladders (Algorithms 4 and 6). In the latter, $\Pi_{\Delta,h-1}$ and $\Pi_{X,h-2}$ are intermediate values of the running products for Π_Δ and for the product Π_X, which the normalization needs in any case. Short Weierstrass curves and Tate normal form do not need Π_D, so for them Algorithm 1 needs no product over the kernel besides its own running products.

Lemma 3.7 (Common root)**.** *Let $C^*(x) = x^3 + \alpha_2 x^2 + \alpha_1 x + \alpha_0$ be a cubic and $\kappa \in K$. Suppose that x_0 is a root of C^* with $C^{*\prime}(x_0) = \kappa$. Put $n_0 = 9\alpha_0 - \alpha_2\alpha_1 + \alpha_2\kappa$. If $\delta := 6\alpha_1 + 3\kappa - 2\alpha_2^2 \neq 0$, then*

$$x_0 = -\frac{n_0}{\delta}.$$

Proof. Division with remainder gives $C^*(x) = \left(\frac{x}{3} + \frac{\alpha_2}{9}\right)\left(C^{*\prime}(x) - \kappa\right) + \frac{1}{9}(\delta x + n_0)$. Evaluating at x_0 gives the claim. □

Remark 3.8. If $\delta = 0$, both roots of $C^{*\prime} - \kappa$ are roots of C^*. For the models of this paper this happens exactly when the codomain has j-invariant 1728 and a specific position of the distinguished 2-torsion point (Proposition 4.10); in that case a norm formula gives the normalization instead.

We use Lemma 3.7 to avoid square roots. For a model with a distinguished 2-torsion point (Montgomery, Edwards, Jacobi quartic), the image of that point is a root of the cubic, and the normalization of the target model fixes the value of the derivative there. That value is determined by a single product over the kernel.

4 Radical steps on each model

In this section we provide the radical steps on each model. All algorithms follow the same pattern, in four stages.

1. *Multiples.* From $x(P)$ in the given model, compute the input of Algorithm 1, i.e. essentially the x-coordinates of half of the kernel: F_n, G_n and the projective differences (Δ_k, D_k) of the first $\frac{1}{2}(N-1)$ multiples. Models with a fast x-only addition law (Montgomery curves and $y^2 = x(x^2 + Ax + B)$) obtain (Δ_k, D_k) as a by-product of their xADD; the others use the shifted recursion of Lemma 3.3. This costs between $3N$ and $4N$ **M**, and the structure of the recursion also gives Π_Δ^2 or Π_D almost for free (Lemma 3.6).

2. *Radical core.* Run Algorithm 1, which costs about $6.5N$ **M** and contains the only Nth root of the step.

3. *Normalization.* By Proposition 3.4, the output describes the Vélu codomain only up to the affine map $x = \Omega x^* + x_1'$. Each model fixes more than that, and this is where the models differ. Short Weierstrass curves only require $a_2 = 0$, which can be read off directly. Models with a distinguished point of order 2 (Montgomery, Edwards, $y^2 = x(x^2 + Ax + B)$, Jacobi quartics) locate its image as a common root (Lemma 3.7), using one product over the kernel. Huff curves, Kummer lines, Legendre curves and Jacobi intersections also place the other points of order 2, with norms over the kernel, and twisted Hessian curves need a product of a cubic rational function over the kernel, combined with the invariants c_4 and c_6 of the Vélu codomain. In each case the next generator P' is expressed in the coordinates of the target model.

4. *Pushing points.* Points are pushed through a model-specific x-only isogeny, at the cost of one or two products or sums over the kernel per point and, for some models, a shared precomputation.

The proofs of the propositions below follow the same order: the multiples are justified by the addition law of the model, the core by Proposition 3.5, and the normalization by the theorem that precedes each algorithm. In the last step of a chain no next generator is needed, hence no root and no radical core: only the multiples and a normalization that does not need P' are computed. This is the "codomain only" variant of each subsection. Throughout, the cost of each line in $\mathbf{M}$ is on the right.

4.1 Tate normal form

For completeness, Algorithm 2 is [17, Algorithm 5.2] with Lemma 3.1, in the notation of Algorithm 1. Here $P = (0, 0)$, so the multiples are the Tate coordinates themselves, in the model scaled by C, with $a_1 = C - c_n$, $a_2 = -b_n C$, $a_3 = -b_n C^2$ and $F(0) = a_3^2$. Points are pushed as in [17, Proposition 5.4]: with the kernel x-coordinates written as $\bar{X}_k/D$ over a common denominator, $\ell_k = X_R D - \bar{X}_k Z_R$, $\Pi = \prod_k \ell_k$ and $K_1/\Pi = \sum_k \ell_k^{-1}$, $K_2/\Pi^2 = \sum_k \ell_k^{-2}$ (including $\bar{X}_1 = 0$ for P itself), formula (2) reads

$$x(\varphi(R)) = \frac{\Pi^2\big(N X_R D - 2\big(\sum_k \bar{X}_k\big) Z_R\big) + D^2\big(D F_h K_2 - G_h K_1 \Pi\big)}{Z_R D\, \Pi^2},$$

with F_h, G_h the homogenizations of F, G at $(X_R : Z_R)$, and the change to the next scaled model is $x \mapsto 2g\mathcal{U}^2(2g^5 x - C^2\mathcal{R})$. We do not accumulate K_2 and Π^2 directly. Instead, $M = \Pi \sum_{i<j} (\ell_i \ell_j)^{-1}$ is the next elementary symmetric function of the ℓ_k, and the identity $\sum_k \ell_k^{-2} = \big(\sum_k \ell_k^{-1}\big)^2 - 2\sum_{i<j} (\ell_i \ell_j)^{-1}$ gives $K_2 = K_1^2 - 2\Pi M$. The updates of M, K_1 and Π cost $3\mathbf{M}$ per index, against $5\mathbf{M}$ for K_1, K_2, Π and Π^2.

Proposition 4.1. *Using Lemma 3.1, the inversion-free radical step of [17, Algorithm 5.2] costs* $10N$ $\mathbf{M}$ *for odd* $N \geq 11$ *(Algorithm 2), instead of* $\frac{1}{2}(25N - 13)$ $\mathbf{M}$ *plus the power* μ^{N-1}.

Proof. Algorithm 2 follows the four stages above. Lines 1–2 compute the Tate coordinates of the multiples by the recursion of [17, Lemma 5.1], i.e. Lemma 3.3 with $x_1 = 0$, in the model scaled by C; here $D_2 = D_3 = 1$. Line 3 is the radical core with $F_n = F(0) = a_3^2$, $G_n = \frac{1}{2}G(0) = \frac{1}{2}a_1 a_3$ and $C = 1$; since $D_2 = D_3 = 1$ and $D_k = X_{k-1}^2 D_{k-2}$, Lemma 3.6(2) gives $\Pi_\Delta^2 = X_2^2 X_h^2 D_{h-1} D_h$. No normalization is needed, since line 4 writes b' and c' of Theorem 2.2 over the common denominator $C' = 2C\mathcal{U}^2$, using u, s and r from Proposition 3.5, and lines 5–6 push points with (2). For the count, in [17, Proposition 5.3] the sum $\Sigma_{11} = \sum e_i^2$ accounts for $N - 2$ squarings. By Lemma 3.1 it costs $\frac{1}{2}(N - 3)$ squarings and one multiplication, and the same lemma halves the cost of Σ_3 and Σ_{12}. The root needs neither Π_D nor the power μ^{N-1} (line 2 of Algorithm 1), and Lemma 3.6(2) replaces the product Π_Δ. The remaining savings come from reusing values, namely $g = \mu e_1$ in line 6 and Γ_2 from line 3 in line 8 of Algorithm 1, and $\mu\Sigma_2$ and $D_2 T_2$ in $\mathcal{S}$ and $\mathcal{R}$. Further savings come from the factors $D_2 = D_3 = 1$ in the recursion and the products over the kernel, the running products for D and the $\bar{X}_k$, and $C\mathcal{S}$, $C\mathcal{U}$ in the output; see Algorithms 1 and 2. □

4.2 Short Weierstrass curves

Short Weierstrass curves are the simplest case of the reduction. We will see that the condition $a_2 = 0$ fixes the affine map of Proposition 3.4 without any product over the kernel.

Theorem 4.2. *Let* $E\colon y^2 = x^3 + a_4 x + a_6$ *and let* $P \in E(K)$ *have order* N. *Compute* $\hat{x}_k$ *for* $2 \leq k \leq (N-1)/2$ *by Lemma 3.3, starting from* $\hat{x}_2 = x([2]P) - x_1$, *and* (b, c) *by Lemma 3.2.*

Algorithm 2 Radical step in Tate normal form

Input: $(b_n : c_n : C)$ with $b = b_n/C$, $c = c_n/C$, and $P = (0,0) \in E_{b,c}$ of order N; points $x(R_j) = (X_{R_j} : Z_{R_j})$

Output: $(b'_n : c'_n : C')$ of the next Tate normal form, and $x(\varphi(R_j))$ in its coordinates

1: $a_1 \leftarrow C - c_n$, $(X_2, X_3) \leftarrow (b_n C, c_n C)$, $a_3 \leftarrow -X_2 C$, $D_2 = D_3 = 1$ — 3

2: $t_1 \leftarrow a_3 D_{k-1}$, $X_k \leftarrow t_1(t_1 + a_1 X_{k-1})D_{k-2} - X_{k-2}X_{k-1}^2$, $D_k \leftarrow X_{k-1}^2 D_{k-2}$ for $4 \le k \le (N-1)/2$ — $\frac{1}{2}(7N - 59)$

3: Lines 1–8 of Algorithm 1 on $(a_3^2, \frac{1}{2}a_1a_3, 1, X_2^2 X_h^2 D_{h-1} D_h, (X_k, D_k))$ (Lemma 3.6(2)); they do not use $G_n = \frac{1}{2}a_1a_3$ — $\frac{1}{2}(13N + 25)$

4: $b'_n \leftarrow g(2b_n g^5 + 2a_1 \mathcal{S} g^3 + 2C\mathcal{S}^2 g - 3C\mathcal{R})$, $c'_n \leftarrow 2\mathcal{U}(C\mathcal{U} - g(a_1 g^2 + 2C\mathcal{S}))$, $C' \leftarrow 2C\mathcal{U}^2$ — 14

5: **Pushing, once:** $D \leftarrow \prod_{k=2}^{(N-1)/2} D_k$, $\bar{X}_1 \leftarrow 0$, $\bar{X}_k \leftarrow X_k \prod_{j \ne k} D_j$ for $2 \le k \le (N-1)/2$, so that $x_k = \bar{X}_k/D$ (from the running products of the D_j from both ends); $\sum_k \bar{X}_k$, D^2, $b_2 \leftarrow a_1^2 - 4b_n C$, $b_4 \leftarrow a_1 a_3$, $b_6 \leftarrow a_3^2$, $v_1 \leftarrow 2g\mathcal{U}^2$, $v_2 \leftarrow 2g^5$, $v_3 \leftarrow C^2\mathcal{R}$ — $2N - 7$

6: **Pushing, for each** $R_j = (X_R : Z_R)$**:** $\ell_k \leftarrow X_R D - \bar{X}_k Z_R$ for $1 \le k \le (N-1)/2$; $\Pi \leftarrow \prod_k \ell_k$, $K_1 \leftarrow \Pi \sum_k \ell_k^{-1}$, $K_2 \leftarrow \Pi^2 \sum_k \ell_k^{-2}$, computed as $M, K_1, \Pi \leftarrow M\ell_k + K_1, K_1\ell_k + \Pi, \Pi\ell_k$ from $(0,0,1)$, followed by $K_2 \leftarrow K_1^2 - 2\Pi M$; $F_h \leftarrow 4X_R^3 + b_2 X_R^2 Z_R + 2b_4 X_R Z_R^2 + b_6 Z_R^3$, $G_h \leftarrow 6X_R^2 + b_2 X_R Z_R + b_4 Z_R^2$; $t_1 \leftarrow \Pi^2\big(NX_R D - 2(\sum_k \bar{X}_k)Z_R\big) + D^2(DF_h K_2 - G_h K_1 \Pi)$, $t_2 \leftarrow Z_R D \Pi^2$; $x(\varphi(R_j)) \leftarrow \big(v_1(v_2 t_1 - v_3 t_2) : t_2\big)$ — $2N + 16$ each

Let r, u and (b', c') be the output of Theorem 2.2, and put $x_1' = x_1 + u_P^2 r$ and $\Omega = u_P^2 u^2$. Then $b_2'\Omega = 12x_1'$, so $\Omega = 12x_1'/b_2'$ if $b_2' \neq 0$, and the Vélu codomain of E with kernel $\langle P \rangle$ is $y^2 = x^3 + a_4'x + a_6'$ with

$$a_4' = 3{x_1'}^2 - \tfrac{1}{2}b_2'\Omega x_1' + \tfrac{1}{2}b_4'\Omega^2,$$
$$a_6' = -{x_1'}^3 + \tfrac{1}{4}b_2'\Omega {x_1'}^2 - \tfrac{1}{2}b_4'\Omega^2 x_1' + \tfrac{1}{4}b_6'\Omega^3,$$

and $x(P') = x_1'$ for a P-distinguished point P' of order N.

Proof. This is Proposition 3.4 with $a_2 = 0$, i.e. $b_2'\Omega = 12x_1'$, expanded in powers of x. □

If $x_1' \neq 0$, then $\Omega = 3x_1'/\alpha_2$ for the cubic $x^3 + \alpha_2 x^2 + \alpha_1 x + \alpha_0$ of $E_{b',c'}$, and the codomain only depends on x_1' and on ratios of the α_i:

$$a_4' = {x_1'}^2\Big(\frac{9\alpha_1}{\alpha_2^2} - 3\Big), \qquad a_6' = {x_1'}^3\Big(2 - \frac{9\alpha_1}{\alpha_2^2} + \frac{27\alpha_0}{\alpha_2^3}\Big).$$

Therefore no product over the kernel is needed, and by Proposition 3.5 the scaled $\bar{a}, \bar{b}, \bar{c}$ can be used as they are. If $x_1' = 0$, which forces $b_2' = 0$ and happens with probability about $1/p$, the ratio α_1/α_2^2 is undefined; then $a_4' = \alpha_1\Omega^2$ and $a_6' = \alpha_0\Omega^3$, with Ω from Proposition 3.5. Rescaling x by t_0^2 with $t_0 = Z_V\bar{a}$ makes the output polynomial. Points are pushed with (2), homogenized as in Algorithm 2: the kernel x-coordinates are written as $\bar{X}_i/D$ over a common denominator, and $F_h = Z_R^3 F(x)$, $G_h = Z_R^2 G(x)$ at $x = X_R/Z_R$; see lines 7–8 of Algorithm 3.

Algorithm 3 Radical step on short Weierstrass curves

Input: $a_4, a_6 \in K$, $(X_1 : Z_1) = x(P)$ with P of order N on $y^2 = x^3 + a_4x + a_6$; points $x(R_j) = (X_{R_j} : Z_{R_j})$

Output: a_4', a_6' with $y^2 = x^3 + a_4'x + a_6' \cong E/\langle P \rangle$, $x(P')$, and $x(\varphi(R_j))$

1: $F_n \leftarrow 4(X_1(X_1^2 + a_4Z_1^2) + a_6Z_1^3)$, $G_n \leftarrow 3X_1^2 + a_4Z_1^2$ — 6
2: $\Delta_2 \leftarrow (X_1^2 - a_4Z_1^2)^2 - X_1(F_n + 8a_6Z_1^3)$, $(\Delta_1, D_1) \leftarrow (0, 1)$ — 2
$(X_1, Z_1, G_n, F_n) \leftarrow (F_nX_1, F_nZ_1, F_n^2G_n, F_n^4)$, $D_2 \leftarrow 1$ — 5
3: $\Delta_{k+1} \leftarrow D_{k-1}D_k(F_nD_k + 2G_n\Delta_k) - \Delta_k^2\Delta_{k-1}$, $D_{k+1} \leftarrow \Delta_k^2 D_{k-1}$ for $2 \leq k < (N-1)/2$ — $\frac{1}{2}(7N - 49)$
4: Algorithm 1 on $(F_n, G_n, 1, \Delta_h^2 D_{h-1}D_h, (\Delta_k, D_k))$ (Lemma 3.6(1)) — $\frac{1}{2}(13N + 69)$
5: $X_V \leftarrow X_1\mathcal{R}_d + F_n\mathcal{R}$, $Z_V \leftarrow Z_1\mathcal{R}_d$, $t_1 \leftarrow X_VZ_V$, $t_0 \leftarrow Z_V\bar{a}$ — 4
6: **if** $\bar{a} = 0$: $t_0 \leftarrow 2F_nZ_1\Delta_2^2 g^6$, $a_4' \leftarrow \bar{b}t_0^2$, $a_6' \leftarrow \bar{c}t_0^3$, $x(P') \leftarrow 0$ — 9
else: $t_2 \leftarrow t_1\bar{a}$, $a_4' \leftarrow t_2^2(9\bar{b} - 3\bar{a}^2)$, $a_6' \leftarrow t_2^3\big(\bar{a}(2\bar{a}^2 - 9\bar{b}) + 27\bar{c}\big)$, $x(P') \leftarrow t_2\bar{a}$ — 8
7: **Pushing, once:** $D \leftarrow Z_1\prod_{k=2}^{(N-1)/2} D_k$, $\bar{X}_1 \leftarrow X_1\prod_{k=2}^{(N-1)/2} D_k$, $\bar{X}_i \leftarrow (X_1D_i + \Delta_i)\prod_{j\neq i} D_j$ for $2 \leq i \leq (N-1)/2$, so that $x_i = \bar{X}_i/D$ (from the running products of the D_j from both ends); $\sum_i \bar{X}_i$, D^2, t_0^2 — $\frac{1}{2}(5N - 25)$
8: **Pushing, for each** $R_j = (X_R : Z_R)$**:** $\ell_i \leftarrow X_RD - \bar{X}_iZ_R$ for $1 \leq i \leq (N-1)/2$, and Π, K_1, K_2 as in line 6 of Algorithm 2; $F_h \leftarrow 4(X_R^3 + a_4X_RZ_R^2 + a_6Z_R^3)$, $G_h \leftarrow 6X_R^2 + 2a_4Z_R^2$; $x(\varphi(R_j)) \leftarrow \big(t_0^2\big(\Pi^2(NX_RD - 2(\sum_i \bar{X}_i)Z_R) + D^2(DF_hK_2 - G_hK_1\Pi)\big) : Z_RD\Pi^2\big)$ — $2N + 13$ each

Proposition 4.3. *Algorithm 3 is correct and costs $10N + 35$ **M** for $N \geq 9$, plus $\frac{1}{2}(5N - 25)$ **M** once and $2N + 13$ **M** per point for pushing. In the case $\bar{a} = 0$ of line 6, the step costs one **M** more.*

Proof. Lines 1–3 compute the input of Algorithm 1: $F_n = Z_1^3 F(x_1)$ and $G_n = \frac{1}{2} Z_1^2 G(x_1)$ in line 1, and the multiples by Lemmas 3.2 and 3.3 in the representation $x_k - x_1 = \Delta_k/(D_k Z_1)$, starting from the duplication formula: $Z_1^4 \psi_3(x_1) = 3X_1^4 + 6a_4 X_1^2 Z_1^2 + 12 a_6 X_1 Z_1^3 - a_4^2 Z_1^4 = -\Delta_2$, which reuses X_1^2, $a_4 Z_1^2$ and $a_6 Z_1^3$ from line 1. The second half of line 2 replaces Z_1 by $F_n Z_1$ and X_1 by $F_n X_1$, which leaves x_1 unchanged; in this representation F_n and G_n become F_n^4 and $F_n^2 G_n$, and $\hat{x}_2 = \Delta_2/(F_n Z_1)$ has $D_2 = 1$. Hence Lemma 3.6(1) applies, and all products by D_1 and D_2, in the recursion and in Algorithm 1, are free. Line 4 is the radical core. Lines 5–6 normalize the codomain, which here only means writing it in short Weierstrass form. By Proposition 3.5 with $C = 1$, line 5 computes $X_V/Z_V = x_1'$, the x-coordinate of P' on the Vélu codomain. The ratios $\alpha_1/\alpha_2^2 = \bar{b}/\bar{a}^2$ and $\alpha_0/\alpha_2^3 = \bar{c}/\bar{a}^3$ do not depend on the scaling ϑ of Proposition 3.5, so the formulae above give $t_0^4 a_4' = t_2^2(9\bar{b} - 3\bar{a}^2)$, $t_0^6 a_6' = t_2^3(2\bar{a}^3 - 9\bar{a}\bar{b} + 27\bar{c})$ and $t_0^2 x_1' = t_2 \bar{a}$ for $t_0 = Z_V \bar{a}$ and $t_2 = X_V Z_V \bar{a}$; this is the isomorphism $(x, y) \mapsto (t_0^2 x, t_0^3 y)$ of the Vélu codomain. Since $\bar{a} = \vartheta \alpha_2$ and $\alpha_2 \Omega = 3x_1'$, we have $\bar{a} = 0$ if and only if $x_1' = 0$, and then this isomorphism degenerates. In that case the first branch of line 6 uses $\vartheta/\Omega = 2F_n Z_1 \Delta_2^2 g^6$ from Proposition 3.5 (with $C = D_2 = 1$): Theorem 4.2 with $x_1' = 0$ gives $a_4' = \alpha_1 \Omega^2 = \bar{b}(\Omega/\vartheta)^2$ and $a_6' = \alpha_0 \Omega^3 = \bar{c}(\Omega/\vartheta)^3$, and the isomorphism $(x, y) \mapsto (t_0^2 x, t_0^3 y)$ with $t_0 = \vartheta/\Omega$ gives line 6. Lines 7–8 push points by (2), followed by the same isomorphism, with the kernel x-coordinates over the common denominator D of line 7: with $x = X_R/Z_R$ we have $\ell_i = D Z_R (x - x_i)$, so $\sum_i (x - x_i)^{-1} = D Z_R K_1/\Pi$ and $\sum_i (x - x_i)^{-2} = D^2 Z_R^2 K_2/\Pi^2$, while $F_h = Z_R^3 F(x)$ and $G_h = Z_R^2 G(x)$. The counts follow line by line; in line 8 the first two indices cost $2\mathbf{M}$ each, every further index $4\mathbf{M}$, and Π^2 and K_2 cost $3\mathbf{M}$ at the end. □

Codomain only. Without a next generator, Vélu's formulae $a_4' = a_4 - 5\sum_i G(x_i)$ and $a_6' = a_6 - 7\sum_i \big(F(x_i) + x_i G(x_i)\big)$ [27, 40] only need the power sums $S_j = \sum_i \bar{X}_i^j$, $j = 1, 2, 3$, of the kernel numerators of line 7. With $h = \frac{1}{2}(N - 1)$, the isomorphic curve

$$a_4' D^4 = D^2\big((1 - 10h) a_4 D^2 - 30 S_2\big), \qquad a_6' D^6 = D^3\big((1 - 28h) a_6 D^3 - 70 S_3 - 42 S_1 \cdot a_4 D^2\big)$$

is obtained from lines 1–3 without the rescaling in line 2, the common denominator of line 7 and $2h + 7$ further multiplications, in total $7N - 17$ $\mathbf{M}$ and without a root.

This shortcut is specific to short Weierstrass curves, whose normalization $a_2 = 0$ is exactly the one of Vélu's formulae. It does not improve the other models. Their codomain-only variants need the codomain in the target model, and Vélu's formulae only give it up to the affine map of Proposition 3.4. Moving it to the target model requires the image of the distinguished torsion point, e.g. by evaluating (2) there, which is another sum over the kernel. The norm and trace formulae of the following subsections are cheaper. Nor does it help the radical step itself, since the power sums do not reveal the next generator P', which needs the radical core in any case.

4.3 Montgomery curves

Let $E_A\colon y^2 = x^3 + Ax^2 + x$, so that $F(x) = 4(x^3 + Ax^2 + x)$ and $G(x) = 2(3x^2 + 2Ax + 1)$. The coefficient B of a Montgomery curve $By^2 = x^3 + Ax^2 + x$ plays no role in x-only arithmetic, so all statements below also hold for $B \neq 1$ and for points on the quadratic twist. Let P have order N, write $x_i = x([i]P)$ and

$$\pi = \prod_{i=1}^{(N-1)/2} x_i, \qquad \sigma = \sum_{i=1}^{(N-1)/2} \Big(x_i - \frac{1}{x_i}\Big).$$

Lemma 4.4 (Costello–Hisil normalization)**.** *The rational map $\phi(x) = x \prod_{i=1}^{(N-1)/2} \big(\frac{x x_i - 1}{x - x_i}\big)^2$ is the x-coordinate map of an isogeny $E_A \to E_{A'}$ with kernel $\langle P \rangle$, where $A' = \pi^2(A - 6\sigma)$ [14, 36]. If x_V denotes the x-coordinate map of the Vélu isogeny with the same kernel, then $\phi = \pi^2(x_V + 2\sigma)$.*

Proof. The formula for A' is [14, 36] in our normalization, where the products run over one point of each pair $\{Q, -Q\}$. For the rest, let $R \in E_A$. For $Q \neq \pm R$, the addition law gives $x(R+Q)\,x(R-Q) = \big(x(R)x(Q)-1\big)^2/\big(x(R)-x(Q)\big)^2$, so grouping S and $-S$ gives $\phi(x(R)) = \prod_{S\in\langle P\rangle} x(R+S)$. As a function of R, this product is invariant under translation by $\langle P\rangle$, and its only poles are double poles at the points of $\langle P\rangle$, since no factor vanishes there ($(0,0) \notin \langle P\rangle$ as N is odd). By Vélu's formula, x_V has the same invariance and the same poles. Hence ϕ is a function on $E_A/\langle P\rangle$ with only a double pole at $\mathcal{O}$, i.e. an affine function of x_V; in particular ϕ is the x-coordinate map of an isogeny with kernel $\langle P\rangle$. At infinity, $x_V = x + O(1/x)$ by Kohel's formula, while $\phi = \pi^2 x \prod_i (1 - \frac{1}{x x_i})^2 (1 - \frac{x_i}{x})^{-2} = \pi^2 x + 2\pi^2\sigma + O(1/x)$. Hence $\phi = \pi^2 x_V + 2\pi^2\sigma$. □

Theorem 4.5. *Let (b, c) be the Tate normal form of (E_A, P) and $r, u, (b', c')$ the output of Theorem 2.2. Put $\Omega = u_P^2 u^2$ and $\omega = \pi^2\Omega$, let $C^*(x) = x^3 + \frac{b'_2}{4}x^2 + \frac{b'_4}{2}x + \frac{b'_6}{4}$ be the cubic of $E_{b',c'}$, and let x_0 be given by Lemma 3.7 with $\kappa = \omega^{-2}$. Then*

$$A' = \omega\Big(\frac{b'_2}{4} + 3x_0\Big), \qquad x(P') = -\omega x_0$$

are the Montgomery coefficient of the codomain $E_{A'}$ of Lemma 4.4 and the x-coordinate of a P-distinguished point $P' \in E_{A'}$ of order N.

Proof. By Proposition 3.4 and Lemma 4.4, the x-coordinate of a point of $E_{b',c'}$ maps to $X = \pi^2(\Omega x^* + x'_1 + 2\sigma) = \omega(x^* - x_0)$ on $E_{A'}$, with $x_0 = -(x'_1 + 2\sigma)/\Omega$. Substituting $X = \omega(x^* - x_0)$ into $X^3 + A'X^2 + X$ gives ω^3 times $(x^* - x_0)^3 + \frac{A'}{\omega}(x^* - x_0)^2 + \omega^{-2}(x^* - x_0)$, which must equal $\omega^3 C^*(x^*)$. Comparing coefficients, $C^*(x_0) = 0$, $C^{*\prime}(x_0) = \omega^{-2}$ and $\frac{b'_2}{4} = -3x_0 + \frac{A'}{\omega}$. Finally, $(0,0) \in E_{b',c'}$ maps to $X = -\omega x_0$. □

Only the product π is needed, not σ, since the image $-x_0$ of the 2-torsion point is recovered from the two polynomial conditions. Compared to computing A' with the twisted Edwards formula of [29] and σ separately, this saves two products over the kernel. Points are pushed with the map ϕ of Lemma 4.4, which needs no precomputation beyond the multiples.

Proposition 4.6. *Algorithm 4 is correct and costs $\frac{1}{2}(21N + 105)$ **M**, excluding the Nth root, plus $2N$ **M** per pushed point.*

Proof. Lines 1–3 compute the input of Algorithm 1. Line 1 gives $F_n = CZ_1^3 F(x_1)$ and $G_n = \frac{1}{2}CZ_1^2 G(x_1)$, line 2 is the doubling, and line 3 runs the Montgomery ladder, in which the xADD for $[k]P$ computes $t_1 - t_2 = 2\Delta_{k-1}$ with $\Delta_k = X_k Z_1 - X_1 Z_k$, so the differences come for free, with $D_k = Z_k$. Moreover $Z_k = 4X_{k-2}\Delta_{k-1}^2$, so Lemma 3.6(3) gives Π_D in line 4 from the running products of Π_Δ and of Π_X, which is therefore started before the core. Line 4 is the radical core. Lines 5–8 normalize the codomain to the Costello–Hisil form of Lemma 4.4, by locating the image of $(0,0)$ with Theorem 4.5. For these lines, Proposition 3.5 gives $\Omega = F_n\mathcal{U}^2/(CZ_1\Delta_2^2 g^6)$ and $\vartheta = 2Z_2 F_n^2 \mathcal{U}^2$, and $\pi = \Pi_X/\Pi_Z$ with $\Pi_Z = Z_1\Pi_D$. Since $Z_2 = CZ_1F_n$ by line 2, this gives

$$\frac{\omega}{\vartheta} = \frac{\pi^2\Omega}{\vartheta} = \frac{\Pi_X^2}{2Z_2F_n g^6 \Pi_Z^2 CZ_1\Delta_2^2} = \frac{\Pi_X^2}{2\Psi}, \qquad \Psi = (Z_2 g^3 Z_1 \Pi_D \Delta_2)^2.$$

The scaled cubic $\bar{C}(y) = y^3 + \bar{a}y^2 + \bar{b}y + \bar{c} = \vartheta^3 C^*(y/\vartheta)$ has the root $\bar{y}_0 = \vartheta x_0$ with $\bar{C}'(\bar{y}_0) = \vartheta^2\omega^{-2} = 4\Psi^2/\Pi_X^4$. Lemma 3.7 for $\bar{C}$, with numerator and denominator multiplied by Π_X^4, gives $\bar{y}_0 = -n_0/\delta$, where δ vanishes if and only if the δ of Theorem 4.5 does. Theorem 4.5 then reads $A' = \frac{\omega}{\vartheta}(\bar{a} + 3\bar{y}_0) = \Pi_X^2(\bar{a}\delta - 3n_0)/(2\Psi\delta)$ and $x(P') = -\frac{\omega}{\vartheta}\bar{y}_0 = \Pi_X^2 n_0/(2\Psi\delta)$, which is line 8. Line 9 pushes points through the map ϕ of Lemma 4.4 [14], since $s_i + t_i = 2(X_R X_i - Z_R Z_i)$ and $s_i - t_i = 2(X_R Z_i - Z_R X_i)$; it costs $4\mathbf{M}$ per index, minus $2\mathbf{M}$ for the first factors of the products, and $4\mathbf{M}$ at the end. Line 7 is Proposition 4.10, with A' from the twisted Edwards formula of the codomain-only paragraph below, i.e. $t_1/t_2 = a'/d'$ and $A' = 2(t_1 + t_2)/(t_1 - t_2)$; there $x(P') = \big(\Pi_X^2\bar{a}/(2\Psi) - A'\big)/3$, which is $\big(\Pi_X^2\bar{a}C' - 2\Psi A'_n : 6\Psi C'\big)$ for $A' = A'_n/C'$. □

Algorithm 4 Radical step on Montgomery curves

Input: $(A_n : C)$ with $A = A_n/C$, $(X_1 : Z_1) = x(P)$ with P of order N; points $x(R_j) = (X_{R_j} : Z_{R_j})$

Output: $(A'_n : C')$, $x(P')$, and $x(\phi(R_j))$ on $E_{A'}$ (Theorem 4.5)

1: $F_n \leftarrow 4X_1(CX_1^2 + A_nX_1Z_1 + CZ_1^2)$, $G_n \leftarrow 3CX_1^2 + 2A_nX_1Z_1 + CZ_1^2$ — 7

2: $(X_2 : Z_2) \leftarrow \left(C^2(X_1^2 - Z_1^2)^2 : CZ_1F_n\right)$ — 3

3: $t_1, t_2 \leftarrow (X_{k-1} - Z_{k-1})(X_1 + Z_1)$, $(X_{k-1} + Z_{k-1})(X_1 - Z_1)$, $\Delta_{k-1} \leftarrow \frac{t_1 - t_2}{2}$; if $k \le (N-1)/2$: $X_k, Z_k \leftarrow Z_{k-2}(t_1 + t_2)^2$, $X_{k-2}(t_1 - t_2)^2$, for $3 \le k \le (N+1)/2$ — $3N - 13$

4: $\Pi_{\Delta,k} \leftarrow \prod_{j=2}^{k} \Delta_j$ for $2 \le k \le h$, $\Pi_{X,h-2} \leftarrow \prod_{i=1}^{h-2} X_i$, $\Pi_D \leftarrow 4^{h-2}Z_2\Pi_{X,h-2}\Pi^2_{\Delta,h-1}$ (Lemma 3.6(3)); Algorithm 1 on $(F_n, G_n, C, \Pi^2_{\Delta,h}, (\Delta_k, Z_k))$ — $\frac{1}{2}(15N + 73)$

5: $\Pi_X \leftarrow \Pi_{X,h-2}X_{h-1}X_h$, $\Psi \leftarrow (Z_2g^3Z_1\Pi_D\Delta_2)^2$ — 7

6: $n_0 \leftarrow \Pi_X^4(9\bar{c} - \bar{a}\bar{b}) + 4\bar{a}\Psi^2$, $\delta \leftarrow \Pi_X^4(6\bar{b} - 2\bar{a}^2) + 12\Psi^2$ — 8

7: **if** $\delta = 0$: $t_1 \leftarrow (A_n + 2C)^N \prod_i (X_i + Z_i)^8$, $t_2 \leftarrow (A_n - 2C)^N \prod_i (X_i - Z_i)^8$, $(A'_n : C') \leftarrow (2(t_1 + t_2) : t_1 - t_2)$ (twisted Edwards formula), $x(P') \leftarrow \left(\Pi_X^2\bar{a}C' - 2\Psi A'_n : 6\Psi C'\right)$ (Proposition 4.10) — $N + 2\mathbf{E}_N + 9$

8: **else**: $A'_n \leftarrow \Pi_X^2(\bar{a}\delta - 3n_0)$, $C' \leftarrow 2\Psi\delta$, $x(P') \leftarrow (\Pi_X^2 n_0 : C')$ — 4

9: **Pushing, for each** $R_j = (X_R : Z_R)$ (no precomputation): $s_i, t_i \leftarrow (X_R - Z_R)(X_i + Z_i)$, $(X_R + Z_R)(X_i - Z_i)$ for $1 \le i \le (N-1)/2$; $x(\phi(R_j)) \leftarrow \left(X_R(\prod_i(s_i + t_i))^2 : Z_R(\prod_i(s_i - t_i))^2\right)$ — $2N$ each

Codomain only. In the last step of a chain no next generator is needed. Then A' is computed from the multiples and the twisted Edwards formula $a' = a^N \prod_i (X_i + Z_i)^8$, $d' = d^N \prod_i (X_i - Z_i)^8$ with $a = A + 2$, $d = A - 2$ [29], and $A' = 2(a' + d')/(a' - d')$. With lines 1–3 restricted to X_i, Z_i (so F_n is not needed and $Z_2 = 4CX_1Z_1(CX_1^2 + A_nX_1Z_1 + CZ_1^2)$) this costs $4N + 2\mathbf{E}_N - 1$ **M** and no root.

4.4 Twisted Edwards curves

Twisted Edwards curves are birationally equivalent to Montgomery curves, and a Möbius transformation relates their y-coordinate to the Montgomery x-coordinate. We will therefore reduce to Algorithm 4, at no extra cost.

Algorithm 5 Radical step on twisted Edwards curves

Input: $(a : d)$ for $E_{a,d}$: $ax^2 + y^2 = 1 + dx^2y^2$, $(Y_1 : Z_1) = y(P)$ with P of order N; points $y(R_j) = (Y_{R_j} : Z_{R_j})$

Output: $(a' : d')$, $y(P')$, and the images $y(\varphi(R_j))$

1: $(A_n : C) \leftarrow (2(a + d) : a - d)$, $(X : Z) \leftarrow (Z + Y : Z - Y)$ for P and all R_j — 0

2: Algorithm 4 — $\frac{1}{2}(21N + 105)$; $2N$ each

3: $(a' : d') \leftarrow (A'_n + 2C' : A'_n - 2C')$, $(Y : Z) \leftarrow (X - Z : X + Z)$ for P' and all images — 0

Corollary 4.7. *Algorithm 5 is correct and has the cost of Algorithm 4:* $\frac{1}{2}(21N + 105)$ **M** *per step and* $2N$ **M** *per pushed point.*

Proof. The birational map $(x,y) \mapsto \left(\frac{1+y}{1-y}, \frac{1+y}{(1-y)x}\right)$ sends $E_{a,d}$ to $By^2 = x^3 + Ax^2 + x$ with $A = 2(a+d)/(a-d)$ and $B = 4/(a-d)$ [1]. Since x-only arithmetic does not depend on B, Algorithm 4 applies to the x-line. Line 1 converts the Edwards data to Montgomery data, $A = 2(a+d)/(a-d)$ and $x = (1+y)/(1-y)$; line 2 is the Montgomery step; and line 3 converts back, using $a' = A' + 2$, $d' = A' - 2$ (projectively) and $y = (x-1)/(x+1)$. All conversions are additions. □

4.5 Curves $y^2 = x(x^2 + Ax + B)$

Montgomery curves are the special case $B = 1$ of the curves

$$E_{A,B}\colon y^2 = x(x^2 + Ax + B), \qquad B(A^2 - 4B) \neq 0,$$

which are exactly the Weierstrass models with a rational 2-torsion point $T = (0,0)$ at the origin; every curve with a rational point of order 2 has such a model. If B is a square, $x \mapsto x/\sqrt{B}$ leads back to a Montgomery x-line, but in general it is not, for instance for the Jacobi quartics of Section 4.8. Another example is the family of curves $y^2 = x^3 + Ax^2 - x$ with $B = -1$ on which CSURF [7] works, since -1 is not a square modulo the CSURF primes $p \equiv 7 \pmod 8$. As $B' = B^N$ and N is odd, a radical step keeps $B = -1$. Here $F(x) = 4x(x^2 + Ax + B)$ and $G(x) = 2(3x^2 + 2Ax + B)$.

Lemma 4.8. *Let $P \in E_{A,B}(K)$ have odd order N, put $x_i = x([i]P)$ and $\pi = \prod_{i=1}^{(N-1)/2} x_i$. Then*

$$\phi_B(x) = x \prod_{i=1}^{(N-1)/2} \left(\frac{xx_i - B}{x - x_i}\right)^2$$

is the x-coordinate map of an isogeny $E_{A,B} \to E_{A',B'}$ with kernel $\langle P\rangle$, where $B' = B^N$, and $\phi_B - \pi^2 x_V$ is constant for the Vélu map x_V.

Proof. On $E_{A,B}$ the addition law gives $x(R+Q)\,x(R-Q) = \big(x(R)x(Q) - B\big)^2 / \big(x(R) - x(Q)\big)^2$ for $Q \neq \pm R$, so $\phi_B(x(R)) = \prod_{S \in \langle P\rangle} x(R+S)$, and the proof of Lemma 4.4 shows that ϕ_B is an affine function of x_V. Since $\phi_B = \pi^2 x + O(1)$ and $x_V = x + O(1)$ at infinity, $\phi_B = \pi^2 x_V + r_B$ for a constant r_B, and ϕ_B is the x-map of the Vélu isogeny followed by $(x,y) \mapsto (\pi^2 x + r_B, \pi^3 y)$. Since $\phi_B(0) = 0$, the codomain has the form $y^2 = x(x^2 + A'x + B')$. Let $e, \bar e$ be the roots of $x^2 + Ax + B$, so $B = e\bar e$, and put $\pi_e = \prod_{i=1}^{(N-1)/2}(x_i - e)$. Then $ex_i - B = e(x_i - \bar e)$, so

$$\phi_B(e) = e^N \left(\frac{\pi_{\bar e}}{\pi_e}\right)^2, \qquad \phi_B(\bar e) = \bar e^{\,N} \left(\frac{\pi_e}{\pi_{\bar e}}\right)^2, \tag{4}$$

and $B' = \phi_B(e)\phi_B(\bar e) = B^N$. □

Theorem 4.9. *Let $P \in E_{A,B}(K)$ have odd order $N \geq 5$, and let Ω, C^* be as in Theorem 4.5. Put $\omega = \pi^2\Omega$ and let x_0 be given by Lemma 3.7 with $\kappa = B^N/\omega^2$. Then the codomain of ϕ_B is E_{A',B^N} with*

$$A' = \omega\Big(\frac{b_2'}{4} + 3x_0\Big), \qquad x(P') = -\omega x_0,$$

for a P-distinguished point P' of order N.

Proof. Identical to the proof of Theorem 4.5, using Lemma 4.8 instead of Lemma 4.4: substituting $X = \omega(x^* - x_0)$ into $X^3 + A'X^2 + B^N X$ and comparing with $\omega^3 C^*(x^*)$ gives $C^*(x_0) = 0$ and $C^{*\prime}(x_0) = B^N/\omega^2$. □

Proposition 4.10 (The exceptional case $\delta = 0$)**.** *In the setting of Theorem 4.9, which includes Theorem 4.5 for $B = 1$, let x_0, e_1, e_2 be the roots of C^*, where x_0 corresponds to the image of $(0,0)$. The following are equivalent:*

1. *$\delta = 0$ in Lemma 3.7;*

2. *one of e_1, e_2 is the midpoint of the other two roots, i.e. $x_0 + e_k = 2e_j$ for $\{j,k\} = \{1,2\}$;*

3. *the codomain $E_{A',B'}$ satisfies $2A'^2 = 9B'$.*

In that case $E_{A',B'}\colon y^2 = x(x-r)(x-2r)$ with $r = -A'/3$, so $j(E_{A',B'}) = 1728$, and $(0,0)$ is one of the two points of order 2 that are not fixed by the automorphism of order 4. Moreover $2B = 2B'/B^{N-1}$ is then a square in K. In particular, $\delta \neq 0$ whenever $2B$ is not a square. For Montgomery, twisted Edwards, Huff curves and Kummer lines ($B = 1$), this means whenever $2 \notin K^{*2}$, e.g. over $\mathbb{F}_p$ with $p \equiv \pm 3 \pmod 8$, which includes all CSIDH primes $p \equiv 3 \pmod 8$.

Proof. By the division in the proof of Lemma 3.7, $\delta = 0$ means that $C^* - \left(\frac{x}{3} + \frac{\alpha_2}{9}\right)(C^{*\prime} - \kappa)$ is a constant, which vanishes at x_0. Therefore $C^* = \left(\frac{x}{3} + \frac{\alpha_2}{9}\right)(C^{*\prime} - \kappa)$, and the mean $-\alpha_2/3$ of the three roots is a root. It is not x_0, since the other roots of C^* would then be the roots of $C^{*\prime} - \kappa$, which include x_0. Hence it is e_j, the midpoint of x_0 and e_k. Conversely, if $2e_j = x_0 + e_k$ then $C^{*\prime}(x_0) = (x_0 - e_j)(x_0 - e_k) = \frac{1}{2}(x_0 - e_k)^2 = C^{*\prime}(e_k)$, so $C^{*\prime} - \kappa$ has the roots x_0 and e_k of C^*, and the degree-1 remainder vanishes at two points, i.e. $\delta = 0$. In the coordinate $X = \omega(x^* - x_0)$ of the codomain, the roots become 0, $r = \omega(e_j - x_0)$ and $2r$, so $A' = -3r$ and $B' = 2r^2$, i.e. $2A'^2 = 9B'$; conversely, $2A'^2 = 9B'$ forces the nonzero roots to be r and $2r$. Translating by r gives $y^2 = x^3 - r^2 x$, with $j = 1728$, and the automorphism $(x,y) \mapsto (-x, iy)$ of the translated curve fixes the point $x = r$, i.e. e_j, and swaps the other two. Finally $2B' = (2r)^2$ and $B' = B^N$. □

In particular, $\delta = 0$ only occurs when a step lands on the curve of j-invariant 1728, for instance on the surface with $p \equiv 7 \pmod 8$, where $y^2 = x^3 - x$ is the starting curve of CSURF [7]. In that case x_0 is recovered from $A' = \omega(\frac{b'_2}{4} + 3x_0)$ once A' is known by other means: by the twisted Edwards formula of Section 4.3 for Montgomery curves ($N + 2\mathbf{E}_N + 9$ $\mathbf{M}$ instead of the 4 $\mathbf{M}$ of line 8 of Algorithm 4), or by Proposition 4.12 below in general ($3N + 2\mathbf{E}_N + 3\mathbf{L}_N + 20$ $\mathbf{M}$ instead of the 7 $\mathbf{M}$ of line 8 of Algorithm 6). Our implementation detects $\delta = 0$ (a comparison, no extra multiplication) and switches to this fallback.

Proposition 4.11. *Algorithm 6 is correct and costs $\frac{1}{2}(23N + 113) + 2\mathbf{E}_N$ $\mathbf{M}$, plus $\frac{1}{2}(5N + 5)$ $\mathbf{M}$ per pushed point.*

Proof. Lines 1–3 compute the input of Algorithm 1, as for Montgomery curves but with the general xADD. Write Z_1° for the input value of Z_1. With the representative $(CX_1 : CZ_1)$ of line 1, line 3 computes $\Delta_{k-1} = X_{k-1}Z_1 - X_1 Z_{k-1} = CZ_{k-1}Z_1^\circ(x_{k-1} - x_1)$ and $t_1 - t_0 Z_{k-1} = CZ_{k-1}Z_1^\circ(x_{k-1}x_1 - B)$, so that $X_k/Z_k = x_k$ by $x_k x_{k-2} = (x_{k-1}x_1 - B)^2/(x_{k-1} - x_1)^2$, and the differences are again a by-product. Since $Z_k = X_{k-2}\Delta_{k-1}^2$, Lemma 3.6(3) applies with $c = 1$. Line 4 is the radical core. Lines 5–8 normalize the codomain to E_{A',B^N}. They are Theorem 4.9 in the scaled coordinates, as in the proof of Proposition 4.6, except that now $\bar{C}'(\bar{y}_0) = \vartheta^2 B^N/\omega^2 = 4\Psi^2 B_n^N/(\Pi_X^4 C^N)$, and Lemma 3.7 with numerator and denominator multiplied by $\Pi_X^4 C^N$ gives $\bar{y}_0 = -n_0/\delta$, so $A' = \Pi_X^2(\bar{a}\delta - 3n_0)/(2\Psi\delta)$, $B' = B_n^N/C^N$ and $x(P') = \Pi_X^2 n_0/(2\Psi\delta)$. Line 9 pushes points through the map ϕ_B of Lemma 4.8: since $u_i - t_2 Z_i = CX_R X_i - B_n Z_R Z_i$ and $u_i - t_3 X_i = C(X_R Z_i - Z_R X_i)$, the two factors are the numerator and denominator of $(xx_i - B)/(x - x_i)$, homogenized over the common denominator C, and three products per index give both. This costs $5\mathbf{M}$ per index, minus $2\mathbf{M}$ for the first factors of the products, and $7\mathbf{M}$ for t_1, t_2, t_3 and the output. Compared to Algorithm 4, the multiples cost $N - 1$ $\mathbf{M}$ more (line 1 and $2\mathbf{M}$ per xADD), the powers $2\mathbf{E}_N$, and lines 6 and 8 together $5\mathbf{M}$ more. Line 7 is Proposition 4.10: $x(P') = \left(\Pi_X^2 \bar{a}/(2\Psi) - A'\right)/3 = (\Pi_X^2 \bar{a} A_2 - 2\Psi A_1)/(6\Psi A_2)$, and the coefficients are written over the same denominator $6\Psi A_2$. □

Algorithm 6 Radical step on $E_{A,B}\colon y^2 = x(x^2 + Ax + B)$

Input: $(A_n : B_n : C)$ with $A = A_n/C$, $B = B_n/C$, $(X_1 : Z_1) = x(P)$ with P of order N; points $x(R_j) = (X_{R_j} : Z_{R_j})$

Output: $(A'_n : B'_n : C')$ with $B' = B^N$, $x(P')$, and $x(\phi_B(R_j))$

$t_0 \leftarrow B_n Z_1$, $(X_1, Z_1) \leftarrow (CX_1, CZ_1)$	3
$F_n, G_n \leftarrow 4X_1(CX_1^2 + A_n X_1 Z_1 + B_n Z_1^2)$, $3CX_1^2 + 2A_n X_1 Z_1 + B_n Z_1^2$, $(X_2 : Z_2) \leftarrow \big((CX_1^2 - B_n Z_1^2)^2 : CZ_1 F_n\big)$	10
$t_1, t_2 \leftarrow X_{k-1}X_1$, $Z_{k-1}Z_1$, $\Delta_{k-1} \leftarrow (X_{k-1} - Z_{k-1})(X_1 + Z_1) - t_1 + t_2$; if $k \le (N-1)/2$: $X_k, Z_k \leftarrow Z_{k-2}(t_1 - t_0 Z_{k-1})^2$, $X_{k-2}\Delta_{k-1}^2$, for $3 \le k \le (N+1)/2$	$4N - 17$
$\Pi_{\Delta,k} \leftarrow \prod_{j=2}^{k} \Delta_j$ for $2 \le k \le h$, $\Pi_{X,h-2} \leftarrow \prod_{i=1}^{h-2} X_i$, $\Pi_D \leftarrow Z_2 \Pi_{X,h-2} \Pi_{\Delta,h-1}^2$ (Lemma 3.6(3)); Algorithm 1 on $(F_n, G_n, C, \Pi_{\Delta,h}^2, (\Delta_k, Z_k))$	$\frac{1}{2}(15N + 73)$
Π_X and Ψ as in Algorithm 4, B_n^N, C^N	$2\mathbf{E}_N + 7$
$n_0 \leftarrow \Pi_X^4 C^N (9\bar{c} - \bar{a}\bar{b}) + 4\bar{a}\Psi^2 B_n^N$, $\delta \leftarrow \Pi_X^4 C^N (6\bar{b} - 2\bar{a}^2) + 12\Psi^2 B_n^N$	10
if $\delta = 0$: $(A_1 : A_2)$ with $A' = A_1/A_2$ from Proposition 4.12, $t_1 \leftarrow 6\Psi A_2$, $(A'_n : B'_n : C') \leftarrow (6\Psi A_1 C^N : B_n^N t_1 : t_1 C^N)$, $x(P') \leftarrow (\Pi_X^2 \bar{a} A_2 - 2\Psi A_1 : t_1)$ (Proposition 4.10)	$3N + 2\mathbf{E}_N + 3\mathbf{L}_N + 20$
else: $t_1 \leftarrow 2\Psi\delta$, $(A'_n : B'_n : C') \leftarrow \big(\Pi_X^2(\bar{a}\delta - 3n_0)C^N : B_n^N t_1 : t_1 C^N\big)$, $x(P') \leftarrow (\Pi_X^2 n_0 : t_1)$	7
Pushing, for each $R_j = (X_R : Z_R)$ (no precomputation): $t_1 \leftarrow CX_R$, $t_2 \leftarrow t_1 + B_n Z_R$, $t_3 \leftarrow t_1 + CZ_R$; $u_i \leftarrow t_1(X_i + Z_i)$ for $1 \le i \le (N-1)/2$; $x(\phi_B(R_j)) \leftarrow \big(X_R(\prod_i (u_i - t_2 Z_i))^2 : Z_R(\prod_i (u_i - t_3 X_i))^2\big)$	$\frac{1}{2}(5N + 5)$ each

Proposition 4.12 (Codomain only)**.** *Let e be the class of x in $L = K[x]/(x^2 + Ax + B)$, let $^-$ denote the conjugation of L/K, so that $e, \bar{e}$ are the roots of $x^2 + Ax + B$ in L, and put $\pi_e = \prod_{i=1}^{(N-1)/2}(x_i - e) \in L$ as in the proof of Lemma 4.8. Then the codomain of ϕ_B is E_{A',B^N} with*

$$A' = -\frac{\mathrm{Tr}_{L/K}\left(e^N \pi_{\bar{e}}^4\right)}{\mathrm{N}_{L/K}(\pi_e)^2}.$$

Computed in the basis $1, Ce$ of L, where $(Ce)^2 + A_n\, Ce + CB_n = 0$, this costs $7N + 4\mathbf{E}_N + 3\mathbf{L}_N + 7\,\mathbf{M}$ and no root; the power $(Ce)^N$ in L accounts for $2\mathbf{E}_N + 3\mathbf{L}_N - 5$.

Proof. By (4), the images of the two other 2-torsion points are

$$\phi_B(e) = e^N \frac{\pi_{\bar{e}}^2}{\pi_e^2} = \frac{e^N \pi_{\bar{e}}^4}{\mathrm{N}(\pi_e)^2}$$

and its conjugate, and their sum is $-A'$. This holds whether or not $e \in K$. The multiples cost $4N - 8\ \mathbf{M}$ (as in Algorithm 6, without F_n and Δ_k). Each factor $CX_i - Ce\, Z_i$ is multiplied into π_e with $6\mathbf{M}$ (Karatsuba and two multiplications by A_n and CB_n), a squaring in L costs $5\mathbf{M}$ and a multiplication by Ce costs $2\mathbf{M}$. □

4.6 Huff curves

A Huff curve $H_{a,b}\colon ax(y^2 - 1) = by(x^2 - 1)$ exists if and only if $E(K)$ contains a subgroup $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/4\mathbb{Z}$ [23]; only the ratio $t = a/b$ matters. Its neutral element is $(0, 0)$, negation is $(x, y) \mapsto (-x, -y)$, and the natural x-line coordinate is $m = y/x$. Isogenies between Huff curves were given in [31], and Huff-based isogeny cryptography was studied in [25, 41]. For CSIDH this model is only available on the surface ($p \equiv 7 \pmod 8$).

Lemma 4.13. *Put $\tilde{x} = (am - b)/(a - bm)$. Then $\tilde{x}$ is the x-coordinate of the Montgomery model $(ab)^{-3}y^2 = \tilde{x}^3 + \tilde{A}\tilde{x}^2 + \tilde{x}$ with $\tilde{A} = t + t^{-1} = (a^2 + b^2)/(ab)$, and $m = (t\tilde{x} + 1)/(\tilde{x} + t)$.*

Proof. By [23], $U = ab(am - b)/(a - bm)$ maps $H_{a,b}$ to $V^2 = U(U + a^2)(U + b^2)$, and $U = ab\tilde{x}$ gives $V^2 = (ab)^3\tilde{x}(\tilde{x} + t)(\tilde{x} + t^{-1})$. Solving for m gives the inverse. □

Since x-only arithmetic and Algorithm 4 do not depend on the coefficient of y^2, the radical step on $\tilde{x}$ gives $\tilde{A}'$ and $\tilde{x}(P')$. It remains to recover t' from $\tilde{A}' = t' + 1/t'$ without a square root.

Theorem 4.14. *Let $P \in H_{a,b}(K)$ have odd order $N \geq 5$, let $(X_i : Z_i) = \tilde{x}([i]P)$ for $1 \leq i \leq (N - 1)/2$, and put*

$$w_n = (a + b)^N \prod_{i=1}^{(N-1)/2} (X_i + Z_i)^4, \qquad w_d = (a - b)^N \prod_{i=1}^{(N-1)/2} (X_i - Z_i)^4.$$

Then the codomain of the step is $H_{a',b'}$ with $(a' : b') = (w_n + w_d : w_n - w_d)$, and $m(P') = (a'\tilde{X}' + b'\tilde{Z}')/(b'\tilde{X}' + a'\tilde{Z}')$, where $(\tilde{X}' : \tilde{Z}') = \tilde{x}(P')$ is the output of Algorithm 4.

Proof. The codomain of Algorithm 4 is the Costello–Hisil codomain, whose coefficient satisfies the twisted Edwards formula of [29, 31]: $\frac{\tilde{A}'+2}{\tilde{A}'-2} = \left(\frac{\tilde{A}+2}{\tilde{A}-2}\right)^N \prod_i \left(\frac{X_i+Z_i}{X_i-Z_i}\right)^8$. With $\tilde{A} = t + t^{-1}$ we have $\frac{\tilde{A}+2}{\tilde{A}-2} = \left(\frac{t+1}{t-1}\right)^2 = \left(\frac{a+b}{a-b}\right)^2$, so $\frac{\tilde{A}'+2}{\tilde{A}'-2} = (w_n/w_d)^2$. The value $t' = (w_n + w_d)/(w_n - w_d)$ satisfies $\frac{t'+1}{t'-1} = \frac{w_n}{w_d}$, hence $t' + 1/t' = \tilde{A}'$ because $z \mapsto \frac{z+2}{z-2}$ is injective. The other root $1/t'$ corresponds to the isomorphic curve $H_{b',a'}$ (swap x and y). This also fixes the quadratic twist. The points with $\tilde{x} = \pm 1$ are the points with $m = \pm 1$, i.e. $(\pm 1, \pm 1)$, which are rational points of order 4 on every Huff curve, and the map ϕ of Lemma 4.4 satisfies $\phi(\pm 1) = \pm 1$. So the codomain of the step has a rational point with $\tilde{x}' = 1$, and since $\tilde{A}' \neq -2$, exactly one of the two quadratic twists with Montgomery coefficient $\tilde{A}'$ has one; as $H_{a',b'}$ does, it is that twist. The formula for $m(P')$ is Lemma 4.13. □

Algorithm 7 Radical step on Huff curves

Input: $(a : b)$ for $H_{a,b}$, $(M_1 : M_2) = m(P)$ with P of order N; points $m(R_j)$
Output: $(a' : b')$, $m(P')$, and the images $m(\varphi(R_j))$

1:	$(A_n : C) \leftarrow (a^2 + b^2 : ab)$	3
2:	$t_1, t_2 \leftarrow (a-b)(M_1+M_2)$, $(a+b)(M_1-M_2)$, $(X_1 : Z_1) \leftarrow (t_1+t_2 : t_1-t_2)$, and so for all R_j	2 (+2 each)
3:	Algorithm 4, keeping the multiples $(X_i : Z_i)$	$\frac{1}{2}(21N+105)$; $2N$ each
4:	$w_n \leftarrow (a+b)^N \prod_i (X_i+Z_i)^4$, $w_d \leftarrow (a-b)^N \prod_i (X_i-Z_i)^4$	$N + 2\mathbf{E}_N + 3$
5:	$(a' : b') \leftarrow (w_n + w_d : w_n - w_d)$	0
6:	$t_1, t_2 \leftarrow (a'+b')(X+Z)$, $(a'-b')(X-Z)$, $m \leftarrow (t_1+t_2 : t_1-t_2)$ for $\tilde{x} = (X : Z)$ the image of P and of all R_j	2 (+2 each)

Proposition 4.15. *Algorithm 7 is correct and costs* $\frac{1}{2}(23N+125) + 2\mathbf{E}_N$ **M**, *plus* $2N+4$ **M** *per pushed point. The codomain-only variant, with only the multiples* $(X_i : Z_i)$ *of Algorithm 4 in line 3 (as in its codomain-only variant), costs* $4N + 2\mathbf{E}_N + 2$ **M**.

Proof. Line 1 computes the Montgomery coefficient $\tilde{A} = (a^2+b^2)/(ab)$ of Lemma 4.13, and line 2 converts m to $\tilde{x}$: $t_1 + t_2 = 2(aM_1 - bM_2)$ and $t_1 - t_2 = 2(aM_2 - bM_1)$. Line 3 is the Montgomery step, which also returns the multiples and pushes points. Lines 4–5 normalize the codomain by Theorem 4.14, recovering $(a' : b')$ from $\tilde{A}'$ without a square root, and line 6 converts $\tilde{x}$ back to m with Lemma 4.13 on the codomain. □

4.7 Kummer lines

Gaudry and Lubicz [20] and Karati and Sarkar [24] work with the Kummer line of a curve in squared theta coordinates. Let $a^2, b^2 \in K^*$ with $a^4 \neq b^4$, and put $A^2 = a^2 + b^2$, $B^2 = a^2 - b^2$. The Kummer line $\mathcal{K}_{a^2,b^2}$ is $\mathbb{P}^1$ with points $[X : Z] = [x^2 : z^2]$, identity $[a^2 : b^2]$, and doubling [24, Table 2]

$$[2][X : Z] = \left[b^2(s_0+t_0)^2 : a^2(s_0-t_0)^2\right], \qquad s_0 = B^2(X+Z)^2, \quad t_0 = A^2(X-Z)^2,$$

together with the matching differential addition. By [24], it is the x-line of the Legendre curve with parameter $a^4/(a^4-b^4)$. Robert and Sarkis [37] computed 2-isogenies between such Kummer lines; we consider odd degrees.

Proposition 4.16. *The coordinate*

$$\tilde{x} = \frac{a^2X - b^2Z}{a^2Z - b^2X}$$

identifies $\mathcal{K}_{a^2,b^2}$ *with the x-line of the Montgomery curve with* $\tilde{A} = a^2/b^2 + b^2/a^2$. *Equivalently,* $\mathcal{K}_{a^2,b^2}$ *is the x-line of the Huff curve* H_{a^2,b^2} *with* $m = X/Z$ *(Lemma 4.13).*

Proof. Write $m = X/Z$ and $\tilde{A} = (a^4+b^4)/(a^2b^2)$. The map $m \mapsto \tilde{x}$ is a Möbius transformation (its determinant is $a^4 - b^4 \neq 0$) sending the identity $[a^2 : b^2]$ to ∞, and

$$\tilde{x} + 1 = \frac{B^2(X+Z)}{a^2Z - b^2X}, \qquad \tilde{x} - 1 = \frac{A^2(X-Z)}{a^2Z - b^2X}, \qquad m = \frac{a^2\tilde{x} + b^2}{b^2\tilde{x} + a^2}.$$

It therefore suffices to show that the doubling and the differential addition of [24] become the Montgomery formulae for $\tilde{A}$ in the coordinate $\tilde{x}$.

Differential addition. For $[X_1 : Z_1]$, $[X_2 : Z_2]$ with difference $[X_d : Z_d]$, [24, Table 2] puts $s_0 = B^2(X_1+Z_1)(X_2+Z_2)$, $t_0 = A^2(X_1-Z_1)(X_2-Z_2)$ and $[X_3 : Z_3] = [Z_d(s_0+t_0)^2 :$

$X_d(s_0-t_0)^2]$, i.e. $m_3m_d = \big((s_0+t_0)/(s_0-t_0)\big)^2$. Put $v_+ = \tilde{x}_1\tilde{x}_2+1$, $v_1 = \tilde{x}_1+\tilde{x}_2$ and $v_2 = \tilde{x}_1\tilde{x}_2$. By the displayed identities, s_0 and t_0 are a common factor times $A^2(v_+ + v_1)$ and $B^2(v_+ - v_1)$, so

$$m_3m_d = \frac{(a^2v_+ + b^2v_1)^2}{(b^2v_+ + a^2v_1)^2}.$$

On the Montgomery side, $\tilde{x}_3\tilde{x}_d = (v_2-1)^2/(\tilde{x}_1-\tilde{x}_2)^2$ and $\tilde{x}_3+\tilde{x}_d = 2\big(v_+v_1+2\tilde{A}v_2\big)/(\tilde{x}_1-\tilde{x}_2)^2$. Using $(v_2-1)^2 = v_+^2-4v_2$, $(\tilde{x}_1-\tilde{x}_2)^2 = v_1^2-4v_2$ and $a^2b^2\tilde{A} = a^4+b^4$,

$$\begin{aligned}(\tilde{x}_1-\tilde{x}_2)^2(a^2\tilde{x}_3+b^2)(a^2\tilde{x}_d+b^2) &= a^4(v_+^2-4v_2) + 2a^2b^2(v_+v_1+2\tilde{A}v_2) + b^4(v_1^2-4v_2)\\ &= (a^2v_+ + b^2v_1)^2,\\ (\tilde{x}_1-\tilde{x}_2)^2(b^2\tilde{x}_3+a^2)(b^2\tilde{x}_d+a^2) &= b^4(v_+^2-4v_2) + 2a^2b^2(v_+v_1+2\tilde{A}v_2) + a^4(v_1^2-4v_2)\\ &= (b^2v_+ + a^2v_1)^2,\end{aligned}$$

so the Montgomery $\tilde{x}_3$ satisfies the Kummer relation for m_3m_d. Since this relation determines m_3 from m_d, the two differential additions agree.

Doubling. Here s_0+t_0 and s_0-t_0 are a common factor times $2(a^2u+2b^2\tilde{x})$ and $2(b^2u+2a^2\tilde{x})$, with $u = \tilde{x}^2+1$. Hence $m(2P) = b^2(a^2u+2b^2\tilde{x})^2/\big(a^2(b^2u+2a^2\tilde{x})^2\big)$, and

$$\begin{aligned}\tilde{x}(2P) &= \frac{a^2b^2\big((a^2u+2b^2\tilde{x})^2 - (b^2u+2a^2\tilde{x})^2\big)}{a^4(b^2u+2a^2\tilde{x})^2 - b^4(a^2u+2b^2\tilde{x})^2}\\ &= \frac{(a^4-b^4)a^2b^2(u^2-4\tilde{x}^2)}{4(a^4-b^4)a^2b^2\tilde{x}(u+\tilde{A}\tilde{x})} = \frac{(\tilde{x}^2-1)^2}{4\tilde{x}(\tilde{x}^2+\tilde{A}\tilde{x}+1)},\end{aligned}$$

which is the Montgomery doubling. The Huff statement is Lemma 4.13 with (a,b) replaced by (a^2,b^2). □

In particular $[1:1]$ and $[-1:1]$ are the x-coordinates $\tilde{x} = \pm 1$ of the points of order 4 above $(0,0)$; the squared theta structure is a level-4 structure on the x-line. Proposition 4.16 reduces radical steps on Kummer lines to Theorem 4.14.

Algorithm 8 Radical step on Kummer lines

Input: $(a^2:b^2)$ for $\mathcal{K}_{a^2,b^2}$, $[X:Z]$ of P of order N; points $[X_{R_j}:Z_{R_j}]$
Output: $(a'^2:b'^2)$, $[X':Z']$ of P', and the images of the R_j

1: Algorithm 7 with $(a:b) \leftarrow (a^2:b^2)$ and $(M_1:M_2) \leftarrow (X:Z)$, and likewise for the R_j	$\frac{1}{2}(23N+125)+2\mathbf{E}_N$; $2N+4$ each
2: $(a'^2:b'^2) \leftarrow (a':b')$, $[X':Z'] \leftarrow m(P')$, and likewise for the images	0

Corollary 4.17. *Algorithm 8 is correct: the codomain is $\mathcal{K}_{a'^2,b'^2}$ with*

$$(a'^2:b'^2) = (w_n+w_d : w_n-w_d),$$
$$w_n = (a^2+b^2)^N \prod_{i=1}^{(N-1)/2}(X_i+Z_i)^4, \qquad w_d = (a^2-b^2)^N \prod_{i=1}^{(N-1)/2}(X_i-Z_i)^4,$$

for the multiples $(X_i:Z_i)$ of $\tilde{x}(P)$, and P' has Kummer coordinate $[a'^2\tilde{X}'+b'^2\tilde{Z}' : b'^2\tilde{X}'+a'^2\tilde{Z}']$. The costs are those of Algorithm 7.

Proof. By Proposition 4.16, the Kummer line $\mathcal{K}_{a^2,b^2}$ is the x-line of the Huff curve H_{a^2,b^2} in the coordinate $m = X/Z$. Thus line 1 of Algorithm 8 is Algorithm 7 for $(a:b) = (a^2:b^2)$, and the codomain and P' follow from Theorem 4.14; line 2 only renames the output. □

A Kummer line $\mathcal{K}_{a^2,b^2}$ with $a^2,b^2 \in K$ exists for E if and only if E or its quadratic twist has a Montgomery model with full rational 2-torsion, i.e. the condition for Huff curves up to twist.

4.8 Jacobi quartic curves

An extended Jacobi quartic $J_{a,d}\colon y^2 = dx^4 + 2ax^2 + 1$ exists if and only if $E(K)$ has a point of order 2 [5,21], and isogenies on this model were treated in [18]. Negation is $(x, y) \mapsto (-x, y)$, the neutral element is $(0, 1)$ and $(0, -1)$ has order 2. We use the x-line coordinate $w = (1 + y)/x^2$.

Lemma 4.18. *The map $(x, y) \mapsto (u, v) = \big(2w + 2a,\ 2u/x\big)$ is an isomorphism from $J_{a,d}$ to $E_{A,B}$ with $A = -4a$ and $B = 4(a^2 - d)$, sending $(0, 1)$ to $\mathcal{O}$ and $(0, -1)$ to $(0, 0)$.*

Proof. We have $u^2 - 4au + 4(a^2 - d) = (u - 2a)^2 - 4d = 4(w^2 - d)$ and $v^2 = 4u^2/x^2$, so the claim is $u/x^2 = w^2 - d$. Multiplying by x^4, this reads $2(1 + y) + 2ax^2 = (1 + y)^2 - dx^4$, which is the curve equation. At $(0, -1)$ we have $w = (dx^2 + 2a)/(y - 1) \to -a$, so $u \to 0$. The map is invertible ($x = 2u/v$), and since it sends the neutral element to $\mathcal{O}$ it is a group isomorphism. □

Corollary 4.19. *Let $P \in J_{a,d}(K)$ have odd order $N \geq 5$ and $w(P) = w_1$. Apply Theorem 4.9 to $(E_{-4a,4(a^2-d)}, u_1 = 2w_1 + 2a)$, with output A' and $u(P')$. Then the codomain is $J_{a',d'}$ with*

$$a' = -\frac{A'}{4}, \qquad a'^2 - d' = 4^{N-1}(a^2 - d)^N, \qquad w(P') = \frac{u(P') - 2a'}{2}.$$

Algorithm 9 costs $\frac{1}{2}(23N + 135) + 2\mathbf{E}_N$ $\mathbf{M}$, plus $\frac{1}{2}(5N + 17)$ $\mathbf{M}$ per pushed point, and the codomain-only variant costs $7N + 4\mathbf{E}_N + 3\mathbf{L}_N + 15$ $\mathbf{M}$ by Proposition 4.12.

Proof. In Algorithm 9, line 1 converts the coordinate w into the coordinate $u = 2w + 2a$ of Lemma 4.18 on the curve $E_{-4a,4(a^2-d)}$. In the weighted projective form $a = a_n/C$ and $d = d_n/C^2$ of the algorithm, this curve has $A_n = -4a_nC$ and $B_n = 4(a_n^2 - d_n)$ over the denominator C^2. Line 2 is Algorithm 6 on this model; it fixes the image of $(0, 0)$, which is the distinguished point of order 2 of $J_{a,d}$. Finally, lines 3 and 4 apply Lemma 4.18 to the codomain: $a' = -A'/4$, $d' = a'^2 - B'/4$ with $B' = B^N$, and $w = (u - 2a')/2$. □

Algorithm 9 Radical step on extended Jacobi quartics

Input: $(a_n : d_n : C)$ with $a = a_n/C$, $d = d_n/C^2$, $(W_1 : V_1) = w(P)$ with P of order N; points $w(R_j)$

Output: $(a'_n : d'_n : C')$, $w(P')$, and the images $w(\varphi(R_j))$

1:	$(X_1 : Z_1) \leftarrow (2(W_1C + a_nV_1) : V_1C)$, and so for all R_j	3 (+3 each)
2:	Algorithm 6 on $(-4a_nC : 4(a_n^2 - d_n) : C^2)$, including its pushing line 9	$\frac{1}{2}(23N + 119) + 2\mathbf{E}_N$; $\frac{1}{2}(5N + 5)$ each
3:	$(a'_n, d'_n) \leftarrow \big(-A'_n/4,\ (A'^2_n - 4B'_nC')/16\big)$, keeping C'	2
4:	$w \leftarrow (XC' - 2a'_nZ : 2ZC')$ for $x = (X : Z)$ the image of P and of all R_j	3 (+3 each)

4.9 Legendre curves and Jacobi intersections

A Legendre curve is $E_\lambda\colon y^2 = x(x-1)(x-\lambda)$ with $\lambda \neq 0, 1$. It exists if and only if $E[2] \subset E(K)$, the three 2-torsion points being normalized to $x = 0, 1, \lambda$. In the notation of Section 4.5, $E_\lambda = E_{-(1+\lambda),\lambda}$, but the step of Theorem 4.9 does not return a Legendre curve, because it only fixes the image of $(0, 0)$. Two more norms restore the normalization.

Theorem 4.20. *Let $P \in E_\lambda(K)$ have odd order $N \geq 5$, let $x_i = x([i]P)$ for $1 \leq i \leq (N-1)/2$, and put $\pi_1 = \prod_{i=1}^{(N-1)/2}(x_i - 1)$ and $\pi_\lambda = \prod_{i=1}^{(N-1)/2}(x_i - \lambda)$. Apply Theorem 4.9 to $E_{-(1+\lambda),\lambda}$ and $x(P)$, with output $x(P')$. Then the codomain is $E_{\lambda'}$ with*

$$\lambda' = \lambda^N\Big(\frac{\pi_1}{\pi_\lambda}\Big)^4, \qquad x_{\mathrm{Leg}}(P') = \Big(\frac{\pi_1}{\pi_\lambda}\Big)^2 x(P').$$

Proof. By (4) with $e = 1$ and $\bar{e} = \lambda$, the map ϕ_B of Lemma 4.8 sends the 2-torsion points with $x = 1$ and $x = \lambda$ to $e'_1 = (\pi_\lambda/\pi_1)^2$ and $e'_2 = \lambda^N(\pi_1/\pi_\lambda)^2$, and $(0, 0)$ to 0. Dividing x by e'_1 gives the Legendre form with $\lambda' = e'_2/e'_1$. Since e'_1 is a square, this rescaling is defined over K and does not twist the curve, and $x(P')$ is rescaled in the same way. □

Algorithm 10 Radical step on Legendre curves and Jacobi intersections

Input: $(\lambda_n : C)$ with $\lambda = \lambda_n/C$, $(X_1 : Z_1) = x(P)$ with P of order N; points $x(R_j)$ (for J_a, see Corollary 4.24)

Output: $(\lambda'_n : C')$, $x(P')$, and the images $x(\varphi(R_j))$ in Legendre normalization

1: Algorithm 6 on $(-(\lambda_n + C) : \lambda_n : C)$, with output $(X' : Z')$ and $(X'_j : Z'_j)$, including its pushing line 9	$\frac{1}{2}(23N + 113) + 2\mathbf{E}_N$; $\frac{1}{2}(5N + 5)$ each
2: $\pi_1 \leftarrow \prod_i (X_i - Z_i)$, $\pi_\lambda \leftarrow \prod_i (CX_i - \lambda_n Z_i)$	$2N - 4$
3: $(\lambda'_n : C') \leftarrow (\lambda_n^N C^N \pi_1^4 : C^2 \pi_\lambda^4)$, $t_1 \leftarrow C^N \pi_1^2$, $t_2 \leftarrow C\pi_\lambda^2$, with λ_n^N, C^N from line 5 of Algorithm 6	10
4: $x(P') \leftarrow (t_1 X' : t_2 Z')$, $x(\varphi(R_j)) \leftarrow (t_1 X'_j : t_2 Z'_j)$	2 (+2 each)

Proposition 4.21. *Algorithm 10 is correct and costs* $\frac{1}{2}(27N + 129) + 2\mathbf{E}_N$ **M**, *plus* $\frac{1}{2}(5N + 9)$ **M** *per pushed point. The codomain-only variant costs* $6N + \mathbf{E}_N - 4$ **M**.

Proof. Line 1 is the step on $E_{-(1+\lambda),\lambda}$, which normalizes the codomain only up to the image of $(0, 0)$. Lines 2–3 place the other two points of order 2 at 1 and λ', and line 4 rescales x accordingly; this is Theorem 4.20 in projective form. The products of line 2 equal those of Theorem 4.20 up to the factors $\prod_i Z_i$ and $C^{(N-1)/2} \prod_i Z_i$, so the ratio π_1/π_λ of the theorem is $C^{(N-1)/2}\pi_1/\pi_\lambda$ in terms of line 2. Hence $\lambda' = \lambda_n^N C^{N-2} \pi_1^4/\pi_\lambda^4$, which is line 3 after multiplying numerator and denominator by C^2, and $x_{\mathrm{Leg}}(P') = C^{N-1}\pi_1^2 x(P')/\pi_\lambda^2 = t_1 x(P')/t_2$. Pushed points are rescaled in the same way. The codomain-only variant uses only the multiples of Algorithm 6 ($4N - 8$ **M**), line 2, and $\lambda' = \big((\lambda_n C)^N \pi_1^4 : C^2 \pi_\lambda^4\big)$. □

Remark 4.22. If $\lambda = k^2$ is a square in K, then x/k is a Montgomery x-coordinate with $A = -(k + k^{-1})$, and the step can be done as on Huff curves (Section 4.6) with $t = -k$, at $\frac{1}{2}(23N + 125) + 2\mathbf{E}_N$ **M**. In general, λ is not a square.

A Jacobi intersection is the space curve $J_a \colon s^2 + c^2 = 1$, $as^2 + d^2 = 1$ with $a \neq 0, 1$ [12,28]. Its neutral element is $\mathcal{O} = (0, 1, 1)$, negation is $(s, c, d) \mapsto (-s, c, d)$, and the three points $(0, -1, -1)$, $(0, -1, 1)$, $(0, 1, -1)$ form the full 2-torsion. Therefore J_a exists if and only if $E[2] \subset E(K)$, and for CSIDH it is only available on the surface.

Lemma 4.23. *The function*

$$x = \frac{1 + d}{1 - c} = \frac{(1 + c)(1 + d)}{s^2} = \frac{a(1 + c)}{1 - d}$$

is an x-coordinate on J_a: it has a double pole at $\mathcal{O}$ and no other poles, and it takes the values $0, 1, a$ at $(0, -1, -1)$, $(0, -1, 1)$, $(0, 1, -1)$. Hence the x-line of J_a is that of the Legendre curve E_a, and conversely

$$c = \frac{x^2 - 2x + a}{x^2 - a}, \qquad d = \frac{x^2 - 2ax + a}{x^2 - a}.$$

Proof. The three expressions agree because $s^2 = 1 - c^2$ and $as^2 = 1 - d^2$. Poles can only occur where $s = 0$. Near $\mathcal{O}$ the parameter s gives $1 - c = s^2/2 + O(s^4)$, so $x = 4/s^2 + O(1)$. At $(0, 1, -1)$ the third expression gives $x = a$, at $(0, -1, 1)$ the first gives $x = 1$, and at $(0, -1, -1)$ the second gives $x = as^2/4 + O(s^4)$. Thus x has degree 2, is even, and has a double pole at $\mathcal{O}$, which makes it the x-coordinate of a Weierstrass model whose 2-torsion points have $x = 0, 1, a$, i.e. of a quadratic twist of $y^2 = x(x - 1)(x - a)$. The inverse follows by eliminating $d = x(1 - c) - 1$ from $d^2 = 1 - a + ac^2$ and discarding the root $c = 1$. □

Corollary 4.24. *Let $P \in J_a(K)$ have odd order $N \geq 5$. Apply Theorem 4.20 to $\lambda = a$ and $x(P) = (z+d)/(z-c)$ for $P = (s : c : d : z)$ in projective coordinates. Then the codomain is $J_{a'}$ with $a' = \lambda'$, and $x(P')$ is $x_{\mathrm{Leg}}(P')$. In particular, Algorithm 10 applies, since the conversions are additions.*

Proof. Lemma 4.23 on both sides: J_a and $J_{a'}$ have the x-lines of E_a and $E_{a'}$, with the 2-torsion points in the same positions. □

4.10 Twisted Hessian curves

A twisted Hessian curve is $H_{a,d}\colon aX^3 + Y^3 + Z^3 = dXYZ$ with $a(d^3 - 27a) \neq 0$. Its neutral element is $\mathcal{O} = (0 : -1 : 1)$, negation is $(X : Y : Z) \mapsto (X : Z : Y)$, and every elliptic curve with a rational point of order 3 has such a model [2]. Isogenies between twisted Hessian curves were given in [15, 35]. We assume $\gcd(N, 6) = 1$ and $\operatorname{char} K \neq 3$, and use the x-line coordinate $\theta = X/(Y + Z)$.

The tangent line at $\mathcal{O}$ is $W = dX + 3Y + 3Z = 0$, and the curve equation can be rewritten as

$$YZ \cdot W = aX^3 + (Y + Z)^3. \tag{5}$$

The nine flexes lie on the lines $X = 0$, $Y = 0$, $Z = 0$; the points with $X = 0$ form the subgroup $\mathcal{K} = \{\mathcal{O}, (0 : -\zeta : 1), (0 : -\zeta^2 : 1)\}$ for a primitive cube root of unity ζ, so x-coordinates centered at $\mathcal{K}$ are rational even when its points are not.

Lemma 4.25 (Weierstrass model)**.** *Put $k = d^3 - 27a$. Then $x = kX/(3W)$, $y = kY/W$ define an isomorphism from $H_{a,d}$ to*

$$E^H_{a,d}\colon y^2 + dxy - \tfrac{k}{3}y = x^3 - d^2x^2 + \tfrac{kd}{3}x - \tfrac{k^2}{27},$$

with $b_2 = -3d^2$, $b_4 = kd/3$, $b_6 = -k^2/27$ and $b_8 = 0$. In particular $x = 0$ on $\mathcal{K} \setminus \{\mathcal{O}\}$, and

$$x = \frac{k\theta}{3(d\theta + 3)}, \qquad \theta = \frac{9x}{k - 3dx}.$$

Proof. The map is linear in $(X : Y : Z)$ and sends the flex $\mathcal{O}$ to the point at infinity with tangent line at infinity, so the image is a Weierstrass cubic; substituting $Z/W = (1 - dX/W - 3Y/W)/3$ into the curve equation gives the stated equation. Since $(Y + Z)/W = (1 - dX/W)/3$, we get $X/W = \theta/(d\theta + 3)$. □

For $Q = (X : Y : Z) \neq \mathcal{O}$ put $q(Q) = X^2/(YZ)$. By (5) and Lemma 4.25,

$$q = \frac{\theta^2(d\theta + 3)}{a\theta^3 + 1} = \frac{27k\,\xi^2}{(k - d\xi)^3 + 27a\,\xi^3}, \qquad \xi = 3x = \frac{kX}{W}. \tag{6}$$

Theorem 4.26. *Let $P \in H_{a,d}(K)$ have order N with $\gcd(N, 6) = 1$, let $\mathcal{G} = \langle P \rangle$, $q_i = q([i]P)$, and*

$$\Pi = \prod_{i=1}^{(N-1)/2} q_i, \qquad \Sigma = \sum_{i=1}^{(N-1)/2} q_i.$$

Then $\varphi(Q) = \big(\prod_{S\in\mathcal{G}} X(Q + S) : \prod_{S\in\mathcal{G}} Y(Q + S) : \prod_{S\in\mathcal{G}} Z(Q + S)\big)$ is an isogeny with kernel $\mathcal{G}$ onto $H_{a',d'}$, where

$$a' = a^N, \qquad d' = \frac{Nd - 6a\Sigma}{\Pi}.$$

Proof. Each factor depends on the representative of $Q+S$ only up to a common scalar, so φ is well defined, and it is invariant under $\mathcal{G}$. Write $u = \prod_S (X/Z)(Q+S)$ and $v = \prod_S (Y/Z)(Q+S)$; these are functions on $E' = H_{a,d}/\mathcal{G}$. Over $\bar{K}$, the line $Z=0$ meets $H_{a,d}$ in a coset $T_0 + \mathcal{K}$ of 3-torsion points, and since $3 \nmid N$ the divisors of u and v have poles exactly at the three points of $\varphi(T_0+\mathcal{K})$. Hence $1, u, v$ is a basis of $L(\varphi(T_0+\mathcal{K}))$, φ embeds E' as a plane cubic C', and the ten monomials of degree at most 3 in u, v satisfy a unique linear relation.

The translation by $(0 : -\zeta^2 : 1) \in \mathcal{K}$ is $(X : Y : Z) \mapsto (X : \zeta Y : \zeta^2 Z)$. It preserves $H_{a,d}$, maps $\mathcal{O}$ into $\mathcal{K}$ and has no fixed points. It multiplies u by ζ^{-2N} and v by ζ^{-N}, so the relation is homogeneous for the grading $\deg u = 2$, $\deg v = 1$ modulo 3. At the three points of $\varphi(\mathcal{K})$ we have $u = 0$ and $v \neq 0$, which excludes the classes $\{v, u^2, uv^2\}$ and $\{u, v^2, u^2 v\}$; so the relation is $\kappa_1 u^3 + \kappa_2 v^3 + \kappa_3 = \kappa_4 uv$. Negation sends (u, v) to $(u/v, 1/v)$, which forces $\kappa_2 = \kappa_3$. Therefore $C' = H_{a',d'}$ for $a' = \kappa_1/\kappa_2$, $d' = \kappa_4/\kappa_2$. Since $\prod_{S\in\mathcal{G}} (Y/Z)(S) = (Y/Z)(\mathcal{O}) \prod_{S\in\mathcal{G}\setminus\{\mathcal{O}\}} (Y/Z)(S) = -1$ (the factors for S and $-S$ are inverse), $\varphi(\mathcal{O}) = (0 : -1 : 1)$, so φ is an isogeny.

For a', let $c^3 = a$ and $Q_0 = (-1 : c : 0)$. The map $(X : Y : Z) \mapsto (Y/c : Z : cX)$ preserves $H_{a,d}$, sends $\mathcal{O}$ to Q_0, and has no fixed points on $H_{a,d}$ (a fixed point $(1 : \zeta^j c : \zeta^{2j} c)$ would force $3a = dc^2$, hence $d^3 = 27a$), so it is the translation by Q_0. Thus $(Y/X)(Q_0 + S) = cZ(S)/Y(S)$ and $\prod_{S\in\mathcal{G}} (Y/X)(Q_0+S) = c^N \prod_{S\in\mathcal{G}} (Z/Y)(S) = -c^N$. The point $\varphi(Q_0)$ has $Z' = 0$, so $a' X'^3 + Y'^3 = 0$ gives $a' = -(Y'/X')^3 = a^N$.

For d' we compare tangent lines at $\mathcal{O}$. Near $\mathcal{O}$ take $u_0 = X/Z$ as local parameter; from the curve equation, $v_0 = Y/Z = -1 - \frac{d}{3} u_0 + O(u_0^3)$. The invariant differential is $\omega = \mathrm{d}u_0/(3v_0^2 - d u_0) = \big(\frac{1}{3} + O(u_0)\big)\mathrm{d}u_0$, and one computes $\mathrm{d}\log(Y/Z)/\omega = d - 3a\,q$. By translation invariance, $\frac{\mathrm{d}}{\mathrm{d}u_0} \log(Y/Z)(Q+S)|_{Q=\mathcal{O}} = \frac{1}{3}(d - 3aq(S))$. Since $(X/Z)(S)\cdot(X/Z)(-S) = q(S)$ and $(Y/Z)(S)(Y/Z)(-S) = 1$,

$$u = \Pi\, u_0 + O(u_0^2), \qquad v = -1 - \Big(\frac{d}{3} + \sum_{S\neq\mathcal{O}} \frac{d - 3aq(S)}{3}\Big) u_0 + O(u_0^2) = -1 - \frac{Nd - 6a\Sigma}{3} u_0 + O(u_0^2).$$

The tangent line of $H_{a',d'}$ at $\mathcal{O}$ is $d'u + 3v + 3 = 0$, which gives $d'\Pi = Nd - 6a\Sigma$. □

Theorem 4.27 (Radical step on twisted Hessian curves)**.** *In the setting of Theorem 4.26, let $x_1 = x(P)$ in the coordinates of Lemma 4.25, compute the Tate data of $(E^H_{a,d}, P)$ with Lemmas 3.2 and 3.3, let r be the output of Theorem 2.2 and $x'_1 = x_1 + u_P^2 r$. Then the codomain $H_{a',d'}$ of φ contains a P-distinguished point P' of order N with*

$$x(P') = \frac{x'_1 - d^2/4}{\Pi^2} + \frac{d'^2}{4}, \qquad \theta(P') = \frac{9x(P')}{k' - 3d'x(P')}, \quad k' = d'^3 - 27a'.$$

Proof. The point P' of Theorem 2.2 has x-coordinate x'_1 on the Vélu codomain E'_V of $E^H_{a,d}$; this part of Proposition 3.4 holds for general Weierstrass equations by Lemma 2.1. The isogeny φ of Theorem 4.26, read in the coordinates of Lemma 4.25 on both sides, equals the Vélu isogeny followed by an isomorphism. The invariant differential of $E^H_{a,d}$ is $\mathrm{d}x/(2y + dx - k/3) = \mathrm{d}u_0/(3v_0^2 - d u_0)$, for every (a, d), and the expansion $u = \Pi u_0 + \dots$ shows that φ multiplies it by Π. The Vélu isogeny preserves it, so the isomorphism multiplies it by Π, i.e. it is $x' = x_V/\Pi^2 + \rho$ for some $\rho \in K$. Vélu's codomain has the same $b_2 = -3d^2$ as $E^H_{a,d}$, and $E^H_{a',d'}$ has $b_2 = -3d'^2$; the change of variables $x' = x_V/\Pi^2 + \rho$ maps b_2 to $b_2/\Pi^2 - 12\rho$, so $\rho = (d'^2 - d^2/\Pi^2)/4$. □

The radical step does not need the sum Σ of Theorem 4.26. The product Π fixes the scaling factor $1/\Pi$ between the Vélu codomain and $E^H_{a',d'}$, and then the invariants c_4 and c_6 fix d', since $a' = a^N$ is known.

Theorem 4.28. *In the setting of Theorem 4.27, let c_4 and c_6 be the invariants of the Vélu codomain E'_V of $E^H_{a,d}$, and put $a' = a^N$, $C_4 = c_4/\Pi^4$, $C_6 = c_6/\Pi^6$,*

$$C_6^+ = 157464a'^2 + C_6, \qquad C_6^- = 577368a'^2 - C_6,$$
$$\nu = 756a'C_4(C_6^+C_6^- + C_4^3), \qquad \varepsilon = (756a'C_6^+)^2 + C_4^3(C_6^+ + C_6^-).$$

Then $\varepsilon d' = \nu$. In particular $d' = \nu/\varepsilon$ if $\varepsilon \neq 0$.

Proof. By the b-invariants of Lemma 4.25, the curve $E^H_{a,d}$ has

$$c_4 = b_2^2 - 24b_4 = d^4 + 216ad, \qquad c_6 = -b_2^3 + 36b_2b_4 - 216b_6 = -d^6 + 540ad^3 + 5832a^2.$$

By the proof of Theorem 4.27, $E^H_{a',d'}$ is obtained from E'_V by $x' = x_V/\Pi^2 + \rho$, which divides c_4 by Π^4 and c_6 by Π^6. Hence d' is a common root of

$$f_4 = X^4 + 216a'X - C_4, \qquad f_6 = -X^6 + 540a'X^3 + 5832a'^2 - C_6.$$

We eliminate powers of X with polynomial combinations of f_4 and f_6. First $r_1 = f_6 + X^2 f_4 = 756a'X^3 - C_4X^2 + 5832a'^2 - C_6$, and $r_2 = 756a'f_4 - Xr_1 = C_4X^3 + C_6^+X - 756a'C_4$, without X^4. Then $r_3 = 756a'r_2 - C_4r_1 = C_4^2X^2 + 756a'C_6^+X - C_4C_6^-$ and $r_4 = Xr_3 - C_4r_2 = 756a'C_6^+X^2 - C_4(C_6^+ + C_6^-)X + 756a'C_4^2$ are quadratic, and

$$756a'C_6^+\, r_3 - C_4^2 r_4 = \varepsilon X - \nu.$$

All these polynomials lie in the ideal (f_4, f_6), so they vanish at $X = d'$. □

If $\varepsilon \neq 0$, Theorem 4.28 determines d' from Π alone. If $\varepsilon = 0$, for instance if f_4 and f_6 have more than one common root, the algorithm falls back to Theorem 4.26; this is a comparison and costs nothing in the generic case.

We work with $\xi = 3x = kX/W$, in which $F(\xi) = 4\xi^3 - 9d^2\xi^2 + 6kd\,\xi - k^2$ and $G(\xi) = 6\xi^2 - 9d^2\xi + 3kd$; as for Montgomery curves, this rescaling of x (a quadratic twist when 3 is not a square) does not affect x-only computations. The rescalings $(a', d') \mapsto (\tau^3 a', \tau d')$ used below are the isomorphisms $(X : Y : Z) \mapsto (X/\tau : Y : Z)$, which multiply x by τ^2; they avoid inversions.

Proposition 4.29. *Algorithm 11 is correct and costs $\frac{1}{2}(31N+121)+2\mathbf{E}_N$ $\mathbf{M}$ for $N \geq 9$: $3.5N$ $\mathbf{M}$ for the multiples, $7N$ $\mathbf{M}$ for Π_D and the radical core, and $5N$ $\mathbf{M}$ for the kernel product ($10\mathbf{M}$ per index). Pushing costs $N + \mathbf{E}_N + 1$ $\mathbf{M}$ once and $\frac{1}{2}(7N + 13)$ $\mathbf{M}$ per point. The codomain-only variant costs $10N + 2\mathbf{E}_N - 14$ $\mathbf{M}$ and needs no root, since it uses lines 1–4 without the rescaling in line 3, and $d' = (Nd\Pi_m - 6a\mathfrak{c}\,\Sigma_n)/d_d$ from Theorem 4.26 with the sums of line 8 ($13\mathbf{M}$ per index).*

Proof. Lines 1–4 compute the input of Algorithm 1: line 1 converts $\theta(P)$ to the coordinate ξ of the Weierstrass model, and lines 2–4 are Lemmas 3.2 and 3.3 for $E^H_{a,d}$; in line 3, $\Delta_2 = -Z_1^4\psi_3(\xi_1)$ with $\psi_3(\xi) = 3\xi^4 - 9d^2\xi^3 + 9kd\xi^2 - 3k^2\xi$, and it reuses X_1^2, $d^2X_1Z_1$, kdZ_1^2 and $k^2Z_1^3$ from line 2. As for short Weierstrass curves, the second half of line 3 replaces Z_1 by F_nZ_1, so that $D_2 = 1$; it also updates X_1^2 and kZ_1, which are reused below. Line 5 is the radical core, with Π_Δ^2 by Lemma 3.6(1); the product Π_D, which the core does not need, is computed directly for line 6 and for pushing. Line 6 computes $\Pi = \mathfrak{c}^h\Pi_n/\Pi_m$ by (6): $q_i = \mathfrak{c}n_i/m_i$ with $n_i = \tilde{X}_i^2 D_i$, and $\prod_i n_i = (\prod_i \tilde{X}_i)^2\Pi_D$ since $D_1 = 1$.

Line 7 is Theorem 4.28 after the isomorphism $(X : Y : Z) \mapsto (X/\tau : Y : Z)$ from $H_{a',d'}$ to $H_{\tau^3 a',\tau d'}$ with $\tau = d_dT$, which multiplies c_4 by τ^4 and c_6 by τ^6. Indeed, by Proposition 3.5 the Vélu codomain is the curve of the scaled cubic $x^3 + \bar{a}x^2 + \bar{b}x + \bar{c}$ with x divided by $\vartheta/\Omega = T$, where $\vartheta = 2D_2F_n^2\mathcal{U}^2$. In the coordinate ξ, its invariants are therefore $16(\bar{a}^2 - 3\bar{b})/T^2$ and

Algorithm 11 Radical step on twisted Hessian curves

Input: $a, d \in K$, $(\Theta : \Xi) = \theta(P)$ with P of order N, $\gcd(N, 6) = 1$; points $\theta(R_j) = (\Theta_R : \Xi_R)$
Output: a', d' with $H_{a',d'} \cong H_{a,d}/\langle P \rangle$, $\theta(P')$, and the images $\theta(\varphi(R_j))$

1:	$k \leftarrow d^3 - 27a$, $(X_1 : Z_1) \leftarrow (k\Theta : d\Theta + 3\Xi)$	4
2:	$F_n \leftarrow X_1(4X_1^2 - 9d^2X_1Z_1 + 6kdZ_1^2) - k^2Z_1^3$, $G_n \leftarrow \frac{1}{2}(6X_1^2 - 9d^2X_1Z_1 + 3kdZ_1^2)$, with $k^2Z_1^3 = (kZ_1)^2Z_1$	10
3:	$\Delta_2 \leftarrow 3\big(X_1 \cdot k^2Z_1^3 - X_1^2(X_1^2 - 3d^2X_1Z_1 + 3kdZ_1^2)\big)$, $(\Delta_1, D_1) \leftarrow (0, 1)$	2
	$(X_1, Z_1, G_n, F_n) \leftarrow (F_nX_1, F_nZ_1, F_n^2G_n, F_n^4)$, $D_2 \leftarrow 1$, and X_1^2, kZ_1 accordingly	7
4:	(Δ_i, D_i) for $3 \le i \le h$ as in line 3 of Algorithm 3	$\frac{1}{2}(7N - 49)$
5:	$\Pi_D \leftarrow \prod_{i=2}^{h} D_i$; Algorithm 1 on $(F_n, G_n, 1, \Delta_h^2 D_{h-1} D_h, (\Delta_i, D_i))$ (Lemma 3.6(1))	$7N + 31$
6:	$\tilde{X}_i \leftarrow X_1D_i + \Delta_i$, $m_i \leftarrow (kZ_1D_i - d\tilde{X}_i)^3 + 27a\tilde{X}_i^3$ for $1 \le i \le h$; $\Pi_m \leftarrow \prod_i m_i$, $\Pi_n \leftarrow \big(\prod_i \tilde{X}_i\big)^2 \Pi_D$	$5N - 10$
7:	$\mathfrak{c} \leftarrow 27kZ_1$, $d_d \leftarrow \mathfrak{c}^h \Pi_n$, $T \leftarrow \mathcal{R}_d g Z_1 F_n$, $a_\tau \leftarrow a^N (d_d T)^3$	$2\mathbf{E}_N + 6$
	$C_4 \leftarrow \frac{16}{9}(\bar{a}^2 - 3\bar{b})\Pi_m^4 T^2$, $C_6 \leftarrow -\frac{32}{27}\big(\bar{a}(2\bar{a}^2 - 9\bar{b}) + 27\bar{c}\big)\Pi_m^6 T^3$	11
	$C_6^+, C_6^- \leftarrow 157464a_\tau^2 + C_6,\ 577368a_\tau^2 - C_6$, $\nu \leftarrow 756a_\tau C_4(C_6^+C_6^- + C_4^3)$, $\varepsilon \leftarrow (756a_\tau C_6^+)^2 + C_4^3(C_6^+ + C_6^-)$	9
8:	**if** $\varepsilon = 0$: $\Sigma_n \leftarrow \sum_i \tilde{X}_i^2 D_i \prod_{j \ne i} m_j$, computed as $\Sigma_n \leftarrow \Sigma_n m_i + \Pi_{m,i-1} \tilde{X}_i^2 D_i$; $a' \leftarrow a^N d_d^3$, $d' \leftarrow Nd\Pi_m - 6a\mathfrak{c}\,\Sigma_n$, $t_0 \leftarrow \Pi_m^2$ (Theorem 4.26)	$\frac{1}{2}(13N - 23) + \mathbf{E}_N$
	else: $a' \leftarrow a_\tau \varepsilon^3$, $d' \leftarrow \nu$, $t_0 \leftarrow \Pi_m^2 (T\varepsilon)^2$	6
9:	$X_V \leftarrow X_1\mathcal{R}_d + F_n\mathcal{R}$, $Z_V \leftarrow Z_1\mathcal{R}_d$	2
10:	$t_1 \leftarrow t_0(4X_V - 3d^2Z_V) + 3d'^2Z_V$, $t_2 \leftarrow 4Z_V$	4
11:	$k' \leftarrow d'^3 - 27a'$, $\theta(P') \leftarrow (3t_1 : k't_2 - d't_1)$	3
12:	**Pushing, once:** $v_1 \leftarrow 3kdZ_1$, $v_2 \leftarrow -k^2Z_1$, $v_3 \leftarrow t_0 Z_1^{N-1} \Pi_D^3$, $v_4 \leftarrow v_2Z_1$; $\tilde{X}_iD_i$ and D_i^2 for $1 \le i \le h$	$N + \mathbf{E}_N + 1$
13:	**Pushing, for each** R_j**:** $(X_R, Z_R) \leftarrow (k\Theta_R, d\Theta_R + 3\Xi_R)$, $w_0 \leftarrow X_R^2$, $w_1 \leftarrow X_RZ_R$, $w_2 \leftarrow -(v_1w_1 + v_2Z_R^2)$, $w_3 \leftarrow -v_4w_1$, $w_4 \leftarrow X_RZ_1$; $X \leftarrow v_3X_R \prod_i (w_0\tilde{X}_i^2 + w_2\tilde{X}_iD_i + w_3D_i^2)$, with $\tilde{X}_i^2$ from line 6, $Z \leftarrow Z_R\Pi_n\big(\prod_i (w_4D_i - Z_R\tilde{X}_i)\big)^2$; $\theta(\varphi(R_j)) \leftarrow (3X : k'Z - d'X)$	$\frac{1}{2}(7N + 13)$ each

$-32(2\bar{a}^3 - 9\bar{a}\bar{b} + 27\bar{c})/T^3$, and in the coordinate $x = \xi/3$ they are 9 and 27 times smaller. With $\Pi = d_d/\Pi_m$ we get $\tau^4 c_4/\Pi^4 = C_4$ and $\tau^6 c_6/\Pi^6 = C_6$ as in line 7, and $\tau^3 a^N = a_\tau$. Hence $\nu/\varepsilon = \tau d'$, and the output $(a_\tau \varepsilon^3, \nu)$ is $H_{a^N,d'}$ scaled by $\tau\varepsilon$. Line 8 is the fallback of Theorem 4.26, with the scaling by d_d instead: by (6), $\Sigma = \mathfrak{c}\,\Sigma_n/\Pi_m$ and $\Pi = d_d/\Pi_m$, so $d' = (Nd\Pi_m - 6a\mathfrak{c}\,\Sigma_n)/d_d$. Its count is that of recomputing the kernel data of line 6 together with the sums, as in the codomain-only variant; this branch is only taken if $\varepsilon = 0$.

Lines 9–11 compute the next generator. By Proposition 3.5, line 9 gives the ξ-coordinate X_V/Z_V of P' on the Vélu codomain. Lines 10–11 are Theorem 4.27 after the scaling by $\tau\varepsilon$, which multiplies x by $(\tau\varepsilon)^2$: the term $(x_1' - d^2/4)/\Pi^2$ becomes $(x_1' - d^2/4)\Pi_m^2 T^2 \varepsilon^2 = (x_1' - d^2/4)t_0$, and $d'^2/4$ becomes $\nu^2/4$. Lines 12–13 push points by Proposition 4.30, scaled in the same way. For the counts, X_1^2 and kZ_1 from line 3 are reused for $i = 1$ in line 6 and in lines 7 and 12, line 13 reuses the squares $\tilde{X}_i^2$ of line 6, and d'^2 from line 10 in k'; line 7 also reuses $\bar{a}^2$ and Π_m^2. □

Proposition 4.30 (Pushing points)**.** *For $Q \in H_{a,d}$ with $x = x(Q)$ and $x_i = x([i]P)$,*

$$x(\varphi(Q)) = \frac{x}{\Pi^2} \prod_{i=1}^{(N-1)/2} \frac{x^2 x_i^2 - b_4 x x_i - b_6(x + x_i)}{x_i^2 (x - x_i)^2},$$

*which costs $\frac{1}{2}(7N+13)$ **M** per point after a shared precomputation of $N+\mathbf{E}_N+1$ **M** (lines 12–13 of Algorithm 11).*

Proof. The function x has divisor $(K_1)+(K_2)-2(\mathcal{O})$ with $\{K_1, K_2\} = \mathcal{K}\setminus\{\mathcal{O}\}$, and $\varphi(\mathcal{K}) = \mathcal{K}'$. Consequently, $x' \circ \varphi$ and $\prod_{S\in\mathcal{G}} x(Q+S)$ have the same divisor, and comparing the leading terms at $\mathcal{O}$ (where $x' \circ \varphi \sim x/\Pi^2$) fixes the constant. The product of $x(Q+S)x(Q-S)$ is the classical formula $(x - x_i)^2 x(Q+S)x(Q-S) = x^2 x_i^2 - b_4 x x_i - b_6(x + x_i) - b_8$ with $b_8 = 0$. With $x = X_R/Z_R$ and $x_i = \tilde{X}_i/(Z_1 D_i)$, the numerator of the ith factor is a quadratic form in (X_R, Z_R) whose coefficients are linear in $\tilde{X}_i^2$, $\tilde{X}_i D_i$ and D_i^2: it is $w_0\tilde{X}_i^2 + w_2\tilde{X}_i D_i + w_3 D_i^2$ in the notation of line 13, where b_4, b_6 and the powers of Z_1 are absorbed into $v_1, \ldots, v_4$. Its denominator is the square of $w_4 D_i - Z_R\tilde{X}_i$, and the squaring is done once, on the product. With $\tilde{X}_i D_i$ and D_i^2 computed once, each index costs $3\mathbf{M}$ for the numerator, $2\mathbf{M}$ for the linear factor and $2\mathbf{M}$ to accumulate. For $i \le 2$ we have $D_i = 1$, so the precomputation and the products by D_i and D_i^2 are free, and the first index needs no accumulation. □

The twisted Hessian step remains more expensive than the others because its normalization involves the product Π of a cubic rational function over the kernel, rather than a product of linear factors. Theorem 4.28 removes the sum Σ, which also appears in the formulae of [35], from the radical step, and saves $3\mathbf{M}$ per index: the numerators n_i and the accumulation of Σ. In the codomain-only variant, where Algorithm 1 and hence the invariants of the Vélu codomain are not available, Σ remains the cheaper option.

5 Implementation and verification

We provide a Magma [6] implementation of all algorithms of Sections 3 and 4, with one file per model. It uses the names of the pseudocode and marks line k of an algorithm with a comment `// Line k`, so that every line can be compared with the corresponding algorithm; temporary values are named $t_0, t_1, \ldots$ as in the pseudocode. For each model, a short script runs three consecutive radical steps of degree $N \in \{5, 7, 9, 11, 13, 15\}$, or $N \in \{5, 7, 11, 13, 25\}$ for twisted Hessian curves, starting from a random curve in the given model with a point P of order N; this includes composite degrees. It works over a prime field $\mathbb{F}_p$ of 60 bits with $\gcd(N, p-1) = 1$, so that Nth roots are unique. Each step is compared with Magma's implementation of Vélu's formulae in three ways. First, the computed codomain must be isomorphic over $\mathbb{F}_p$ to $E/\langle P\rangle$,

which is stronger than having the same j-invariant. Second, the computed $x(P')$ must belong to a rational point of order N that the dual isogeny maps to $\pm P$. Third, the pushed points must be the images of random points under Vélu's isogeny followed by this isomorphism. The codomain-only variants are checked in the same way. Since the projective parameters $(a : d)$ of a twisted Edwards curve only determine it up to a quadratic twist, which y-only arithmetic does not see, the Edwards script compares on the Montgomery x-line. The script for Jacobi intersections converts every point with Lemma 4.23 and checks it on J_a itself, and the script for Kummer lines also uses the doubling and differential addition of [24], to check the orders of the kernel generator and the next generator and that pushing commutes with doubling. A separate script runs the fallbacks for $\bar{a} = 0$ in Algorithm 3 and for $\delta = 0$ in Algorithms 4 and 6 on curves that trigger them. For $N = 5$ the pair sums for Σ_3 in line 5 of Algorithm 1 are empty and $\Sigma_3 = \Lambda e_h$; the code treats this case separately. The scripts are meant as sanity checks and are kept short. The code is available [13].

Statements and Declarations

Funding. Thomas Decru is funded by Fonds Wetenschappelijk Onderzoek – Vlaanderen (FWO) with reference number 1245025N, by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement ISOCRYPT – No. 101020788), by the Research Council KU Leuven grant C14/24/099 and by CyberSecurity Research Flanders with reference number VOEWICS02.

Competing interests. The author has no competing interests to declare that are relevant to the content of this article.

Code availability. The Magma implementation of all algorithms, together with the scripts of Section 5, is available at `https://github.com/TDecru/Radical_models`.

Data availability. No datasets were generated or analysed in this study.

Use of large language models. A large language model (Claude, Anthropic) was used to help optimize parts of the algorithms and to assist with the implementation and testing of the code. The author has verified all results and takes full responsibility for the content of this article.

References

[1] Daniel J. Bernstein, Peter Birkner, Marc Joye, Tanja Lange, and Christiane Peters. Twisted Edwards curves. In *Progress in Cryptology – AFRICACRYPT 2008*, volume 5023 of *Lecture Notes in Computer Science*, pages 389–405. Springer, 2008. `https://doi.org/10.1007/978-3-540-68164-9_26`.

[2] Daniel J. Bernstein, Chitchanok Chuengsatiansup, David Kohel, and Tanja Lange. Twisted Hessian curves. In *Progress in Cryptology – LATINCRYPT 2015*, volume 9230 of *Lecture Notes in Computer Science*, pages 269–294. Springer, 2015. `https://doi.org/10.1007/978-3-319-22174-8_15`.

[3] Daniel J. Bernstein, Luca De Feo, Antonin Leroux, and Benjamin Smith. Faster computation of isogenies of large prime degree. In *ANTS XIV: Proceedings of the Fourteenth Algorithmic Number Theory Symposium*, volume 4 of *The Open Book Series*, pages 39–55. Mathematical Sciences Publishers, 2020. `https://doi.org/10.2140/obs.2020.4.39`.

[4] Daniel J. Bernstein and Tanja Lange. Faster addition and doubling on elliptic curves. In *Advances in Cryptology – ASIACRYPT 2007*, volume 4833 of *Lecture Notes in Computer Science*, pages 29–50. Springer, 2007. https://doi.org/10.1007/978-3-540-76900-2_3.

[5] Olivier Billet and Marc Joye. The Jacobi model of an elliptic curve and side-channel analysis. In *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes – AAECC-15*, volume 2643 of *Lecture Notes in Computer Science*, pages 34–42. Springer, 2003. https://doi.org/10.1007/3-540-44828-4_5.

[6] Wieb Bosma, John Cannon, and Catherine Playoust. The Magma algebra system. I. The user language. *Journal of Symbolic Computation*, 24(3–4):235–265, 1997. https://doi.org/10.1006/jsco.1996.0125.

[7] Wouter Castryck and Thomas Decru. CSIDH on the surface. In *Post-Quantum Cryptography – PQCrypto 2020*, volume 12100 of *Lecture Notes in Computer Science*, pages 111–129. Springer, 2020. https://doi.org/10.1007/978-3-030-44223-1_7.

[8] Wouter Castryck, Thomas Decru, Marc Houben, and Frederik Vercauteren. Horizontal racewalking using radical isogenies. In *Advances in Cryptology – ASIACRYPT 2022, Part II*, volume 13792 of *Lecture Notes in Computer Science*, pages 67–96. Springer, 2022. https://doi.org/10.1007/978-3-031-22966-4_3.

[9] Wouter Castryck, Thomas Decru, and Frederik Vercauteren. Radical isogenies. In *Advances in Cryptology – ASIACRYPT 2020, Part II*, volume 12492 of *Lecture Notes in Computer Science*, pages 493–519. Springer, 2020. https://doi.org/10.1007/978-3-030-64834-3_17.

[10] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: An efficient post-quantum commutative group action. In *Advances in Cryptology – ASIACRYPT 2018, Part III*, volume 11274 of *Lecture Notes in Computer Science*, pages 395–427. Springer, 2018. https://doi.org/10.1007/978-3-030-03332-3_15.

[11] Jesús-Javier Chi-Domínguez and Krijn Reijnders. Fully projective radical isogenies in constant-time. In *Topics in Cryptology – CT-RSA 2022*, volume 13161 of *Lecture Notes in Computer Science*, pages 73–95. Springer, 2022. https://doi.org/10.1007/978-3-030-95312-6_4.

[12] David V. Chudnovsky and Gregory V. Chudnovsky. Sequences of numbers generated by addition in formal groups and new primality and factorization tests. *Advances in Applied Mathematics*, 7(4):385–434, 1986. https://doi.org/10.1016/0196-8858(86)90023-0.

[13] Radical isogenies on every model: Magma code. https://github.com/TDecru/Radical_models, 2026.

[14] Craig Costello and Huseyin Hisil. A simple and compact algorithm for SIDH with arbitrary degree isogenies. In *Advances in Cryptology – ASIACRYPT 2017, Part II*, volume 10625 of *Lecture Notes in Computer Science*, pages 303–329. Springer, 2017. https://doi.org/10.1007/978-3-319-70697-9_11.

[15] Thinh Dang and Dustin Moody. Twisted Hessian isogenies. Cryptology ePrint Archive, Paper 2019/1003, 2019. https://eprint.iacr.org/2019/1003.

[16] Thomas Decru. Radical $\sqrt[n]{\text{élu}}$ isogeny formulae. In *Advances in Cryptology – CRYPTO 2024, Part V*, volume 14924 of *Lecture Notes in Computer Science*, pages 107–128. Springer, 2024. https://doi.org/10.1007/978-3-031-68388-6_5.

[17] Thomas Decru. Getting to the right root: Radical isogenies of every degree. Cryptology ePrint Archive, Paper 2026/2595, 2026. https://eprint.iacr.org/2026/2595.

[18] Łukasz Dzierzkowski and Michał Wroński. How to compute an isogeny on the extended Jacobi quartic curves? *International Journal of Electronics and Telecommunications*, 68(3):463–468, 2022. https://doi.org/10.24425/ijet.2022.139890.

[19] Harold M. Edwards. A normal form for elliptic curves. *Bulletin of the American Mathematical Society*, 44(3):393–422, 2007. https://doi.org/10.1090/S0273-0979-07-01153-6.

[20] Pierrick Gaudry and David Lubicz. The arithmetic of characteristic 2 Kummer surfaces and of elliptic Kummer lines. *Finite Fields and Their Applications*, 15(2):246–260, 2009. https://doi.org/10.1016/j.ffa.2008.12.006.

[21] Huseyin Hisil, Kenneth Koon-Ho Wong, Gary Carter, and Ed Dawson. Jacobi quartic curves revisited. In *Information Security and Privacy – ACISP 2009*, volume 5594 of *Lecture Notes in Computer Science*, pages 452–468. Springer, 2009. https://doi.org/10.1007/978-3-642-02620-1_31.

[22] Marc Joye and Jean-Jacques Quisquater. Hessian elliptic curves and side-channel attacks. In *Cryptographic Hardware and Embedded Systems – CHES 2001*, volume 2162 of *Lecture Notes in Computer Science*, pages 402–410. Springer, 2001. https://doi.org/10.1007/3-540-44709-1_33.

[23] Marc Joye, Mehdi Tibouchi, and Damien Vergnaud. Huff's model for elliptic curves. In *Algorithmic Number Theory – ANTS-IX*, volume 6197 of *Lecture Notes in Computer Science*, pages 234–250. Springer, 2010. https://doi.org/10.1007/978-3-642-14518-6_20.

[24] Sabyasachi Karati and Palash Sarkar. Kummer for genus one over prime order fields. In *Advances in Cryptology – ASIACRYPT 2017, Part II*, volume 10625 of *Lecture Notes in Computer Science*, pages 3–32. Springer, 2017. https://doi.org/10.1007/978-3-319-70697-9_1.

[25] Suhri Kim. Complete analysis of implementing isogeny-based cryptography using Huff form of elliptic curves. *IEEE Access*, 9:154500–154512, 2021. https://doi.org/10.1109/ACCESS.2021.3128515.

[26] Suhri Kim, Kisoon Yoon, Young-Ho Park, and Seokhie Hong. Optimized method for computing odd-degree isogenies on Edwards curves. In *Advances in Cryptology – ASIACRYPT 2019, Part II*, volume 11922 of *Lecture Notes in Computer Science*, pages 273–292. Springer, 2019. https://doi.org/10.1007/978-3-030-34621-8_10.

[27] David Kohel. *Endomorphism Rings of Elliptic Curves over Finite Fields*. PhD thesis, University of California, Berkeley, 1996. https://www.i2m.univ-amu.fr/perso/david.kohel/pub/thesis.pdf.

[28] Pierre-Yvan Liardet and Nigel P. Smart. Preventing SPA/DPA in ECC systems using the Jacobi form. In *Cryptographic Hardware and Embedded Systems – CHES 2001*, volume 2162 of *Lecture Notes in Computer Science*, pages 391–401. Springer, 2001. https://doi.org/10.1007/3-540-44709-1_32.

[29] Michael Meyer and Steffen Reith. A faster way to the CSIDH. In *Progress in Cryptology – INDOCRYPT 2018*, volume 11356 of *Lecture Notes in Computer Science*, pages 137–152. Springer, 2018. https://doi.org/10.1007/978-3-030-05378-9_8.

[30] Peter L. Montgomery. Speeding the Pollard and elliptic curve methods of factorization. *Mathematics of Computation*, 48(177):243–264, 1987. https://doi.org/10.1090/S0025-5718-1987-0866113-7.

[31] Dustin Moody and Daniel Shumow. Analogues of Vélu's formulas for isogenies on alternate models of elliptic curves. *Mathematics of Computation*, 85(300):1929–1951, 2016. https://doi.org/10.1090/mcom/3036.

[32] Tomoki Moriya, Hiroshi Onuki, and Tsuyoshi Takagi. How to construct CSIDH on Edwards curves. In *Topics in Cryptology – CT-RSA 2020*, volume 12006 of *Lecture Notes in Computer Science*, pages 512–537. Springer, 2020. https://doi.org/10.1007/978-3-030-40186-3_22.

[33] Katsuyuki Okeya, Hiroyuki Kurumatani, and Kouichi Sakurai. Elliptic curves with the Montgomery-form and their cryptographic applications. In *Public Key Cryptography – PKC 2000*, volume 1751 of *Lecture Notes in Computer Science*, pages 238–257. Springer, 2000. https://doi.org/10.1007/978-3-540-46588-1_17.

[34] Hiroshi Onuki and Tomoki Moriya. Radical isogenies on Montgomery curves. In *Public-Key Cryptography – PKC 2022, Part I*, volume 13177 of *Lecture Notes in Computer Science*, pages 473–497. Springer, 2022. https://doi.org/10.1007/978-3-030-97121-2_17.

[35] Fouazou Lontouo Perez Broon, Thinh Dang, Emmanuel Fouotsa, and Dustin Moody. Isogenies on twisted Hessian curves. *Journal of Mathematical Cryptology*, 15(1):345–358, 2021. https://doi.org/10.1515/jmc-2020-0037.

[36] Joost Renes. Computing isogenies between Montgomery curves using the action of $(0, 0)$. In *Post-Quantum Cryptography – PQCrypto 2018*, volume 10786 of *Lecture Notes in Computer Science*, pages 229–247. Springer, 2018. https://doi.org/10.1007/978-3-319-79063-3_11.

[37] Damien Robert and Nicolas Sarkis. Computing 2-isogenies between Kummer lines. *IACR Communications in Cryptology*, 1(1):Article 26, 2024. https://doi.org/10.62056/abvua69p1.

[38] Joseph H. Silverman. *The Arithmetic of Elliptic Curves*, volume 106 of *Graduate Texts in Mathematics*. Springer, second edition, 2009. https://doi.org/10.1007/978-0-387-09494-6.

[39] Nigel P. Smart. The Hessian form of an elliptic curve. In *Cryptographic Hardware and Embedded Systems – CHES 2001*, volume 2162 of *Lecture Notes in Computer Science*, pages 118–125. Springer, 2001. https://doi.org/10.1007/3-540-44709-1_11.

[40] Jacques Vélu. Isogénies entre courbes elliptiques. *Comptes Rendus de l'Académie des Sciences de Paris, Série A*, 273:A238–A241, 1971.

[41] Michał Wroński. Application of Velusqrt algorithm to Huff's curves. *Publicationes Mathematicae Debrecen*, 100(Supplementum):639–653, 2022. https://doi.org/10.5486/PMD.2022.suppl.6.