Fast pairings via biextensions and cubical arithmetic

Preliminary version for April 1st

DAMIEN ROBERT

Abstract. Biextensions associated to line bundles on abelian varieties allows to reinterpret the usual Weil, Tate, Ate, optimal Ate, …, pairings as monodromy pairings. We introduce a cubical arithmetic, derived from the canonical cubical torsor structure of these line bundles, to obtain an efficient arithmetic of these biextensions.

This unifies and extends Miller's standard algorithm to compute pairings along with other algorithms like elliptic nets and theta functions, and allows to adapt these algorithms to pairings on any model of abelian varieties with a polarisation Φ_D, as long as we have an explicit theorem of the square for D.

In particular, we give explicit formulas for the arithmetic of the biextension (and cubical torsor structure) associated to the divisor $D = 2(0_E)$ on an elliptic curve. We derive very efficient pairing formulas on certain models of elliptic curves and Kummer lines. Notably for generic pairings on Montgomery curves, our cubical biextension ladder algorithm to compute pairings costs only $15M$ by bits, which as far as I know is faster than any pairing doubling formula in the literature.

1. Introduction

Pairing based cryptography has been thoroughly optimised over the years, and the pairings are set up via parameters and subgroups tailored for speed. For instance the Tate pairing is restricted to $\mathbb{G}_1 \times \mathbb{G}_2$ and the embedding degree d is often chosen even to benefit from denominator elimination, we have tools for Miller loop reduction like the Ate and optimal Ate pairing on $\mathbb{G}_2 \times \mathbb{G}_1$, and so on.

Nevertheless, pairings are important in other aspects than pairing based cryptography, in which case we need to compute "generic pairings". In particular, for "generic pairings", we cannot assume that denominator elimination is available, nor that our points $P, Q \in E[\ell]$ are in specific eigenspaces of the Frobenius.

This is notably the case in isogeny based cryptography, where pairings are an important tool. They are used to speed up smooth order DLPs on elliptic curves, generate canonical basis, test the degree of an isogeny, compress messages and signatures, and so on. In these examples the points P, Q are arbitrary points of ℓ-torsion. We refer to [Rei23] for other examples.

By contrast to pairing based cryptography, pairings are quite slow in the generic case, usually much slower than a curve scalar multiplication. The best formulas I have been able to find in the literature, in [BELL10], uses $10M + 9S$ for doubling, and $11.5M + 3S$ by addition.

1.1. Efficient generic pairings on Kummer lines.

In this paper, we introduce a novel algorithm, that is much faster for generic pairings, and is potentially interesting even in the context of pairing based cryptography. We first give a general framework for any elliptic curves and even abelian varieties (not necessarily principally polarised).

Date: July 3, 2024.

Then we focus on the particular case of the Montgomery model. Montgomery elliptic curves E, or more precisely their associated Kummer lines $E/\pm 1$, have a very efficient scalar multiplication in the form of the Montgomery ladder, which costs $5M + 4S + 1m_0$ by bits. Here we denote by M a multiplication on the base field, S a square, and m_0 a multiplication by a curve constant (which in the case of a Montgomery curve $E : By^2 = x^3 + \mathcal{A}x^2 + 1$ will be the multiplication by $(\mathcal{A} + 2)/4$). We recall that if $P = (x(P), y(P)) = (X(P) : Y(P) : Z(P))$ in affine (resp. projective) Weierstrass coordinates, its representation on the Kummer line is $x(P)$ (resp. $(X(P) : Z(P))$).

Due to their fast arithmetic (Mongomery curves are also birationally equivalent to twisted Edwards curves, and Curve25519 is also a Montgomery curve), the Montgomery model is usually used in isogeny based cryptography, which makes it a natural target for generic pairing formulas.

For the Montgomery model, our pairing framework gives:

Theorem 1.1. *Let $E : By^2 = x^3 + \mathcal{A}x^2 + 1$ be an elliptic curve in Montgomery form over a finite field $\mathbb{F}_q$. Let ℓ be an integer such that $\mu_\ell \subset \mathbb{F}_q$, $P \in E[\ell](\mathbb{F}_q), Q \in E(\mathbb{F}_q)$. Let $\kappa = 2$ if ℓ is odd, and $\kappa = 1$ if ℓ is even.*

Assume that we are given $\frac{\mathcal{A}+2}{4}$, and the coordinates $x(P), x(Q), x(P+Q)$ and their inverses. Then one can compute a projective representation of the non reduced Tate pairing (i.e., its numerator and denominator) to the power κ, $e_{T,\ell}(P,Q)^\kappa$, via a cubical biextension ladder which costs $8M + 6S + 1m_0$ by bits.

A similar algorithm holds for the Weil pairing $e_{W,\ell}(P,Q)^\kappa$ when $P, Q \in E[\ell]$, using two cubical biextension ladders rather than one; and also for the the Ate and optimal Ate pairings (to the power κ).

Special cases include:

- *When $\ell = 2^m$, or for self pairings when $P = Q$, the cubical biextension ladder costs $5M + 4S + 1m_0$ by bits.*
- *For batch pairing computations $e_{T,\ell}(P, Q_i)$, with the same base point P, after the first pairing the following ones cost $3M + 2S$ by bits for the ladder.*

We remark that our cubical biextension ladder cost is much more in line with the cost of the Montgomery Kummer line ladder for scalar multiplication, and that it costs less than the doubling formulas of [BELL10] (which needs to compute additions too!).

An implementation in Sage of Theorem 1.1 (along with many other algorithms) is available at [Rob23b].

Remark 1.2.

- Theorem 1.1 is proven in Section 5.2. We also have a variant which uses a more standard double and add algorithm to compute a biextension exponentiation, where each doubling costs $5M + 4S + 1m_0$, but with much more expensive additions of $32M + 4S + 2m_0$ (there is probably still some room for optimisation for the addition, as evidenced by the fact that I also have a DoubleAndAdd formula which costs only $17M + 8S + 3m_0 \ldots$). This only makes the double and add approach worthwhile compared to a ladder when using a large enough window (or for ℓ of low Hamming weight).
- If $P, Q, P + Q$ are given by their projective $(X(P) : Z(P))$ coordinates on the Kummer line, computing $x(P) = X(P)/Z(P), x(Q) = X(Q)/Z(Q), x(P + Q) = X(P + Q)/Z(P + Q)$ and their inverses only requires one inversion and several multiplications using Montgomery's batch inversion algorithm.

- From $x(P), x(Q), x(P+Q)$, one may only recover P, Q up to a common sign. This involves a square root computation to compute (P, Q) and $(-P, -Q)$. By bilinearity, $e(P, Q) = e(-P, -Q)$, which explains why Theorem 1.1 only needs to know $P, Q, P+Q$ on the Kummer line.
- If we are only given $x(P), x(Q)$ but not $x(P+Q)$, we can only recover the symmetrised pairings $e(P, Q) + e(P, Q)^{-1}$. This involves doing a cubical biextension ladder over the degree two algebra $\mathbb{F}_q[X]/((X - x(P+Q))(X - x(P-Q)))$. This is standard, see [GL08] for elliptic curves, [LR16] for abelian varieties, and also Section 4.7.
- If E is given by a twisted Edwards model, there is a birational map to a Montgomery curve (which by [CGF08] is particularly simple at the level of the Kummer lines: $(Y : Z) \mapsto (Y + Z : Z - Y)$), so we can apply Theorem 1.1.
- During the execution of the standard Miller algorithm for pairings, intermediate zeroes and poles are introduced, which can result in undefined values. Standard solutions are to switch the evaluation point (this does not pose a problem in pairing based cryptography where we have a lot of points, but can be a problem in number theoretic applications when me might have none), or to use Taylor series expansion along a uniformiser (see for instance [Rob21a, Lemma 3.5.3]). By contrast, our cubical biextension ladder is complete, i.e., is always defined, as long as $P, Q, P+Q \neq (0 : 1)$ (see Remark 5.3).
- We have variants of Theorem 1.1 for different models of Kummer lines. For a level 2 theta model, the complexity of the cubical biextension ladder is $7M + 7S + 2m_0$ by bits (there is a variant with a tradeoff of $1S + 2m_0 - 1M$), see Section 5.3. For a short Weierstrass model, however, the complexity is much worse, at $15M + 8S + 6m_0$ by bits, see Section 5.4.

1.2. **Overview of the algorithm: the practical point of view.** The algorithm proceeds as follows: given the x-only coordinates $x(P), x(Q), x(P+Q)$, we can use an extended Montgomery ladder to compute $x(mP), x(mP+Q)$ for any $m > 0$; this costs one doubling and two differential additions by bits.

In practice, to prevent a division at each step, we want to work with projective coordinates $(X(mP) : Z(mP))$ rather than $x(mP)$. Now in the computer, these projective coordinates are represented by affine coordinates, $X(mP), Z(mP)$, and so define a "affine point" $\widetilde{mP} = (X(mP), Z(mP))$ lying "above" P. We will see in Section 4.5 that $\widetilde{mP}$ is what we call a cubical point, and the coordinates $(X(mP), Z(mP))$ is the affine lift representation of cubical points. And the interesting thing is that there is a well defined cubical arithmetic, which lift the projective arithmetic coming from the addition law on E.

So in our pairing algorithm, we compute cubical points $\widetilde{mP}, \widetilde{mP+Q}$, using variants of the standard doubling and differential additions which are carefully tailored to give the cubical arithmetic. We will call these variants the cubical or affine doubling and differential additions, and the resulting ladder the cubical ladder or affine ladder.

Now we start with $\widetilde{P} = (x(P), 1)$, $\widetilde{Q} = (x(Q), 1)$, $\widetilde{P+Q} = (x(P+Q), 1)$, and we can use our cubical ladder to compute $\widetilde{\ell P + Q}, \widetilde{\ell P}$. Since the cubical arithmetic lift the elliptic curve arithmetic, and $\ell P = 0_E$, the point $\widetilde{\ell P + Q}$ differs from $\widetilde{Q}$ by some projective factor $\lambda_{P,1}$: $\widetilde{\ell P + Q} = \lambda_{1,P}\widetilde{Q}$, and since $\widetilde{Q}$ is normalised we have $\lambda_{1,P} = Z(\widetilde{\ell P + Q})$. Likewise, $\widetilde{\ell P}$ lies above the neutral point $0_E = (1 : 0)$ so is of the form $\widetilde{\ell P} = (X(\widetilde{\ell P}), 0) = \lambda_{0,P}(1, 0)$.

An important result on cubical arithmetic is that the (square of) the non reduced Tate pairing is precisely given by the monodromy: $e_{T,\ell}(P, Q)^2 = \lambda_{1,P}/\lambda_{0,P} = Z(\widetilde{\ell P + Q})/X(\widetilde{\ell P})$.

The fact that we compute the square of the usual Tate or Weil pairing is not really a problem in practice when ℓ is odd (after all for the reduced Tate pairing it suffices to adjust the final exponentiation). But it loses one bit of information when ℓ is even. Luckily in this case we can use the action of the theta group $G(2(0_E))$ on cubical points to recover the usual Tate and Weil pairing rather than their squares.

Now when we say that we need to carefully adjust the standard doubling and differential additions formulas on the Montgomery Kummer line to get meaningful cubical arithmetic, as Algorithms 5.4 and 5.5 shows the usual Montgomery ladder is actually already almost the correct cubical ladder! The doubling Algorithm 5.4 is exactly the same. And a minor difference in Algorithm 5.5 is an extra factor of 4, which does not matter for pairings (it does matter for other applications of the cubical arithmetic, like the DLP monodromy leak). The major difference is as follows: both algorithms compute $X(R+S)X(R-S), Z(R+S)Z(R-S)$ from $X(R), Z(R), X(S), Z(S), X(R-S), Z(R-S)$ using exactly the same formulas (up to this factor 4). Now, the usual algorithm, which only cares about projective coordinates, compute $(X(R+S) : Z(R+S)) = (X(R+S)X(R-S)Z(R-S), Z(R+S)Z(R-S)X(R-S))$. For the cubical arithmetic, we really want to use $(X(R+S)X(R-S)/X(R-S), Z(R+S)Z(R-S)/Z(R-S))$, i.e. use two divisions rather than two multiplications. Luckily, during the cubical ladder to compute $\widetilde{\ell P}, \widetilde{\ell P + Q}$, the base points $R-S$ will be P, Q, or $P+Q$. That's why we need the inverses of $x(P), x(Q), x(P+Q)$ in Theorem 1.1, which the usual projective Montgomery ladder does not need; once these have been precomputed, the two divisions by $X(R-S)$ and $Z(R-S)$ only become one multiplication by $1/x(R-S)$ (since our cubical points $\widetilde{P}, \widetilde{Q}, \widetilde{P+Q}$ have been normalised to have $Z=1$).

1.3. **Overview of the algorithm: the conceptual point of view.** Now we need to explain where this cubical arithmetic which gives pairings come from. We have seen above that our pairings were given by some kind of monodromy information. The correct framework for these monodromy considerations is the concept of biextension.

Biextensions were introduced by Mumford in [Mum69], and their theory thoroughly developed in [Gro72, Exposés VII et VIII]. As mentioned above, biextensions provide the correct theoretical framework to study pairings on abelian varieties and even abelian schemes, and as explained by Grothendieck in [Gro72] allows to keep track of pairing informations on a Néron model even when the special fiber degenerates to a non semi-abelian variety. Notably, he uses biextensions to constructs a pairing between the connected components of the special fiber of the Néron model of an abelian variety A and the one of its dual $\hat{A}$. This pairing is the key in his proof of the semi-stability theorem that an abelian variety always admits a semi-abelian Néron model over a finite field extension.

Now, reading through the 179 pages of abstract cohomological diagram chasing arguments of [Gro72, Exposés VII et VIII] might make biextensions seem like a very abstract theoretical concept, suitable to prove theorems but with no algorithmic applications. This was the impression of the author until recently. Luckily, Stange in [Sta08; Sta11] showed the algorithmic applications of biextensions (in the guise of elliptic nets), and in [Sta08, Theorem 17.1.1] she extends Grothendieck's monodromy interpretation of the Weil-Cartier pairing to the case of the Tate pairing.

And in fact, a biextension is something very concrete. Let us detail the case of the biextension $X_{(0_E)}$ associated to the canonical polarisation (0_E) on an elliptic curve E. Let $D=(0_E)$, a biextension element is a tuple $(P,Q,g_{P,Q}) \in X_{(0_E)}$ where $P,Q \in E$ and $g_{P,Q}$ is a function with divisor $D_{P+Q}+D_0-D_P-D_Q$. Here D_P denotes the divisor $(-P)-(0_E)$ (so $D_0=0$ and we will often drop it in the notations). The unusual convention on the signs will be explained in Section 1.6. Modulo our non standard sign convention, the functions $g_{P,Q}$ are

exactly like the functions $\mu_{P,Q}$ we use in pairing based cryptography (which are usually normalised at infinity).

There are two partial group laws $\star_1, \star_2$ on the biextension, which can be used to compute a product $(P_1, Q_1, g_{P_1,Q_1}) \star (P_2, Q_2, g_{P_2,Q_2})$ whenever $Q_1 = Q_2$ for $\star_1$ and whenever $P_1 = P_2$ for $\star_2$. We refer to Equations (9) and (10) for the definitions. A surprising, but very useful fact, is that the biextension $X_{(0_E)}$ is symmetric, which means that $\star_2 = \iota^* \star_1$ where ι is the swapping of arguments.

Now in the context of the non reduced Tate pairing $e_{T,\ell}(P, Q)$, by [Stao8] its monodromy interpretation via biextensions is as follows: take any rational biextension element $(P, Q, g_{P,Q})$ above (P, Q) and compute the biextension exponentiation $(P, Q, g_{P,Q})^{\star_1,\ell} = (\ell P, Q, g_{\ell P,Q})$. Since $\ell P = 0$, $g_{\ell P,Q}$ has for divisor $D_{\ell P+Q} + D_0 - D_{\ell P} - D_Q = 0$ so is a constant. This constant is precisely $e_{T,\ell}(P, Q)$, or more precisely its class in $\mathbb{F}_q^*/\mathbb{F}_q^{*,\ell}$. In Section 3.4 we give a monodromy interpretation of the Ate and optimal Ate pairings; this seems to be new.

To get efficient pairing formulas, we need an efficient representation of biextension elements. The function $g_{P,Q}$ is completely determined, via its divisor, from P and Q, up to a constant; so is completely determined by its value at any base point R_0. We call this the evaluation representation, and we will see that in Section 3.3 that the biextension exponentiation in the evaluation representation is precisely Miller's algorithm. This subsumes Miller's algorithm in terms of biextension arithmetic.

To go further, we introduce cubical points and the cubical representation of biextension elements. A very informal way of describing the cubical representation, which will be made more rigorous in Section 4.5 is as follows: let $(mP, Q, g_{mP,Q}) = (P, Q, g_{P,Q})^{\star_1,m}$. The function $R \mapsto g_{mP,Q}(R)$ has for divisor $D_{mP+Q} + D_0 - D_{mP} - D_Q$. We can decompose it as a product of "cubical functions":

$$g_{mP,Q}(R) = \frac{Z(mP \widetilde{+ Q} + R) Z(\widetilde{R})}{Z(m\widetilde{P +} R) Z(\widetilde{Q +} R)}$$

where the cubical functions $\widetilde{R} \mapsto Z(mP \widetilde{+ Q} + R)$, $\widetilde{R} \mapsto Z(\widetilde{R})$, $\widetilde{R} \mapsto Z(m\widetilde{P +} R)$, $\widetilde{R} \mapsto Z(\widetilde{Q +} R)$, are completely determined from choices of $\widetilde{R}, \widetilde{P + R}, \widetilde{Q + R}, \widetilde{P}, \widetilde{Q}, \widetilde{P + Q}$ and are "functions" with divisors D_{mP+Q}, D_0, D_{mP}, and D_Q respectively. These divisors are not principal, so these "functions" do not make sense on E, but they do make sense as cubical functions.

And a way to represent these cubical functions is via their "evaluation" at a base point $\widetilde{R_0}$. So we represent the biextension elements $(P, Q, g_{P,Q})^{\star_1,m}$ by the cubical points

$$[m\widetilde{P +} R_0, \widetilde{Q +} R_0; \widetilde{R_0}, mP \widetilde{+ Q} + R_0].$$

In practice, we will use $R_0 = 0_E$ as our base point since it is the most convenient, and our representation will be given by $[\widetilde{mP}, \widetilde{Q}; \tilde{0}, m\widetilde{P +} Q]$.

The link with the previous evaluation representation is: $g_{mP,Q}(0_E) = \frac{Z(m\widetilde{P+}Q)Z(\tilde{0})}{Z(\widetilde{mP})Z(\widetilde{Q})}$). (Or more rigorously, since 0_E is a zero of order 1 of both sides, this becomes an equality after dividing both sides by the same uniformiser).

So using cubical points, we have a way to split the functions $g_{mP,Q}$ as quotients of cubical functions, and a way to split the evaluation representation $g_{mP,Q}(R_0)$ as a quotient of the evaluation of these cubical functions at R_0 (given in practice by the. value of Z on suitable cubical points). We formalize this concept of "cubical points" in Section 4; the interesting

things as already mentioned is that cubical points admits a cubical arithmetic (which is not defined everywhere, nor is given by partial group laws).

This cubical arithmetic arises from the notion of cubical torsor structure, which is developed in [Bre83] and [Mor85, Chapitre 1]. In particular, Breen introduces in [Bre83] the concept of symmetric biextension, and explains how a cubical torsor structure give symmetric biextensions and conversely. For a biextension X_D associated to a divisor D on an abelian variety A with structure sheaf $\mathcal{O}_A$, and interpreting D as a line bundle $\mathcal{L} = \mathcal{O}_A(D)$, i.e. as a $\mathbb{G}_m$ torsor for the Zariski (or étale or fppf) topology; the theorem of the square $\mathcal{L}_{a+b} \otimes \mathcal{L} \simeq \mathcal{L}_a \otimes \mathcal{L}_b$ (where $\mathcal{L}_a := t_a^*\mathcal{L}$ and $t_a : A \to A, P \mapsto a+P$ is the translation by a) already gives a squared structure. This squared structure induces a canonical cubical torsor structure, i.e. isomorphisms $\mathcal{L}_{a+b+c} \otimes \mathcal{L}_{a+b}^{-1} \otimes \mathcal{L}_{b+c}^{-1} \otimes \mathcal{L}_{a+c}^{-1} \otimes \mathcal{L}_a \otimes \mathcal{L}_b \otimes \mathcal{L}_c \otimes \mathcal{L}^{-1} \simeq \mathcal{O}_A$ satisfying various natural compatibility conditions; along with higher dimensional structures (which we won't need). This cubical torsor structure will be used for our cubical arithmetic, for which we give explicit formulas, and we use these explicit formulas to recover in Theorem 4.16 the fact from [Bre83; Mor85, Chapitre 1] that the cubical torsor structure determines the symmetric biextension X_D.

The story so far is: pairings can be interpreted as monodromy informations on biextensions, this monodromy can be computed via biextension exponentiations, and cubical points and their arithmetic provide a convenient way to compute the biextension arithmetic. This is starting to get a bit long winded, and we still need an efficient way to describe our cubical points and their arithmetic. Thanksfully, that's the last step we will need for Theorem 1.1. We will see in Section 4 that a cubical point $\widetilde{P}$ is abstractly a rigidification of our $\mathbb{G}_m$-torsor (i.e. line bundle) $\mathcal{L}$ at P. Going back to the case $\mathcal{L} = \mathcal{O}_E(2(0_E))$ of Kummer lines, we have the two sections $X, Z \in \Gamma(2(0_E))$, which give the projective coordinates $(X(P) : Z(P))$ of a point $P \in E$. We remark that the function $x = X/Z$ is a well defined function on the elliptic curve. But via our rigidification $\widetilde{P} : \mathcal{L}(P) \to \mathcal{O}_E(P) = k(P)$ we can interpret $\widetilde{P} \circ X, \widetilde{P} \circ Z$ as elements of $k(P)$ and so define their "values" at P, which we will denote by $X(\widetilde{P}), Z(\widetilde{P})$ (since the values depend on the choice of rigidification $\widetilde{P}$ above P; changing this rigidification $\widetilde{P}$ to $\lambda \cdot \widetilde{P}$ changes the coordinates by a factor λ: $X(\lambda \cdot \widetilde{P}) = \lambda \cdot X(\widetilde{P})$). We call this the affine lift representation of our cubical point $\widetilde{P}$, indeed since $x = X/Z$ is a genuine function on E, we have $X(\widetilde{P})/Z(\widetilde{P}) = x(P)$, so $(X(\widetilde{P}), Z(\widetilde{P}))$ is a point in $\mathbb{A}^2$ above the point $(X(P) : Z(P)) \in \mathbb{P}^1$.

Now as an aside, for our cubical representation, since we use the level 2 coordinates $X, Z \in \Gamma(2(0_E))$, this means that we will be working with the biextension $X_{2(0_E)}$ rather than $X_{(0_E)}$, hence compute the Tate and Weil pairings associated with the divisor $2(0_E)$, This is where the square factor in Theorem 1.1 comes from, compared to the usual Tate and Weil pairings associated to (0_E). We saw in Section 1.2 that we can still compute the standard Tate and Weil pairing when ℓ is even even while working on $X_{2(0_E)}$, by using the natural action of the theta group $G(2(0_E))$ on it.

1.4. **Comparison with elliptic nets and the theta coordinates algorithm.** We saw in Section 1.3 that Miller's algorithm is just a way to compute the arithmetic in biextensions via the evaluation representation.

Elliptic nets [Sta08; Sta11] give an alternative approach to compute pairings, and yet another approach is given through theta functions [LR10; LR15]. It was already remarked in [Tra14] that these two approaches are very similar, and how theta functions give "abelian varieties nets".

We can go further: both approaches are a way to represent cubical points and their arithmetic. In fact the theta function approach is precisely the affine lift representation of cubical points as defined in Section 1.3, when we use as coordinates a basis $\theta_0, \dots, \theta_m$ of theta functions.

For the elliptic nets approach, the idea is to work with the biextension $X_{(0_E)}$ and so cubical points for (0_E). We only have (up to a scalar) one global section $Z_1 \in \Gamma(0_E)$ (such that in our notations of Section 1.3, $Z = Z_1^2$), but we can still define the value $Z_1(\widetilde{P})$ of a cubical point. Given cubical points $\widetilde{P}, \widetilde{Q}, \widetilde{P+Q}$, the cubical arithmetic allows to compute all $\widetilde{mP+nQ}$ $m, n \in \mathbb{Z}$, and the associated elliptic net is then $Z_1(\widetilde{mP+nQ})$.

The main drawback of elliptic nets is that $Z_1(\widetilde{P})$ is not enough to determine P. It is actually quite remarkable though, that thanks to the recurrence relation of elliptic nets, the data of $Z_1(\widetilde{mP+nQ})$ for small values of m, n is enough to recover all of them. In Section 4.9, we introduce another representation of cubical points $\widetilde{P}$ for (0_E), which is simply given by $(P, Z_1(\widetilde{P}))$. This is enough to completely determine P (obviously), and also $\widetilde{P}$ except when $P = 0_E$ because in this case $Z_1(0_E) = 0$. We refer to Section 4.9.5 for more details.

In Section 4.9.4, we look at cubical points for $2(0_E)$. We could represent a cubical point $\widetilde{P}$ by $(P, X(\widetilde{P}), Z(\widetilde{P}))$, but since $(X(\widetilde{P}), Z(\widetilde{P}))$ is already enough to recover $(X(P) : Z(P))$, and since $(X(P) : Z(P))$ completely determines $\pm P$, the values $X(\widetilde{P}), Z(\widetilde{P})$ are almost enough to determine $\widetilde{P}$. Also when $P = 0_E$, $X(\tilde{0}_E) \neq 0$, hence X, Z are enough to determine $\tilde{0}_E$ too. As explained in Section 1.2, this is the representation used for Theorem 1.1.

1.5. **Applications.** The main goal of the paper is to give efficient generic pairing formulas. But the tools we develop for this, notably the efficient arithmetic of the biextension and cubical points associated to the divisor $2(0_E)$ on an elliptic curve, have applications beyond this goal.

In Section 6 we discuss some of these further applications: pairing based cryptography, radical isogenies, supersingularity testing, and a novel side channel attack which I call the monodromy leak. In particular, while pairings only use the biextension arithmetic, the monodromy leak application of Section 6.4 really needs the full power of the refined cubical arithmetic.

1.6. **Conventions and notations.** I had two choices when writing this paper. First, develop the general theory of cubical arithmetic for abelian varieties (or even abelian schemes), and then specialize only at the end to elliptic curves. But I feared this would seem too abstract.

The second approach would have been to restrict to elliptic curves only and develop only the biextension and cubical arithmetic associated to the divisors (0_E) and $2(0_E)$. But for the applications mentioned in Section 1.5, I really wanted to develop the general theory of cubical arithmetic over an abelian variety. Besides, isogeny based cryptography uses higher dimensional isogenies more and more, so deriving efficient formulas in any dimension seemed worthwhile.

In the end, I opted for a mix of the two approaches: either first describing the general case and then specializing to elliptic curves, or only detailing the case of elliptic curves and then quickly indicating the generalisation to abelian varieties (or let the reader work them out). This comes at a cost of some redundancy in the exposition, but I hope it will make the exposition more accessible.

I also tried to prove (almost) all statements using the explicit formulas given in this paper, rather than resorting to abstract proofs. Notably, I give explicit formula based proofs that the biextension Weil and Tate pairings are the standard Weil and Tate pairings (up to a sign), and that the cubical arithmetic induces the biextension arithmetic.

One particular conundrum I had in the exposition is the following. On abelian varieties, it is much more convenient to use the language of line bundles $\mathcal{L}$ than the language of divisors D. This is because we are almost always only interested in the isomorphism class of $\mathcal{L}$, and it is much easier to work with line bundles up to isomorphisms than to work with divisors up to linear equivalences.

On the other hand, in the case of elliptic curves, divisors have a very convenient representation as a (formal) sum of points (we don't have such convenient representations for abelian varieties). Since pairings on elliptic curves are my main application, I made the choice to work (as much as possible) with divisors.

A side effect is that I need different sign conventions than usual. Indeed, given a line bundle $\mathcal{L}$ on an abelian variety A, the polarisation associated to $\mathcal{L}$ is $\Phi_{\mathcal{L}} : A \to \widehat{A}, P \mapsto t_P^*\mathcal{L} \otimes \mathcal{L}^{-1}$, where t_P is the translation by P. If $\mathcal{L}$ is induced by a divisor D, we can rewrite $\Phi_{\mathcal{L}}$ as $\Phi_D : A \to \widehat{A}, P \mapsto t_P^*D - D = t_{-P}D - D$ (or rather its linear equivalence class). In particular, the canonical polarisation $\Phi_{(0_E)}$ associated to E on an elliptic curve is $P \mapsto (-P) - (0_E)$. Notice the sign change compared to the usual identification of E and $\hat{E}$! But the one we use in this article is really the correct one according to [Con04, Example 2.5]. To mitigate this, we introduce the following notation: $D_P := \Phi_D(P) = t_P^*D - D$. This notation has the convenient advantage that a cycle $Z = \sum n_i(P_i)$ on an abelian variety is mapped through Φ_D to the divisor $\sum n_i D_{P_i}$ (we also have $D_0 = 0$), so we can use a cycle notation for both elliptic curves and abelian varieties.

An apology: I gave up on using the language of divisors in Section 4 to define cubical points. Indeed a cubical point is a rigidification of the line bundle $\mathcal{L}$ at P. This can be rephrased in terms of a choice of a non zero local section evaluated at P, which can also be described using the language of divisors, but the formalism of rigidifications is much more practical, so I switched languages.

1.7. **Genesis and thanks.** The starting point of this research project was the wonderful paper [CHM+23]. In this paper, the authors use self pairings to find weak instances of class group action isogeny based cryptography.

In a nutshell: assume that $\mathrm{End}(E) = \mathbb{Z}[\sqrt{\alpha}]$ is of discriminant Δ and α is totally imaginary. Then the Weil-Cartier pairing associated to α is a non degenerate pairing $e_\alpha : E[\alpha] \times E[\tilde{\alpha}] \to \mu_\Delta$. Furthermore, $E[\alpha]$ is cyclic (otherwise α would be divisible by an integer), and $\tilde{\alpha} = \overline{\alpha} = -\alpha$, so $E[\alpha] = E[\tilde{\alpha}]$. We obtain a non degenerate self pairing e_α on the cyclic group $E[\alpha]$ which can be used to recover torsion point informations for isogenies arising from the class group action. (In [CHM+23] the authors mainly use the "generalised α-Tate pairing" rather than the α-Weil-Cartier pairing, but the overall approach is the same.)

An open question in that paper is how to compute this α-Weil-Cartier pairing and the generalised α-Tate pairing, without going back to the usual Weil and Tate pairings (which can be costly).

At that time, I knew that the arithmetic of theta groups naturally gave rise to the Weil and Tate pairings, so I tried to extend [CHM+23] by looking at theta group informations (not necessarily coming from self pairings) preserved by class group isogenies. This is how I first found the monodromy leak attack of Section 6.4, formulated at the time in term of canonical lift of points of ℓ-torsion in the theta group rather than in terms of cubical points. I only found out afterwards that Lauter and Stange already had very similar ideas much earlier in [LS08], using the elliptic nets framework.

The key idea to rephrase the theta group approach in terms of biextensions (biextensions are a convenient way to package families of theta groups) is due to Stange, who mentioned

during a discussion last year with the authors of [CHM+23] and myself that elliptic nets were a way to compute the biextension arithmetic, as she had proven in [Sta11, Chapter 15].

Thanks to my work with David Lubicz in [LR10; LR15] on computing pairings via theta functions, I quickly realised that we could define "algebraic Riemann relations" (see Section 4.1) which could also be used to compute the arithmetic of biextensions. Like in our work where we used affine lift of theta points, I could define affine lift of points for any models of abelian varieties, use the algebraic Riemann relations to encode some sort of arithmetic on these affine points, and represent biextension elements via these affine points. Working out the formulas, I found out this generalised not only the theta coordinates approach but also the elliptic nets approach (see Sections 1.4 and 4.9.5), so I knew I was on the right path. I also implemented Theorem 1.1 in Sage in September 2023, and it did indeed compute correctly the pairings (and much faster than Sage's default implementation)! Trying to make sense of the corresponding arithmetic of these affine lifts of projective points, I found out, thanks to the work of [Bre83] and [Mor85, Chapitre 1], that the correct notion was the cubical torsor structure. Thanks to these firm existing theoretical foundations, extending the work of [Sta11; LR10; LR15] to the general case of cubical points and cubical arithmetic was straightforward.

In summary, this paper owes a lot to Stange and her PhD on elliptic nets, and obviously to Lubicz through our collaboration on [LR10; LR15]. I also benefited from various discussions with Stange and the authors of [CHM+23], notably with Castryck and Vercauteren; and also with Reijnders about [Rei23] and with Guillevic on the current state of the art on pairing based cryptography. Notably, I had not realised before [Rei23] that generic pairings in isogeny based cryptography were so slow; this motivated me to write Theorem 1.1 in the Montgomery model.

I also thank the authors of [CLZ24] for sending me a preliminary version of their pairing formulas. Lastly, special thanks are due to Giacomo Pope who converted my toy Sage implementation from [Rob23b] to an efficient Rust implementation.

We thank Jianming Lin for pointing several typos in the formulas.

1.8. **Related work.** As mentioned, the best generic pairings formula I found in the litterature are from [BELL10]. The paper [Rei23] by Reijnders focus specifically on pairings for isogeny based cryptography.

There has been relatively recent work in [DZZZ22; DZZ23] to optimize pairing formula for pairing based cryptography in the odd embedding degree case, which is interesting because this case is closer to the generic case than the even embedding degree case with denominator elimination. This work has been applied to pairings for isogeny based cryptography in [LWXZ23] and very recently the paper [CLZ24] gave fast generic pairing formulas for supersingular curves over $\mathbb{F}_{p^2}$ via Miller's algorithm in modified Jacobian coordinates, with $10M + 7S$ for a Miller doubling and $20M + 11S$ for a Miller double and add.

1.9. **Outline.** We first define the arithmetic of theta groups in Section 2, and how to recover pairings from this arithmetic. We then move on to the theory of biextensions in Section 3, which are a convenient way to package families of theta groups, we also reformulate pairings in terms of monodromy on biextensions. We then introduce cubical points and the cubical arithmetic as a refinement of the biextension arithmetic in Section 4, and we reframe yet again pairings using cubical points. In Section 5 we specialize our formulas to the case of pairings on Kummer lines, and prove Theorem 1.1. We briefly mention some applications in Section 6 and give some perspectives in Section 7.

Warning: This paper was supposed to be written soon after my talk on this subject at the Leuven Isogeny Days 4 in October 2023. However, this plan was sidetracked by the

discovery of the Clapotis algorithm, and its applications. Meanwhile, several people have already started using the cubical arithmetic formulas for faster pairing computations in isogeny based cryptography. This current version of the article is a preliminary version that I am publishing as a preprint paper because a larger diffusion of the algorithm to the community is probably worth it. Beside I could not resist the idea of publishing a preliminary version on April first. Beware that there are probably many typos (I mean April fool's!) still.

Update for April 16: the paper is now complete; apart from the many remaining typos I still need to correct.

2. Theta groups arithmetic

We first introduce theta groups of elliptic curves in Section 2.1 and how to use the theta group arithmetic to compute the usual Weil and Tate pairings associated to the divisor (0_E) on elliptic curves in Section 2.2. We briefly describe the general case of pairings on an abelian variety with a polarisation Φ_D in Section 2.3. As mentioned in Section 1, this general case of handling non necessarily principally polarised abelian varieties will be useful even in the context of elliptic curves, because when ℓ is even, we can use the theta group action of $G(2(0_E))$ to compute the standard Weil and Tate pairing rather than their square even while working with the non principal polarisation $2(0_E)$.

2.1. Theta groups on elliptic curves.

Biextensions are families of theta groups. We first define theta groups and explain their link with pairings, before introducing biextensions. For much more details on theta groups we refer to Mumford [Mum70; Mum66].

Let k be a field and E/k be an elliptic curve, and D a divisor of degree n. There is an associated polarisation $\Phi_D : E \to \hat{E}, P \mapsto t_P^*D - D = t_{-P}D - D \simeq D - t_PD$, whose kernel is $\mathrm{Ker}\,\Phi_D = E[n]$. (If $n = 0$, we make the convention that $E[0] = E$.) The theta group $G(D)$ is an extension of $E[n]$ by $\mathbb{G}_m$, defined as follows: its elements are given by couples $(P, g_P) \in G(D)$ where $P \in \mathrm{Ker}\,\Phi_D$ and g_P is any function whose divisor is the principal divisor $t_P^*D - D$.

For simplicity, we will often refer to the element $(P, g_P) \in G(D)$ simply by the function g_P. We will also say that $(P, g_P) \in G(D)(k)$ if $P \in E(k)$ and g_P is defined over k. In the special case where $P = 0_E$, the divisor $t_{0_E}^*D - D$ is trivial, so g_{0_E} is a constant.

The (non commutative) group law is given by

$$(P, g_P) \cdot (Q, g_Q) = (P + Q, g_P(\cdot)g_Q(\cdot + P)). \tag{1}$$

There is also a canonical action of $G(D)$ on $\Gamma(D)$, given for $s \in \Gamma(D)$ (i.e. a function such that $\mathrm{div}\, s + D \geq 0$) by

$$(P, g_P) \cdot s = g_P(\cdot)s(\cdot + P). \tag{2}$$

Given two divisors D_1, D_2, and an element $P \in \pi_1(G(D_1)) \cap \pi_2(G(D_2))$, where $\pi : G(D) \to E[\deg D]$ is the projection map $(P, g_P) \mapsto P$, we have a morphism:

$$(P, g_{1,P}) \in \pi_1^{-1}(P), (P, g_{2,P}) \in \pi_2^{-1}(P) \mapsto (P, g_{1,P}g_{2,P}) \in G(D_1 + D_2). \tag{3}$$

Likewise, if $D_1 \sim D_2$, we have an isomorphism $G(D_1) \simeq G(D_2)$. Namely, if α is any function with divisor $D_2 - D_1$, this isomorphism is given by

$$(P, g_P) \in G(D_1) \mapsto (P, g_P(\cdot)\alpha(\cdot + P)/\alpha(\cdot)) \in G(D_2). \tag{4}$$

We remark that the isomorphism does not depend on the choice of α.

If $c \in E$, we also have an isomorphism:

$$(P, g_P) \in G(D) \mapsto (P, t_c^*g_P) \in G(t_c^*D). \tag{5}$$

Example 2.1. Let $D_\ell = \ell(0_E)$ for $\ell \in \mathbb{N}^*$, then $\mathrm{Ker}\,\Phi_D = E[\ell]$. An element $(P, g_P) \in G(D_\ell)$ above $P \in E[\ell]$ is a function $\mathfrak{f}_{\ell,P}$ (not necessarily normalised) with divisor $\ell(-P) - \ell(0_E)$.

This is (up to a change of sign in P) the usual Miller function involved in the Tate and Weil pairing. Let us explain how to recover the Weil pairing in the context of the theta group $G(D_\ell)$.

Given another element $(Q, g_Q) = \mathfrak{f}_{\ell,Q} \in G(D_\ell)$, the failure of the commutativity of the group law of $G(D_\ell)$ is measured by the commutator $[(P, g_P), (Q, g_Q)] = (P, g_P)(Q, g_Q)(P, g_P)^{-1}(Q, g_Q)^{-1}$. We compute $(g_P \cdot g_Q)(x) = \mathfrak{f}_{\ell,P}(x)\mathfrak{f}_{\ell,Q}(x+P)$ while $(g_Q \cdot g_P)(x) = \mathfrak{f}_{\ell,Q}(x)\mathfrak{f}_{\ell,P}(x+Q)$. We thus have $g_P \cdot g_Q = \lambda g_Q \cdot g_P$ with $\lambda = \frac{\mathfrak{f}_{\ell,P}((x)-(x+Q)))}{\mathfrak{f}_{\ell,Q}((x)-(x+P))}$, for any x. Here we use the usual convention for a function f evaluated on a divisor $D = \sum n_i(P_i)$ of degree 0: $f(D) = \prod f(P_i)^{n_i}$. We recover the usual formula for the Weil pairing $e_{W,\ell}$ (up to a sign depending on the sign convention), which is not surprising since Mumford proves in [Mum70, p. 183] that the commutator pairing is the Weil pairing.

2.2. **Theta groups and pairings for elliptic curves.** We will be mainly interested in the case where D is a degree zero divisor. In this case, $\mathrm{Ker}\,\Phi_D = E$, the commutator pairing is trivial, so $G(D)$ is a commutative extension of E by $\mathbb{G}_m$.

Up to linear equivalence, we can assume that D is of the form $D = D_Q := \Phi_{(0_E)}(Q) = t_Q^*(0_E) - (0_E) = (-Q) - (0_E) \in \hat{E} := \mathrm{Pic}^0(E)$. The theta group $G(D_Q)$ thus gives an element of $\mathrm{Ext}^1(E, \mathbb{G}_m)$, which gives an explicit isomorphism $\hat{E} \simeq \mathrm{Ext}^1(E, \mathbb{G}_m), D_Q \mapsto G(D_Q)$. Using this isomorphism and the long exact sequence of cohomology, we obtain a canonical isomorphism between $\mathrm{Ker} f, f : E_1 \to E_2$ an isogeny, and the Cartier dual of the kernel $\mathrm{Ker}\,\hat{f}$ of the dual isogeny. This isomorphism induces the usual Weil-Cartier pairing, and the standard Weil pairing is the Weil-Cartier pairing associated with the polarisation $\Phi_{\ell(0_E)} = \ell\Phi_{(0_E)}$, where $\Phi_{(0_E)} : P \mapsto (-P) - (0_E) \sim (0_E) - (P)$ is the canonical principal polarisation.

Remark 2.2. The formula for $\Phi_{(0_E)}$ might seem to be the opposite of the usual isomorphism taken for $E \simeq \hat{E}$, but is the correct one such that the pullback of the Poincare bundle gives an ample line bundle (see [Cono4, Example 2.5]). Since we want our arguments to apply "as is" to abelian varieties, we will stick with this choice. Unfortunately, this means that our sign convention on divisors will be the opposite of the usual ones taken in the pairing literature, for instance our function $\mu_{P,Q}$ will be the normalised (at infinity) function with divisor $(-P-Q) + (0_E) - (-P) - (-Q)$. If $\boldsymbol{\mu}_{P,Q}$ is the usual normalised function with divisor $(P) + (Q) - (P+Q) - (0_E)$, we have $\mu_{P,Q}(R) = 1/\boldsymbol{\mu}_{P,Q}(-R)$.

We now explain how to recover the Weil and Tate pairing from our theta groups $G(D_Q)$. If $P \in E$, we will denote by $g_{P,Q} \in G(D_Q)$ an element in the theta group above P; its divisor will then be $t_P^* D_Q - D_Q = D_{P+Q} - D_P - D_Q = (-P-Q) + (0_E) - (-P) + (-Q)$. The function $g_{P,Q}$ then also gives an element of $G(D_P)$ above Q.

Now, given $P_1, P_2 \in E$ and two functions $g_{P_1,Q}, g_{P_2,Q} \in G(D_Q)$ above P_1, P_2 respectively, we have two possible group operations. The first one is given by the multiplication

$$(P_1, g_{P_1,Q}) \cdot (P_2, g_{P_2,Q}) = (P_1 + P_2, g_{P_1,Q}(\cdot)g_{P_2,Q}(\cdot + P_1)) \tag{6}$$

in the theta group $G(D_Q)$. The second one is to interpret $(Q, g_{P_i,Q})$ as an element of $G(D_{P_i})$ above Q and use Equation (3) to get an element $(Q, g_{P_1,Q}(\cdot)g_{P_2,Q}(\cdot))$ above Q in the theta group $G(D_{P_1} + D_{P_2})$. Now $D_{P_1} + D_{P_2} \simeq D_{P_1+P_2}$, so if g_{P_1,P_2} is any function with divisor $D_{P_1+P_2} - D_{P_1} - D_{P_2} = (-P_1-P_2) + (0_E) - (-P_1) - (-P_2)$ then by Equation (4) we have an

isomorphism $G(D_{P_1}+D_{P_2}) \simeq G(D_{P_1+P_2})$, $(Q,g_Q) \mapsto (Q, g(\cdot)g_{P_1,P_2}(\cdot+Q)/g_{P_1,P_2}(\cdot))$. Composing this isomorphism with the map above, we get an element $(Q,g) \in G(D_{P_1+P_2})$ which induces an element $(P_1+P_2, g_{P_1+P_2,Q}) \in G(D_Q)$. This gives us the second group operation:

$$(P_1,g_{P_1,Q}) \cdot' (P_2,g_{P_2,Q}) = \left(P_1+P_2, g_{P_1,Q}(\cdot)g_{P_2,Q}(\cdot)\frac{g_{P_1,P_2}(\cdot+Q)}{g_{P_1,P_2}(\cdot)}\right). \tag{7}$$

It is not obvious, but Equations (6) and (7) actually give the same group law:

Proposition 2.3. $(P_1,g_{P_1,Q})\cdot'(P_2,g_{P_2,Q}) = (P_1,g_{P_1,Q})\cdot(P_2,g_{P_2,Q}) = (P_1+P_2,g_{P_1+P_2,Q})$.

Proof. This follows from the unicity of the biextension associated to the principal polarisation $\Phi_{(0_E)}$, see Proposition 3.4 and Equation (11). □

Corollary 2.4. *Let $(P,g_{P,Q}) \in G(D_Q)$. Then $(P,g_{P,Q})^\ell = (\ell P, g_{P,Q}(\cdot)^\ell \mathfrak{f}_{\ell,P}((\cdot+Q)-(\cdot))$, where $\mathfrak{f}_{\ell,P}$ is a function with divisor $D_{\ell P} - \ell D_P = -\ell(-P) + (-\ell P) + (\ell-1)(0_E)$.*

Proof. By Proposition 2.3, we can use Equation (7) rather than Equation (6) when computing $(P,g_{P,Q})^\ell$. We obtain $(P,g_{P,Q})^\ell = (\ell P, g_{P,Q}(\cdot)^\ell (g_{P,P}g_{P,2P}\cdots g_{P,(\ell-1)(P)})((\cdot+Q)-(\cdot))$ and we observe that $(g_{P,P}g_{P,2P}\cdots g_{P,(\ell-1)(P)})$ has for divisor $D_{\ell P} - \ell D_P$. □

Corollary 2.5. *If $P,Q \in E[\ell]$, take any $(P,g_{P,Q}) \in G(D_P)$, and let λ_P the constant such that $(P,g_{P,Q})^\ell = (0_E,\lambda_P)$ (alternatively, $(P,g_{P,Q})^{\ell+1} = (P,\lambda_P g_{P,Q})$). Likewise, let λ_Q be the constant such that $(Q,g_{P,Q})^\ell = (0_E,\lambda_Q)$, where $(Q,g_{P,Q}) \in G(D_Q)$. Then (up to a sign), the Weil pairing $e_{W,\ell}(P,Q) = \lambda_P/\lambda_Q$.*

Assume that $k=\mathbb{F}_q$, $\mu_\ell \subset \mathbb{F}_q$, and let $Q \in E(\mathbb{F}_q)$, $P \in E[\ell](\mathbb{F}_q)$. Take any $(P,g_{P,Q}) \in G(D_Q)(\mathbb{F}_q)$. Then $(P,g_{P,Q})^\ell = (0_E,\lambda_P)$ where λ_P is (up to a sign) the non reduced Tate pairing $e_{T,\ell}(P,Q)$. And $(P,g_{P,Q})^{q-1} = (0_E,\lambda'_P)$ (alternatively, $(P,g_{P,Q})^q = (P,\lambda'_P g_{P,Q})$), where λ'_P is the reduced Tate pairing.

Proof. This follows from Corollary 2.4 and the usual formulas for the Tate and Weil pairing.

Indeed, we have $\lambda_P/\lambda_Q = \frac{g_{P,Q}(\cdot)^\ell \mathfrak{f}_{\ell,P}((\cdot+Q)-(\cdot))}{g_{P,Q}(\cdot)^\ell \mathfrak{f}_{\ell,Q}((\cdot+P)-(\cdot))} = e_{W,\ell}(P,Q)$, and $\lambda_P = g_{P,Q}(\cdot)^\ell \mathfrak{f}_{\ell,P}((\cdot+Q)-(\cdot))$ is equivalent to the non reduced Tate pairing $\mathfrak{f}_{\ell,P}((\cdot+Q)-(\cdot))$ since $g_{P,Q}$ is assumed to be rational.

For the last statement, we remark that $(P,g_{P,Q})^{q-1} = \left((P,g_{P,Q})^\ell\right)^{(q-1)/\ell} = (0_E,\lambda_P)^{(q-1)/\ell} = (0_E,\lambda_P^{(q-1)/\ell})$.

As a corollary, we see that λ_P/λ_Q does not depends on the choice of $g_{P,Q}$, and likewise for the class of λ_P modulo $\mathbb{F}_q^{*,\ell}$. This can be directly seen as follows: changing $g_{P,Q}$ by $\lambda g_{P,Q}$ changes λ_P to $\lambda^\ell\lambda_P$. □

Example 2.6 (Radical isogenies)**.** Let $P \in E[\ell]$ of exact order ℓ, and consider the isogeny $\phi : E \to E' = E/\langle P\rangle$. By descent theory [Mum66, Proposition 1 p. 291], the divisor D_Q descends to E' (i.e., there exists some rational degree 0 divisor D' such that $\phi^*D' = D_Q$) if and only if the kernel $\mathrm{Ker}\,\phi$ lifts to a rational subgroup in the theta group $G(D_Q)$. Since $\mathrm{Ker}\,\phi$ is cyclic, this is equivalent to finding a rational element $(P,g_P) \in G(D_Q)$ above P of order ℓ.

Take $g_P \in G(D_Q)(k)$ an arbitrary rational element. We have $(P,g_P)^\ell = (0_E,\lambda_P)$ where λ_P is the non reduced Tate pairing by Corollary 2.5. And $(P,\mu g_P)^\ell = (0_E,\mu^\ell\lambda_P)$. It follows that we can find a rational g_P of order ℓ if and only the non reduced Tate pairing $e_{T,\ell}(P,Q)$ is an ℓ-th power.

Now by definition of the dual isogeny $\hat{\phi}$, we have $\hat{\phi}(D') = \phi^*(D_Q)$. We have proved that $\hat{\phi}^{-1}(D_Q)$ contains a rational point D' if and only if the non reduced Tate pairing is an ℓ-th power. We recover (a special case of) the geometric interpretation from [Rob23c] of the Tate pairing as an étale torsor.

Example 2.7 (The Tate pairing as an obstruction to finding rational elements of ℓ-torsion in the theta group)**.** Let $D = \ell(0_E)$ and $P \in E[\ell]$ a point of ℓ-torsion. A natural question is whether there is an element $(P, g_P) \in G(D)$ which is both rational and still of ℓ-torsion. (Equivalently: the group $K = \langle P \rangle$ admits a rational lift to $G(D)$, hence by descent theory D descends to a rational divisor on E/K.)

The divisor D is symmetric, so we can even ask for symmetric lifts of order ℓ. If ℓ is odd, there are two symmetric elements above P, and a unique one of order ℓ, which has to be rational by unicity.

If ℓ is even, the symmetric elements are of order ℓ or 2ℓ; this obstruction is measured by $e_{D,\star}(\ell/2 \cdot P) = \pm 1$ in the notation of [Mum66, p.307–309]. For our divisor, $e_{D,\star}(\ell/2 \cdot P) = 1$, so the two symmetric elements are of order ℓ. But they might live in a quadratic extension.

Let g_P be an arbitrary theta group element above P. We can compute $(P, g_P)^\ell$, this is a constant λ_P, equal to $g_P(x)g_P(x+P)g_P(x+2P)\cdots g_P(x+(\ell-1)P)$. We can correct g_P by a rational projective factor α so that $(P, \alpha g_P)$ is of ℓ-torsion if and only if $\lambda_P \in k^{*,\ell}$. Since g_P has for divisor $D_P = \ell(-P) - \ell(0_E)$, we see that the class of λ_P in $k^*/k^{*,\ell}$ is given by (the non reduced Tate pairing): $\prod_{i=1}^{\ell-1} g_P(x+iP)/g_P(x) = \prod_{i=1}^{\ell-1} e_{T,\ell(0_E)}(P, iP) = e_{T,\ell(0_E)}(P, \ell(\ell-1)/2P)$.

In particular, we see that the obstruction vanishes for ℓ odd (as we already knew), and that for ℓ even, the quadratic obstruction is given by whether the reduced Tate pairing of P and $\ell/2 \cdot P$ is equal to 1 or -1.

We already saw λ_P in [RS24, § 3.1] for the case $\ell = 2$, where the same obstruction governed the possible formulas for a 2-isogeny on a Kummer line.

2.3. **Theta groups and pairings for polarized abelian varieties.** As explained in the introduction, for simplicity we mainly work with elliptic curves, but we carefully state our results such that they are easily adapted for any abelian varieties.

In this section, we briefly explain what happens for a general polarised abelian variety (A, Θ_A), with Θ_A an ample divisor. Usually when working with abelian varieties, it is more convenient to work with (isomorphisms classes of) line bundles than (linear equivalence classes of) divisors, because unlike for elliptic curves divisor do not have a convenient description. Here for the sake of uniformity we stick to divisors.

Fix any divisor D. There is a morphism associated to D, $\Phi_D : A \to \widehat{A}, P \mapsto t_P^* D - D$, it only depends on the algebraic equivalence class of D. We denote by $A[D]$ its kernel. We remark that Φ_D is a polarisation when $D = \Theta_A$ is ample. If $D = n\Theta_A$, with Θ_A ample giving a principal polarisation, then $A[D] = A[n]$.

For simplicity, we will use the same notation to denote $\Phi_D(P)$ as the divisor $D_P := t_P^* D - D$, and as an element of $\widehat{A} = \mathrm{Pic}^0(A)$, i.e. as a linear equivalence class.

An element (P, g_P) of the theta group $G(D)$ is given by a point $P \in A[D]$ and a function with divisor $D_P := \Phi_D(P)$, with group law given by Equation (1), and natural action on sections given by Equation (2). Equations (3) and (4) also apply for abelian varieties.

2.3.1. *Pairings on an abelian variety.* We first introduce the Weil-Cartier pairing for an abelian variety.

The reader can skip without harm the following paragraph, which will only serve as a motivation for the notion of biextension in Section 3.1. Let A/k be an abelian variety, and

$B\mathbb{G}_m = [\mathrm{Spec}\, k/\mathbb{G}_m]$ be the classifying stack associated to $\mathbb{G}_m$ (say for the fppf topology). Then we define $\widehat{A}/k$ to be $\mathrm{Hom}(A, B\mathbb{G}_m)$, where we require the morphisms to be morphisms of Picard stacks. By definition of the classifying stack $B\mathbb{G}_m$, a morphism of stack $A \to B\mathbb{G}_m$ corresponds to a $\mathbb{G}_m$-torsor, i.e. a line bundle $\mathcal{L}$. Requiring the morphism to be a morphism of Picard stack imposes $\mathcal{L}$ to be translation invariant (and rigidifies it). Since translation invariant line bundles are the same as the ones which are algebraically equivalent to 0, this gives an isomorphism $\mathrm{Pic}^0(A) = \widehat{A} = \mathrm{Hom}(A, B\mathbb{G}_m)$. Now by general abstract nonsense, we have an isomorphism $\mathrm{Hom}(A, B\mathbb{G}_m) = \mathrm{Ext}^1(A, \mathbb{G}_m)$ as fppf sheafs. If $f : A \to B$ is a morphism of abelian varieties, $K = \mathrm{Ker} f$, applying the derived functor $\mathcal{R}\mathrm{Hom}(\cdot, \mathbb{G}_m)$ to the exact sequence $0 \to K \to A \to B \to 0$ gives a distinguished triangle in the derived category, whose associated long exact sequence is $0 \to \mathrm{Hom}(B, \mathbb{G}_m) \to \mathrm{Hom}(A, \mathbb{G}_m) \to \mathrm{Hom}(K, \mathbb{G}_m) \to \mathrm{Ext}^1(B, \mathbb{G}_m) \to \mathrm{Ext}^1(A, \mathbb{G}_m) \to \cdots$. Since $\mathrm{Hom}(A, \mathbb{G}_m) = 0$ because A is proper and $\mathbb{G}_m$ affine, and $\mathrm{Ext}^1(B, \mathbb{G}_m) \to \mathrm{Ext}^1(A, \mathbb{G}_m)$ is the dual isogeny $\hat{f}$ via the identification above, we see that $\mathrm{Ker}\hat{f} \simeq \mathrm{Hom}(K, \mathbb{G}_m)$ is canonically isomorphic to the Cartier dual of K. This abstract isomorphism gives the Weil-Cartier pairing $e_f : \mathrm{Ker} f \times \mathrm{Ker}\hat{f} \to \mathbb{G}_m$, via the identification above and the canonical pairing from Cartier duality: $K \times \mathrm{Hom}(K, \mathbb{G}_m) \to \mathbb{G}_m$. The standard Weil pairing is the Weil-Cartier pairing applied to the isogeny $[\ell]$.

We now recall the explicit definition of the Weil and Tate pairing associated to the polarisation Φ_Θ associated to an ample divisor $D = \Theta = \Theta_A$. For more details, we refer to [Rob21b, Chapter 4; Rob21a, Chapter 3; Rob23c] and the references there. The Weil pairing $e_{W,\Theta,\ell} : A[\ell\Theta] \times A[\ell\Theta] \to \mathbb{G}_m$ associated to Φ_Θ is the Cartier Weil pairing associated to the polarisation $\ell\Phi_\Theta : A \to \widehat{A}$, and it is also the commutator pairing on the theta group $G(\ell\Theta)$. If $(P, Q) \in A[\ell]$, and $e_{W,\ell}$ is the usual Weil pairing on $A[\ell] \times \widehat{A}[\ell]$, we have $e_{W,\Theta,\ell}(P, Q) = e_\ell(P, \Phi_\Theta(Q))$. Likewise, the Tate pairing $e_{T,\Theta,\ell} : A[\ell\Theta](k) \times \widehat{A}(k)/(\ell\Phi_\Theta A(k)) \to H^1(k, \mu_\ell)$ is the Tate-Cartier pairing associated to the polarisation $\ell\Phi_\Theta : A \to \widehat{A}$. If $P \in A[\ell](k)$, $e_{T,\Theta,\ell}(P, \Phi_\Theta(Q)) = e_{T,\ell}(\Phi_\Theta(P), Q)$, where $e_{T,\ell}$ is the standard Tate pairing associated to the isogeny $[\ell] : A \to A$. If $\Theta = m\Theta_1$, and $P, Q \in A[\ell]$ (resp. $P \in A[\ell](k)$), we have $e_{W,\Theta,\ell}(P, Q) = e_{W,\Theta_1,\ell}(P, Q)^m$, $e_{T,\Theta,\ell}(P, Q) = e_{T,\Theta_1,\ell}(P, Q)^m$.

Formulas are as follows: let Z_P, Z_Q be any degree 0 0-cycles equivalent to $(P) - (0)$ and $(Q) - (0)$, and $D_{Z_P}, D_{Z_Q} = \Phi_\Theta(Z_P), \Phi_\Theta(Z_Q)$. Here we extend Φ_Θ to cycles by additivity; by the theorem of the square their linear equivalence class only depends on P, Q.

Explicitly, if $Z = \sum n_i(P_i)$ is of degree 0, $D_Z := \sum n_i(t_{P_i}^* D - D) = \sum n_i D_{P_i}$ (and we remark that $D_P = D_{(P)-(0)}$). The divisor D_Z is principal if and only if $s(Z) := \sum n_i P_i \in A[D]$, in which case we will write f_Z or f_{D_Z} a function with divisor D_Z. In particular, $D_Z \sim D_{s(Z)}$.

We remark that if $P \in A[\ell\Theta]$, then by definition $\ell D_{Z_P} \sim \ell\Phi_\Theta(P)$ is a principal divisor; and by definition $f_{\ell Z_P}$ is any function with this divisor. We will evaluate it on degree 0 cycles, so the evaluation does not depends on the choice of $f_{\ell Z_P}$. We will also denote by $f_{\ell,P}$ the function associated to the divisor $D_{\ell P} - \ell D_P$ associated to the cycle $(\ell P) + (\ell - 1)0 - \ell(P)$.

We have

$$e_{T,\Theta,\ell}(P, Q) = f_{\ell Z_P}(Z_Q) \in k^*/k^{*,\ell}$$

if $Q \in A[\ell\Theta](k)$ and

$$e_{W,\Theta,\ell}(P, Q) = f_{\ell Z_P}(Z_Q)/f_{\ell Z_Q}(Z_P) \in \mathbb{G}_m$$

if $P, Q \in A[\ell\Theta]$. (As the proof of Theorem 2.9 will show, we even have $e_{W,\Theta,\ell}(P,Q) = f_{\ell Z_P}(t_x^* Z_Q)/f_{\ell Z_Q}(t_x^* Z_P)$ as long as we translate by the same point $x \in A$.)

Finally we remark that the formulas also holds for any divisor D instead of an ample divisor Θ, but in this case we won't have non degeneracy.

Remark 2.8. When we evaluate a function f at some cycle Z, it might happen that some of the points P in the support of Z are zeroes or poles of f. One way to still define the evaluation $f(Z)$ is to fix a uniformiser π_P at these points P, and to define $f(P)$ as the value of $(f/\pi_P^{v_P(f)})(P)$. We sometimes call this the extended value. We will see in the monodromy interpretation of pairings from Section 3 that our pairings will be obtained as a quotient of two functions f_1, f_2 which differ by a constant c. If we represent f_1, f_2 by their extended value at P, as long as we pick the same uniformizer π_P both for f_1, f_2, this constant c will be correctly determined by the extended value $f_1(P)/f_2(P)$ even if P is a pole or zero of the f_i.

2.3.2. *Pairings via theta groups.* We have seen in Section 2.2 that the map $D_Q = (-Q) - (0_E) \in \mathrm{Pic}^0(E) \mapsto G(D_Q) \in \mathrm{Ext}^1(E, \mathbb{G}_m)$ gives an *explicit* isomorphism $\hat{E} := \mathrm{Pic}^0(E) \simeq \mathrm{Ext}^1(E, \mathbb{G}_m)$. This extends to abelian varieties: the canonical isomorphism $\hat{A} \simeq \mathrm{Ext}^1(A, \mathbb{G}_m)$ is given by $D \in \mathrm{Pic}^0(A) \mapsto G(D)$. We can thus extend Corollary 2.5 to abelian varieties.

We fix D a divisor, and recall that we denote by D_Q the divisor $\Phi_D(Q) = t_Q^* D - D$. The theta group $G(D_Q)$ is an abelian extension of A by $\mathbb{G}_m$, whose elements are given by $(P, g_{P,Q})$ with $P \in A$ and $g_{P,Q}$ a function with divisor $t_P^* D_Q - D_Q = t_{P+Q}^*(D) + D - t_P^* D - t_Q^* D = D_{P+Q} - D_P - D_Q$. The group laws Equations (6) and (7) still hold for $G(D_Q)$, and they are the same because Proposition 2.3 is also valid.

The only subtlety now is that if $P \in A[\ell\Theta]$ and we take $(P, g_{P,Q}) \in G(D_Q)$, then $(P, g_{P,Q})^\ell = (\ell P, g_{\ell P,Q})$ needs not be given by a constant function $g_{\ell P,Q} = \lambda_P$, because P is not necessarily of ℓ-torsion. However, $\ell P \in A[D]$, and we are able to use the action of theta groups on sections.

Theorem 2.9. *Let D be a divisor on an abelian variety A, $Q \in A$, and $D_Q = \Phi_D(Q)$. Let $(P, g_{P,Q}) \in G(D_Q)$, with $P \in A[\ell D]$. Let $(\ell P, g_{\ell P}) \in G(D)$ be any element above ℓP. Let $g_{\ell P,Q}$ be such that $(P, g_{P,Q})^\ell = (\ell P, g_{\ell P,Q})$. Then $g_{\ell P,Q} g_{\ell P}((\cdot) - (\cdot + Q))$ is a constant λ_P. If $Q \in A[\ell D]$, the Weil pairing is (up to a sign) $e_{W,D,\ell}(P,Q) = \lambda_P/\lambda_Q$.*

If $P \in A[\ell D](k)$ and we take $g_{P,Q}, g_{\ell P}$ rational, then the non reduced Tate pairing is given by (up to a sign) $e_{T,D,\ell}(P,Q) = \lambda_P$.

We remark that $(Q, g_{\ell P,Q}) \in G(D_{\ell P})$ and that $g_{\ell P}^{-1}$ is a section of $D_{\ell P}$. By Equation (2), the constant $g_{\ell P,Q} g_{\ell P}((\cdot) - (\cdot + Q))$ is given by $\frac{(Q, g_{\ell P,Q}) \cdot g_{\ell P}^{-1}}{g_{\ell P}^{-1}}$.

Proof. First the function $g_{\ell P,Q}$ has for divisor $D_{\ell P+Q} - D_{\ell P} - D_Q$, while $g_{\ell P}$ has for divisor $D_{\ell P}$, so $g_{\ell P}((\cdot) - (\cdot + Q))$ has for divisor $D_{\ell P} - (D_{\ell P+Q} - D_Q)$. Their multiplication has trivial divisor, so is a constant.

Now the same proof as in Corollary 2.4 shows that $g_{\ell P,Q} = g_{P,Q}^\ell \mathfrak{f}_{\ell,P}((\cdot + Q) - (\cdot))$ with $\mathfrak{f}_{\ell,P}$ a function with divisor $D_{\ell P} - \ell D_P$. The function $\mathfrak{f}_{\ell,P}/g_{\ell P}$ has for divisor $-\ell D_P$. We now conclude as in Corollary 2.5, using the formulas for the Weil and Tate pairings associated to D. □

As a consequence of the definitions and the proof of Theorem 2.9, we have the following result which will be useful to get algorithms to compute the Ate and optimal Ate pairings.

Porism 2.10. *Let $(P_1, g_{P_1,Q}) \in G(D_Q)$, $(P_2, g_{P_2,Q}) \in G(D_Q)$, and $(P_1+P_2, g_{P_1+P_2,Q} \in G(D_Q)$ be their product. Let g_{P_1,P_2} be a function with divisor $D_{P_1+P_2} - D_{P_1} - D_{P_2}$. Then g_{P_1,P_2} evaluated on the cycle $(R+Q)-(R)$ is given by $\frac{g_{P_1+P_2,Q}}{g_{P_1,Q}g_{P_2,Q}}(R)$.*

For any P, Q, let $(P, g_{P,Q}) \in G(D_Q)$, $(P, g_P)^\ell = (\ell P, g_{\ell P,Q})$, and let $f_{\ell,P}$ a function with divisor $D_{\ell P} - \ell D_P$. Then the function $f_{\ell,P}$ evaluated on the cycle $(R+Q)-(R)$ is given by $\frac{g_{\ell P,Q}}{g^\ell_{P,Q}}(R)$.

And if $P \in A[\ell D]$, $f_{-\ell D_P}((R+Q)-(R)) = \frac{g_{\ell P,Q}}{g^\ell_{P,Q}}(R) g_{lP}((R)-(R+Q))$.

Remark 2.11. The reason we need to go through the trouble of explaining how to compute the Weil and Tate pairing associated to some divisor D on the full $A[\ell D]$ rather than just $A[\ell]$ is the following.

In Section 5, to compute pairings on Kummer lines, we will use the biextension and cubical arithmetic associated to the divisor $D = 2(0_E)$ to compute $e_{W,2(0_E),\ell}, e_{T,2(0_E),\ell}$. On the ℓ-torsion, these are the square of the usual Weil and Tate pairings $e_{W,(0_E),\ell}, e_{T,(0_E),\ell}$. This is not a really a problem when ℓ is odd, but this lose one bit of information when ℓ is even.

Instead, in this case we will write $\ell = 2m$, and use the fact that since $m\Phi_{2(0_E)} = \ell\Phi_{(0_E)}$, then by the standard compatibility between pairings and isogenies, we have $e_{W,2(0_E),m} = e_{W,(0_E),\ell}$, $e_{T,2(0_E),m} = e_{T,(0_E),\ell}$. But $E[m]$ is a strict subset of $E[\ell] = E[m(2(0_E))]$, so we need the full generality of Theorem 2.9 to handle this case properly.

One needs to be careful with the Tate pairing, because of its arithmetic nature (by contrast of the geometric nature of the Weil pairing). Let us assume that $D = mD_1$, D_1 a divisor associated to a principal polarisation. In Theorem 2.9, for the Weil pairing, when $P, Q \in A[\ell D] = A[\ell m]$, we correctly compute $e_{W,\ell D}(P,Q) = e_{W,\ell m D_1}(P,Q) \in \mu_{\ell m}$ via the monodromy.

However, the non reduced Tate pairing $e_{T,\ell D}(P,Q) \in k^*/k^{*,\ell}$ is computed in a smaller group than $e_{T,\ell m D_1}(P,Q) \in k^*/k^{*,\ell m}$.

Looking at the formula from Theorem 2.9 and Porism 2.10, we see that to get a value in $k^*/k^{*,\ell m}$ when using the divisor D, we need to also keep track of $g^\ell_{P,Q}$; the monodromy information λ_P is not enough (it only gives the information in $k^*/k^{*,\ell}$).

More precisely, we have $e_{T,\ell m D_1}(P,Q) = f_{-\ell D_P}((R+Q)-(R)) = \frac{g_{\ell P,Q}}{g^\ell_{P,Q}}(R) g_{lP}((R)-(R+Q)) \in k^*/k^{*,\ell m}$ (for any $R \in A(k)$), where $g_{\ell P,Q}(R) g_{lP}((R)-(R+Q))$ is the monodromy constant λ_P, and $\frac{1}{g^\ell_{P,Q}}(R)$ is the correcting factor to have the pairing in $k^*/k^{*,\ell m}$.

There is a special case where we can compute the correct value in $k^*/k^{*,\ell m}$ just from the monodromy information. We have a map $G(D_{1,Q}) \to G(D_Q)$ given by $g_{P,Q} \mapsto g^{\otimes m}_{P,Q}$, where the tensor product $g^{\otimes m}_{P,Q}$ is simply given by the function product: $g^{\otimes m}_{P,Q}(x) = g_{P,Q}(x)^m$. If, when applying Theorem 2.9, our $g_{P,Q} \in G(D_Q)$ comes from the m-th tensor product of a rational theta group element in $G(D_{1,Q})$, then the monodromy λ_P naturally gives the correct value of the Tate pairing $k^*/k^{*,\ell})$, because $g^\ell_{P,Q}(R)$ is then already in $k^{*,\ell m}$.

One last subtlety about the Tate pairing. For the Weil pairing $e_{W,\ell D}$, replacing D by an equivalent divisor D' still give the correct value $e_{W,\ell m D_1}$, not only on $A[\ell]$ but even on $A[\ell D]$. However, for the Tate pairing, for this to be the case, we need that $D' = mD'_1$, with D'_1 a rational divisor. (In particular, the class of $D'_1 - D_1 \in \mathrm{Pic}^0(A)$ is a point of m-torsion.) Indeed, when this is the case, we have $f_{\ell D_P}((Q)-(0))/f_{\ell D'_P}((Q)-(0)) = f_{\ell m(D_1-D'_1)_P}((Q)-(0))$, and since D_1 is algebraically equivalent to D'_1, $D_{1,P}$ is linearly

equivalent to $D'_{1,P}$, so $f_{\ell m(D_1-D'_1)_P}$ is a $m\ell$-th power. So we get the same value for the non reduced Tate pairing by using D or D'. $k(A)$).

3. Biextensions arithmetic

We define biextensions in Section 3.1, and interpret the Weil and Tate pairing as monodromy pairings in Section 3.2 (as already shown by Grothendieck and Stange respectively). We explain how to recover Miller's standard algorithm in terms of biextensions in Section 3.3, and we give a monodromy interpretation of the Ate and optimal Ate pairings in Section 3.4.

3.1. **Biextensions.** Biextensions were introduced by Mumford in [Mum69]. For a complete (but quite dry) reference, we refer to [Gro72, Exposés VII et VIII]. Beware of some sign errors in [Gro72, pp. VIII 2.3.10], corrected in [BBM79, § 5.1].

Let us first give an informal motivation for the notion of biextension. Pairings are bilinear maps. When working with modules, it is much more convenient to interpret a bilinear map $B : G_1 \times G_2 \to G_3$ as an element of $\mathrm{Hom}(G_1 \otimes G_2, G_3)$ than as an element of $\mathrm{Hom}(G_1, \mathrm{Hom}(G_2, G_3))$. In other words: we want the decurryfication of the map $G_1 \to G_2 \to G_3, g_1 \mapsto B(g_1, \cdot)$. A biextension is an analogue to a categorified decurryfication of the map $A \mapsto \hat{A} = \mathrm{Hom}(A, B\mathbb{G}_m) \simeq \mathrm{Ext}^1(A, \mathbb{G}_m)$.

We first begin with some abstract definitions and results before moving to much more concrete formulas. Given some abelian groups G_1, G_2, G_3 (over some topos), a biextension X of $G_1 \times G_2$ by G_3 is an element of the topos with projection maps π_1, π_2 to G_1, G_2 and an action of G_3 on X, such that for all $P_1 \in G_1$, $X_{P_1} := \pi_1^{-1}(P_1)$ is an extension of G_2 by G_3, and for all $P_2 \in G_2$, $X_{P_2} := \pi_2^{-1}(P_2)$ is an extension of G_1 by G_3. This defines two partial group structure $\star_2, \star_1$ on X, and we further require that they satisfy some "obvious compatibility relations". The biextensions of (G_1, G_2) by G_3 form a category $\mathrm{BiExt}(G_1, G_2; G_3)$.

More concretely, (working in the internal logic of the topos), an element $x \in X$ is said to be above (g_1, g_2) if $\pi_i(x) = g_i$. All other elements x' above (g_1, g_2) are of the form $g_3 \cdot x$ for a unique $g_3 \in G_3$: they form a torsor under G_3. We will often use the notation x_{g_1,g_2} to say that x is above (g_1, g_2). The biextension structure induces two partial group law. The first one $x_{g_1,g_2} \star_1 y_{g'_1,g_2} = z_{g_1+g'_1,g_2}$ is valid whenever $\pi_2(x) = \pi_2(y)$. The second one $x_{g_1,g_2} \star_2 y_{g_1,g'_2} = z_{g_1,g_2+g'_2}$ is valid whenever $\pi_1(x) = \pi_1(y)$. The "obvious compatibility relations" requires that given $x_{g_1,g_2}, x_{g_1,g'_2}, x_{g'_1,g_2}, x_{g'_2,g'_2}$, we have

$$(x_{g_1,g_2} \star_1 x_{g'_1,g_2}) \star_2 (x_{g_1,g'_2} \star_1 x_{g_1,g'_2}) = (x_{g_1,g_2} \star_2 x_{g_1,g'_2}) \star_1 (x_{g'_1,g_2} \star_2 x_{g'_1,g_2}). \tag{8}$$

For an description of the compatibility relations Equation (8) as a commutative diagram in the external logic, see [Gro72, VII Définition 2.1].

We summarize the main results of [Gro72] on biextensions (which we state for completeness, we won't need to use these results, but instead we will rely on down to earth computations with explicit formulas). The functor BiExt is triadditive [Gro72, VII.(2.6.1)], covariant and cofibrant in G_3, and contravariant and fibrant in G_1, G_2 [Gro72, VII § 2], and left exact in each argument [Gro72, VII Proposition 3.7.6]. The category $\mathrm{BiExt}(G_1, G_2; G_3)$ is a stack [Gro72, VII § 2.8], whose homotopical invariants are given by $\mathrm{BiExt}(G_1, G_2; G_3)^0 := \pi_1(\mathrm{BiExt}(G_1, G_2; G_3)) =$ group of endomorphisms of any biextension $X \simeq \mathrm{Hom}(G_1 \otimes G_2, G_3)$ [Gro72, VII § 2.5]; and furthermore $\mathrm{BiExt}(G_1, G_2; G_3)^1 := \pi_0(\mathrm{BiExt}(G_1, G_2; G_3))$, the set of isomorphism classes of biextensions, has a natural group structure defined in [Gro72, VII § 2.5] and such that $\mathrm{BiExt}(G_1, G_2; G_3)^1 \simeq \mathrm{Ext}^1(G_1 \overset{L}{\otimes} G_2, G_3)$ [Gro72, VII (3.6.5)]. This isomorphism is the main result of [Gro72, p. VII], and is used in [Gro72, p. VIII] to

define a pairing associated to a biextension [Gro72, VIII § 2]. This construction is extended in [Sta08, Theorem 17.1.1] to a Tate like pairing associated to a biextension.

Now we specialize these results to the case of abelian varieties (and then elliptic curves), where $G_1, G_2 = A, B$ are abelian varieties, and $G_3 = \mathbb{G}_m$ is the multiplicative group. This is the main case of interest of [Gro72, pp. VII, VIII], and Grothendieck proves:

Theorem 3.1 (Grothendieck). *Let A, B be abelian schemes. Then we have canonical isomorphisms (in the fppf topos)* $\mathrm{BiExt}(A, B; \mathbb{G}_m) \simeq \mathrm{BiRigidifiedTorsors}(A, B; \mathbb{G}_m) \simeq \mathrm{Correspondances}(A, B) \simeq \mathrm{Hom}(A, \widehat{B}) \simeq \mathrm{Hom}(B, \widehat{A})$.

Notably, given a morphism $f : A \to B$, there is a unique biextension X_f associated to it, and if f is an isogeny, the pairing associated to this biextension is the Weil-Cartier pairing e_f.

Proof. The first statement is [Gro72, VII Exemple 2.9.5 et Remarque 2.9.6], and the second is [Gro72, pp. VIII 2.3]. (It is stated to be the opposite of the Weil-Cartier pairing there, but there was a sign mistake corrected in [BBM79, § 5.1].) □

Example 3.2. For instance, applying Theorem 3.1 to the identity morphism $A \to A$, we obtain the (birigidified) Poincaré line bundle and the Poincaré biextension.

Let A be an abelian variety and D be an ample divisor, and $\Phi_D : A \to \widehat{A}$ be the associated polarisation. By Theorem 3.1, there is a unique biextension X_D associated to Φ_D (which uniquely depends on the polarisation, hence on the algebraic equivalence class of D). Since $\widehat{\widehat{A}} \simeq A$ by biduality, X_D is a biextension of $A \times A$ by $\mathbb{G}_m$, and the corresponding birigidified torsor is the pullback of the Poincaré line bundle by $\mathrm{Id} \times \Phi_D$, and suitably rigidified along $A \times 0$ and $0 \times A$.

Remark 3.3. The fact that the biextension pairing corresponds to the Weil-Cartier pairing for abelian schemes follows from abstract diagram chasings in [Gro72, pp. VIII 2.3]. A more elementary proof for elliptic curves is given by Stange in [Sta08, Theorem 17.1.2]; where it is also proven that her Tate like pairing associated to a biextension is indeed the usual Tate pairing for elliptic curves.

We will only need the unicity part of Theorem 3.1, and we will reprove below in Theorem 3.11 that the biextension pairings are the Weil and Tate pairings.

We have a canonical isomorphism $\iota : \mathrm{BiExt}(G_1, G_2; G_3) \simeq \mathrm{BiExt}(G_2, G_1; G_3)$ which consists in permuting the groups G_1, G_2 and the partial laws $\star_1, \star_2$. From the unicity part of Theorem 3.1 it follows that if $f : A \to B$ is a morphism of abelian schemes and X_f the associated biextension, we have $\iota(X_f) = X_{\hat{f}}$. Now if we apply this to our polarisation Φ_D, since it is autodual we obtain the following symmetry formula:

Proposition 3.4. *Let A be an abelian variety, $\Phi_D : A \to \hat{A}$ a polarisation associated to an ample divisor, and X_D the associated biextension of $A \times A$ by $\mathbb{G}_m$. Then X_D is symmetric: if $\iota(x_{a_1,a_2}) = \iota(x)_{a_2,a_1}$ denotes the same element seen above (a_2, a_1) rather than (a_1, a_2) (via the isomorphism $(a_1, a_2) \mapsto (a_2, a_1)$), then $x_{a_1,b} \star_1 x_{a_2,b} = \iota(x)_{b,a_1} \star_2 \iota(x)_{b,a_2}$.*

Proof. The unicity argument above shows that the biextension laws are the same, up to a global automorphism of biextension. But a biextension of abelian varieties by $\mathbb{G}_m$ only has trivial automorphisms. □

Remark 3.5. We even have the stronger statement that the biextension X_D is symmetric in the sense of Breen [Bre83, § 1] (i.e. the symmetry above is compatible with the various natural structures on the biextension), owing to the fact that (the line bundle associated to

D) is a cubical torsor (see [Bre83, S 2] and [Mor85, Chapitre 1, § 2, 3]). We will come back to this in Section 4.

We are now ready to state explicit formulas for the biextension associated to a polarisation Φ_D on an abelian variety or an elliptic curve. Since the polarisation Φ_D depends only on the algebraic equivalence class of D (which for an elliptic curve is determined by $\deg D$), for an elliptic curve we can assume that $D = D_n := n(0_E)$.

Theorem 3.6. *Let A be an abelian variety, D a divisor, and $\Phi_D : A \to \widehat{A}$ the associated morphism. The biextension X_D of $A \times A$ by $\mathbb{G}_m$ associated to D can be described as follows.*

Its elements are tuples $(P, Q, g_{P,Q})$ such that the function $g_{P,Q}$ on A has for divisor $t_P^* D_Q - D_Q = D_{P+Q} - D_P - D_Q$, where $D_Q = \Phi_D(Q) = t_Q^* D - D$. The projection $\pi : X_D \to A \times A$ sends $(P, Q, g_{P,Q})$ to (P, Q). We will often drop (P, Q) when referring to a biextension element $g_{P,Q} \in X_D$.

The divisors determine $g_{P,Q}$ up to some invertible constants, so the preimage of π is indeed a torsor under $\mathbb{G}_m$. Since the divisor is invariant under permutation by P, Q, the function $g_{P,Q}$ can also be interpreted as an element $\iota(g)_{Q,P}$ above (Q, P).

Fixing Q, the group law on $\pi_2^{-1}(Q)$ is equal to the group law (see Equations (1) and (6)) on the theta group $G(D_Q)$. Fixing P, the group law on $\pi_1^{-1}(P)$ is equal to the one defined in Equation (7) via Equations (3) and (4).

Explicitly we have, if g_{Q_1,Q_2} is any function with divisor $D_{Q_1+Q_2} - D_{Q_1} - D_{Q_2}$:

$$g_{P_1,Q} \star_1 g_{P_2,Q} = g_{P_1+P_2,Q} := g_{P_1,Q}(\cdot) g_{P_2,Q}(\cdot + P_1)) \tag{9}$$

$$g_{P,Q_1} \star_2 g_{P,Q_2} = g_{P,Q_1+Q_2} := g_{P,Q_1}(\cdot) g_{P,Q_2}(\cdot) \frac{g_{Q_1,Q_2}(\cdot + P)}{g_{Q_1,Q_2}(\cdot)}. \tag{10}$$

And since X_D is a symmetric biextension:

$$g_{P_1,Q} \star_1 g_{P_2,Q} = \iota(g)_{Q,P_1} \star_2 \iota(g)_{Q,P_2}, \tag{11}$$

therefore we also have:

$$g_{P_1,Q} \star_1 g_{P_1,Q} = g_{P_1+P_2,Q} = g_{P_1,Q}(\cdot) g_{P_2,Q}(\cdot) \frac{g_{P_1,P_2}(\cdot + Q)}{g_{P_1,P_2}(\cdot)}$$

Proof. We saw in Section 2.3 that Equations (6) and (7) are still valid for an abelian variety.

We now check that the formulas in Theorem 3.6 define a structure of biextension on X_n. By definition of the theta groups, the group structures on $\pi_1^{-1}(P)$ and $\pi_2^{-1}(Q)$ we use do give extensions of A by $\mathbb{G}_m$ as expected. An immediate computation also shows that the partial laws $\star_1, \star_2$ satisfy the compatibility relations of Equation (8), so X_D is a biextension.

By Theorem 3.1, X_D is the biextension associated to some morphism $\Phi : A \to \hat{A}$. We let to the reader the fun diagram chasing exercice to unravel the definitions of [Gro72, p. VII] and check that this Φ is the polarisation Φ_D we started with. (This diagram chasing becomes easier when using the fact that the symmetric biextension X_D is associated to the explicit cubical structure on (the line bundle associated to) D, see [Bre83; Mor85, Chapitre 1]. Alternatively, it is shown in [Gro72, VIII § 2] that the pairing associated to the biextension associated to Φ is the Weil pairing associated to Φ, and we will see in Theorem 3.11 that the pairing associated to our X_D is Φ_D, so $\Phi = \Phi_D$.)

Equation (11) follows from Proposition 3.4. □

Example 3.7. The inversion $g_{P,Q}^{\star_1,-1}$ is given by

$$g_{-P,Q} = \frac{1}{g_{P,Q}} \frac{g_{P,-P}}{g_{P,-P}(\cdot + Q)}.$$

Remark 3.8. The biextension X_D only depends on the polarisation Φ_D, so if D is ample only on its algebraic equivalence class. Explicit isomorphisms can be given as follows: if $D_1 \sim D_2$ and α is any function with divisor $D_2 - D_1$, then $t_Q^*\alpha/\alpha$ has for divisor $D_{2,Q} - D_{1,Q}$ hence by Equation (4) the isomorphism is given by

$$(P, g_{P,Q}) \in G(D_{1,Q}) \subset X_{D_1} \mapsto (P, g_P(\cdot)\frac{\alpha(\cdot + P + Q)\alpha(\cdot)}{\alpha(\cdot + P)\alpha(\cdot + Q)}) \in G(D_{2,Q}) \subset X_{D_2}.$$

And if $D_2 = t_c^* D_1$, then $D_{2,Q} = t_c^* D_{1,Q}$ so by Equation (5) the isomorphism is:

$$(P, g_{P,Q}) \in G(D_{1,Q}) \subset X_{D_1} \mapsto (P, t_c^* g_P) \in G(D_{2,Q}) \subset X_{D_2}.$$

3.2. **Monodromy and pairings in biextensions.** In Section 2, we saw how the exponentiation in theta groups $G(D_Q)$ naturally gave rise to the Weil and Tate pairings on an elliptic curve E or abelian variety, but we often had to juggle and switch between different theta groups. In Section 3 and Theorem 3.6, we saw that the biextension X associated to the divisor (0_E) is a convenient way to package all the theta groups $G(D_Q)$ together. We will now, following [Gro72, p. VIII] and [Sta08, Theorem 17.1.1], reinterpret the Weil and Tate pairings as monodromy information on the biextension.

We first look at the case of an elliptic cruve E. The monodromy is as follows: let $P \in E[\ell]$ and $g_{P,Q} \in X$ be an element in the biextension associated to (0_E), $g_{P,Q}$ is a function with divisor $D_{P+Q} - D_P - D_Q = (-P - Q) + (0_E) - (-P) - (-Q)-$ by Theorem 3.6. Since $\ell P = 0_E$, we have $g_{P,Q}^{\star_1,\ell}$ is an element above $(0_E, Q)$, so a function with trivial divisor, so a constant λ_P. However, even through P is of order ℓ, $g_{P,Q}$ need not be, so we may have $\lambda_P \neq 1$. We do have $g_{P,Q}^{\star_1,\ell m} = \lambda_P^m$, so if $k = \mathbb{F}_q$ is a finite field and $g_{P,Q}$ is rational, $g_{P,Q}$ is at most of order $\ell(q-1)$. We call λ_P the monodromy associated to $g_{P,Q}$ (beware of the notation, it also depends on Q).

We remark that changing $g_{P,Q}$ to $\mu g_{P,Q}$, we have $(\mu g_{P,Q})^{\star_1,\ell} = \mu^\ell \lambda_P$, so the class of λ_P in $k^*/k^{*,\ell}$ only depends on (P, Q), not on $g_{P,Q}$. Also, if $k = \mathbb{F}_q$ and $\ell \mid q-1, \lambda_P^{(q-1)/\ell} = g_{P,Q}^{\star_1,q-1}$ is a ℓ-th root of unity which does not depend on the choice of the (rational) $g_{P,Q}$ but only on (P, Q).

As expected, this monodromy λ_P will give pairings. We will also see in Section 3.4 how the Ate and optimal Ate pairings can also be interpreted as monodromy associated to endomorphisms of the form $\sum c_i \pi_q^i$.

In this article, we will need to be able to compute pairings associated with non principal polarisation. Reusing the notations of Section 2.3, if D is a divisor on A, X_D the associated biextension, and $g_{P,Q} \in X_D$ where $P \in A[\ell D]$, we may not have $\ell P = 0$, so $g_{P,Q}^\ell$ may not be a constant. We had the same problem for Theorem 2.9, and we will use the same solution.

Recall that a biextension element $g_{P,Q}$ has for divisor $D_{P+Q} + D_0 - D_P - D_Q$, which is principal because it is associated to the cycle $(P+Q) + (0) - (P) - (0)$ (with our conventions $D_0 = 0$ so we often omit it). But when $P \in A[D]$, D_P is already principal, so we may take an associated function g_P (we remark that (P, g_P) is an element of the theta group $G(D)$). The function $g_P(\cdot + Q)/g_P$ has for divisor $D_{P+Q} - D_P - D_Q$, so is an element of X_D above (P, Q). We remark that it does not depend on our choice of g_P.

Lemma 3.9. *Fix $Q \in A$. We let $(X_{D,Q}, \star_1)$ be the group of all biextension elements above (P', Q) for some P'. The map $s_Q : A[D] \to X_{D,Q}, P \mapsto s_{P,Q} := g_P(\cdot + Q)/g_P$ is a group morphism. This induces a group action $P \cdot g_{P',Q} := s_Q(P) \star_1 g_{P',Q}$ of $A[D]$ on X_D.*

Proof. $\frac{g_{P_1}(\cdot+Q)}{g_{P_1}} \star_1 g_{P_2}(\cdot+Q)g_{P_2} = \frac{g_{P_1}(\cdot+Q)g_{P_2}(\cdot+P_1+Q)}{g_{P_1}g_{P_2}(\cdot+P_1)} = \frac{g_{P_1+P_2}(\cdot+Q)}{g_{P_1+P_2}}$ where $g_{P_1+P_2} = g_{P_1}g_{P_2}(\cdot+P_1)$ is the element of $G(D)$ above P_1+P_2 coming from the composition $(P_1, g_{P_1}) \cdot (P_2, g_{P_2})$. □

By Theorem 3.6, the reformulation of Porism 2.10 in terms of biextension is:

Porism 3.10. *Let $g_{P,Q} \in X_D$ and let $g_{\ell P,Q} = g_{P,Q}^{\star_1,\ell}$. Then the function $f_{\ell,P}$ evaluated on the cycle $(x+Q)-(x)$ is given by $\frac{g_{\ell P,Q}}{g_{P,Q}^{\ell}}$.*

If furthermore $P \in A[\ell D]$, $f_{-\ell D_P}((R+Q)-(R)) = \frac{(-\ell P)\cdot g_{\ell P,Q}}{g_{P,Q}^{\ell}}(R)$.

Theorem 3.11 (Monodromy pairings)**.** *Let X_D be the biextension associated to a divisor D on an abelian variety A. Let $(P, Q, g_{P,Q}) \in X_D$ be a biextension element above (P, Q). If $P \in A[\ell D]$, we let $g_{\ell P,Q} = g_{P,Q}^{\star_1,\ell}$ is above $(\ell P, Q)$. Furthermore, $\ell P \in A[D]$, so by Lemma 3.9 there is a canonical biextension element $(\ell P, Q, s_{\ell P,Q})$ above $(\ell P, Q)$. The element $g_{\ell P,Q} \star_1 s_{\ell P,Q}^{\star_1,-1} = g_{\ell P,Q} \star_1 s_{-\ell P,Q} = (-\ell P) \cdot g_{\ell P,Q}$ is above $(0, Q)$ so is a constant λ_P. We say that λ_P is the ℓ-monodromy associated to $(P, Q, g_{P,Q})$.*

If $Q \in A[\ell D]$, the Weil pairing is given (up to a sign) by: $e_{W,D,\ell}(P, Q) = \lambda_P/\lambda_Q$

If $P \in A[\ell D](k)$, and $(P, Q, g_{P,Q}) \in X_D(k)$, i.e. $g_{P,Q}$ is chosen to be rational, then the non reduced Tate pairing is given by: $e_{T,D,\ell}(P, Q) = \lambda_P$.

Proof. This is a translation of Theorem 2.9 in terms of the biextension formulas from Theorem 3.6. □

The exact same remark as in Remark 2.11 applies for the extended Tate pairing computed through biextensions. If $D = mD_1$ with D_1 associated to a principal polarisation, and $P \in A[mD_1]$, and we want to compute the Tate pairing as an element of $k^*/k^{*,\ell m}$, we need to keep track of the function $g_{P,Q}^{\ell}$. There is a tensor map $X_{D_1} \to X_D, g_{P,Q} \mapsto g_{P,Q}^m$, and if our starting biextension element $g_{P,Q}$ is in the image of this map (on a rational element), then $g_{P,Q}^{\ell}(x)$ lies in $k^{*,\ell m}$ for x rational so we can express the extended Tate pairing purely in terms of the monodromy λ_P. But we stress that one needs to be careful that, starting with a general biextension element $g_{P,Q}$ (which won't be a m-fold tensor in general), we cannot compute the Tate pairing for $\ell m D_1$ while working on the biextension X_D purely from monodromy information, we need to keep track of a corrective factor.

Corollary 3.12. *Let X_Θ be the biextension associated to an ample divisor Θ on an abelian variety A.*

Fix any biextension element $(P, Q, g_{P,Q})$ above P, Q, we can also see it as an element of the group $(X_{\Theta,Q}, \star_1)$. If $P \in A[\ell]$, the exponentiation $g_{P,Q}^{\star_1,\ell}$ is a constant λ_P, which is the ℓ-monodromy associated to $(P, Q, g_{P,Q})$. Alternatively, we have $g_{P,Q}^{\star_1,\ell+1} = \lambda_P g_{P,Q}$.

If $P, Q \in A[\ell]$, the Weil pairing $e_{W,\Theta,\ell}$ is (up to a sign) λ_P/λ_Q. If $P \in A[\ell](k)$, assuming that $g_{P,Q}$ is rational, the non reduced Tate pairing is (up to a sign) $e_{T,\Theta,\ell} = \lambda_P$. If $k = \mathbb{F}_q$ is a finite field and $\mu_\ell \subset \mathbb{F}_q$, we can also define λ'_P as the constant $g_{P,Q}^{\star_1,q-1}$. Alternatively, we have $g_{P,Q}^{\star_1,q} = \lambda'_P g_{P,Q}$. Then the reduced Tate pairing is (up to a sign) is given by λ'_P.

Proof. This is a direct application of Theorem 3.11, using the fact that if $\ell P = 0$, $s_{0,Q} = 1$. See also Corollary 2.5. □

Remark 3.13. Let $g_{P,Q}$ be as in Corollary 3.12, and λ_P the monodromy: $g_{P,Q}^{\star_1,\ell} = \lambda_P$. Then if $n_2 = m\ell + n_1, n_1, n_2, m \in \mathbb{Z}$, then $g_{P,Q}^{\star_1,n_2} = \lambda_P^m g_{P,Q}^{\star_1,n_1}$. In particular, we have $g_{P,Q}^{\star_1,\ell+1} = \lambda_P g_{P,Q}$ and $g_{P,Q}^{\star_1,\ell-1} = \lambda_P g_{P,Q}^{\star_1,-1}$. Sometimes, it is easier to compute the monodromy λ_P using these relations.

Remark 3.14 (Refined bilinearity for the Tate pairing)**.** The (non reduced) Tate pairing is only bilinear when we consider its value in $k^*/k^{*,\ell}$. However, by Remark 3.13 we have a refined version of bilinearity: assume we take $g_{P,Q}$ to compute the Tate pairing $e_\ell(P,Q)$ via the monodromy relation $g_{P,Q}^{\star_1,\ell} = \lambda_P \in k^*$.

Now assume that to compute the Tate pairing $e_\ell(iP,Q)$, we take $g_{iP,Q} = g_{P,Q}^{\star_1,i}$ rather than an arbitrary element. Then we have $g_{iP,Q}^{\star_1,\ell} = g_{P,Q}^{\star_1,i\ell} = \lambda_P^i \in k^*$.

Likewise, for the Tate pairing $e_\ell(P,iQ)$, if we take $g_{P,iQ} = g_{P,Q}^{\star_2,i}$, we have $g_{P,iQ}^{\star_1,\ell} = (g_{P,Q}^{\star_1,\ell})^{\star_2,i} = \lambda_P^i \in k^*$ by the compatibility between $\star_1$ and $\star_2$.

Of course, there is a priori no canonical choice of $g_{iP,Q}$ for all i, depending only on iP, such that $g_{iP,Q} = g_{P,Q}^{\star_1,i}$ (at least without computing a DLP of iP with respect to P), so no way to exploit this refined bilinearity. But see Remark 4.20 for a partial choice.

3.3. **The arithmetic of biextensions on elliptic curves: the evaluation representation.** To exploit Theorem 3.11 for computing the Weil and Tate pairings, we need to develop efficient arithmetic on biextensions. In particular, by Corollary 3.12, to compute the polarised Weil and Tate pairing efficiently on an abelian variety (A,Θ), we need a fast exponentiation in the group $(X_{\Theta,Q},\star_1)$ induced by the biextension X_Θ. In particular we can apply all well known techniques for group exponentiation: double and add, windowing, slidings windows, NAF, combings…These tools are of course well known in the pairing literature. But beware that the context is different for pairings than for scalar multiplication: in the ECC context, the same base point P is multiplied by different scalars, whereas in the pairing context different biextension elements $g_{P,Q}$ are multiplied by the same scalar ℓ.

For simplicity, we go back to the case of elliptic curves, but as usual everything holds for general abelian varieties. It remains to do the basic group operations, using $\star_1$ or $\star_2$ since it gives the same result by Proposition 3.4. For an elliptic curve E, taking $D = (0_E)$ the canonical principal polarisation, an element $g_{P,Q}$ of the biextension X_{0_E}, since it is a function with divisor $D_{P+Q} - D_P - D_Q = (-P-Q) + (0) - (-P) - (-Q)$ is (in the generic case) of the form

$$g_{P,Q}(x,y) = c\frac{x - x_{P+Q}}{y - y_P - \alpha(x - x_P)} \tag{12}$$

where α is the slope of the line $l_{P,Q}$ going through P and Q. To compute the biextension law $g_{P_1,Q} \star_1 g_{P_2,Q}$ it suffices to plug in the formula Equation (10) (using the elliptic group law), and then reduce modulo the elliptic curve equation to obtain an equation of $g_{P_1+P_2,Q}$ of the form above. However, this is not very efficient. We will instead try to find a more efficient representation of biextension elements.

First, we remark that $g_{P,Q}$ is completely determined, up to a constant, from (P,Q) (which gives its divisor $D_{P+Q} - D_P - D_Q$. Using the full function form of $g_{P,Q}$ to determine this function is thus overkill; a more efficient representation is to simply use its evaluation

$g_{P,Q}(R_0)$ at some base point R_0 The biextension element is then represented by $(P, Q, c := g_{P,Q}(R_0))$, we call this the evaluation representation.

Note that if R_0 is a pole or zero of R_0, we can use the standard trick of fixing a uniformizer π_{R_0} at R_0, and "defining" $g_{P,Q}(R_0)$ to be the first coefficient of the Laurent series expansion of $g_{P,Q}$ along π_{R_0} (we called this the extended value oin Remark 2.8).

It is customary to take $R_0 = 0_E$; if

$$\mu_{P,Q}(x, y) = \frac{x - x_{P+Q}}{y - y_P - \alpha(x - x_P)}$$

is the "usual" (recall that we use a somewhat non standard sign convention, see Remark 2.2) normalised Miller function with divisor $(-P-Q)+(0_E)-(-P)-(-Q)$ and $g_{P,Q}$ is as in Equation (12), then $g_{P,Q} = c\mu_{P,Q}$ so $g_{P,Q}(0_E) = c$ (the extended value for the uniformiser $\pi_{0_E} = x/y$).

The biextension formulas (using either law $\star = \star_1, \star_2$) then gives:

$$(P_1, Q, c_1) \star (P_2, Q, c_2) = c_1 c_2 \frac{g_{P_1,P_2}(R_0 + Q)}{g_{P_1,P_2}(R_0)} \tag{13}$$

from which it follows that

$$g_{P,Q}^{\star_1,\ell} = g_{P,Q}(R_0)^{\ell} f_{\ell,P}((R_0 + Q) - (R_0)) \tag{14}$$

where $\operatorname{div} f_{\ell,P} = D_{\ell P} - \ell D_P = (-\ell P) + (\ell - 1)(0_E) - \ell(P)$.

Thus, the biextension arithmetic and exponentiation in the evaluation representation gives exactly the usual Miller algorithm, modulo our different sign conventions.

Going through all the theory of biextensions only to recover the standard Miller algorithm might seem overkill. We will be rewarded in later sections when using other biextension representations.

There are still some useful information we can glean from the biextension interpretation of Miller's algorithm. First, as mentioned above, it is well known in the pairing literature that Miller's formula form a group law, to which we can apply the standard group exponentiation algorithms. The biextension gives a geometric interpretation of this group law. In particular, it gives a geometric interpretation of the various relations on the functions $f_{\ell,P}$ used to define the ate and optimal ate pairings (we will go back to this in Section 3.4). For instance, by Corollary 3.12 the reduced Tate pairing is given by $f_{q-1,P}((R_0 + Q) - (R_0))$; of course since the field arithmetic is faster than the biextension arithmetic, it is more efficient to first compute the reduced Tate pairing via $f_{\ell,P}((R_0 + Q) - (R_0))$ and then proceed via the final exponentiation by field arithmetic (equivalently: working on the biextension over $(0, Q)$).

Secondly, it shows that the *values* $f_{\ell,P}((R_0 + Q) - (R_0))$ we compute during Miller's algorithm are simply a convenient representations of the *functions* $g_{\ell P,Q}$ coming from the biextension. During the execution of Miller's algorithm, it can happen that we need to evaluate our intermediate Miller functions on a pole or zero. The standard solution is to use a uniformiser, explicit formulas are given in [Rob21a, Lemma 3.5.3]. Another solution is to change the evaluation point R_0 to a new point R_0'. But rather than restarting Miller's algorithm from scratch, we can change the evaluation point *on the fly*, by going from the evaluation representation on R_0 back to the function representation back to the evaluation representation on R_0'. Explicitly, if we have the representation $(P, Q, g_{P,Q}(R_0))$, we can compute any function (eg the normalised one) $\mu_{P,Q}$ with divisor $D_{P+Q} - D_P - D_Q$, and then use $g_{P,Q}(R_0') = \mu_{P,Q}(R_0')g_{P,Q}(R_0)/\mu_{P,Q}(R_0)$.

Thirdly, the symmetry relation $g_{P_1,Q} \star_1 g_{P_2,Q} = g_{Q,P_1} \star_2 g_{Q,P_2}$ from Proposition 3.4 gives the following relation on the normalised functions $\mu_{P,Q}$:

Lemma 3.15. $\mu_{P_1,P_2}(P_3) = \mu_{P_2,P_3}(P_1) = \mu_{P_3,P_1}(P_2)$.

Proof. By Theorem 3.6, we have $\mu_{P_1,Q} \star_1 \mu_{P_2,Q} = \mu_{P_1,Q}(\cdot)\mu_{P_2,Q}(\cdot + P_1))$, while $\mu_{P,Q_1} \star_2 \mu_{P_2,Q} = \mu_{P_1,Q}(\cdot)\mu_{P_2,Q}(\cdot)\frac{\mu_{P_1,P_2}(\cdot+Q)}{\mu_{P_1,P_2}(\cdot)}$. It follows that $\mu_{P_2,Q}\mu_{P_1,P_2}(\cdot + Q) = \mu_{P_2,Q}(\cdot + P_1)\mu_{P_1,P_2}$. Evaluating this equaliting on 0_E (multiplying both members by the appropriate uniformiser to that the evaluation is well defined) gives $\mu_{P_1,P_2}(Q) = \mu_{P_2,Q}(P_1)$, from which the lemma follows by a change of variable. □

Remark 3.16. Recall from Remark 2.2 that we use a different sign convention than usual.

In this remark only, we go back to the standard sign convention, and let $\boldsymbol{\mu}_{P,Q}$ be the standard normalised Miller function with divisor $(P) + (Q) - (P + Q) - (0_E)$, and $\mathbf{f}_{\ell,P}$ the standard normalised Miller function with divisor $\ell(P) - (\ell P) - (\ell - 1)(0_E)$. Then Lemma 3.15 becomes on these standard functions: $\boldsymbol{\mu}_{P_1,P_2}(-P_3) = \boldsymbol{\mu}_{P_2,P_3}(-P_1) = \boldsymbol{\mu}_{P_3,P_1}(-P_2)$.

We leave as an exercice to the reader to prove this fact (and its generalisation to abelian varieties) without the theory of biextensions.

This gives the following interesting tweaks on Miller's algorithm: for the Miller addition, we have $\mathbf{f}_{\ell+1,P}(Q) = \mathbf{f}_{\ell,P}(Q)\boldsymbol{\mu}_{\ell P,P}(Q) = \mathbf{f}_{\ell,P}(Q)\boldsymbol{\mu}_{P,-Q}(-\ell P)$. In other words, rather than evaluating the different Miller functions $\boldsymbol{\mu}_{\ell P,P}$ on the same point, we could evaluate the same (precomputed) Miller function $\boldsymbol{\mu}_{P,-Q}$ on the different points $-\ell P$.

Likewise, for the Miller doubling: $\mathbf{f}_{\ell+\ell,P}(Q) = \mathbf{f}_{\ell,P}(Q)^2\boldsymbol{\mu}_{\ell P,\ell P}(Q) = \mathbf{f}_{\ell,P}(Q)^2\boldsymbol{\mu}_{\ell P,-Q}(-\ell P)$. The numerator of $\boldsymbol{\mu}_{\ell P,-Q}$ is $y - y(\ell P) - \alpha(x - x(\ell P))$, which evaluated on $-\ell P$ is given by the simple formula $-2y(\ell P)$. However, the evaluated denominator is $x(\ell P) - x(\ell P - Q)$, which requires to compute $\ell P - Q$. This will be a recurring theme in our latter algorithm than in our pairing algorithms we will compute arithmetic informations both from ℓP and $\pm Q + \ell P$.

We remark that faster formulas for the standard Miller's algorithm have been obtained by slightly tweaking the Miller functions. In [DZZZ22] the authors introduce functions with divisors $\ell(P) + (-\ell P) - (\ell + 1)(0)$ which give a streamlined double and add formula. In [BELL10], the following formula is used: $\mathbf{f}_{\ell_1+\ell_2,P} = \frac{1}{\mathbf{f}_{-\ell_1,P}\mathbf{f}_{-\ell_2,P}l_{-\ell_1 P,-\ell_2 P}}$, instead of the standard formula: $\mathbf{f}_{\ell_1+\ell_2,P} = \mathbf{f}_{\ell_1,P}\mathbf{f}_{\ell_2,P}\mu_{\ell_1 P,\ell_2 P}$. We let open the question of whether combining these tweaks with Lemma 3.15 could give further speed ups.

3.4. **The Ate and optimal Ate pairings as monodromy pairings.** We also can give a monodromy interpretation of the Ate pairing.

If X_D is a biextension associated to an abelian variety (A, D) over a field k, and k/k_0 is a Galoisian extension, the Galoisian action can be be described as follows. Let $(P, Q, g_{P,Q}) \in X_D$ and $\sigma \in \mathrm{Gal}(k/k_0)$, then $\sigma(P, Q, g_{P,Q}) = (\sigma(P), \sigma(Q), g^{\sigma}_{P,Q})$ where $g^{\sigma}_{P,Q}$ is the function with divisor $D_{\sigma(P+Q)} - D_{\sigma(P)} - D_{\sigma(Q)}$ such that $g^{\sigma}_{P,Q}(\sigma(R)) = \sigma(g_{P,Q}(R))$.

Now assume for simplicity that (A, D) is an elliptic curve $(E, (0_E))$ over a finite field $\mathbb{F}_q$; as usual our formulas will still be valid for a general abelian variety over $\mathbb{F}_q$ (for a definition of the Ate and optimal Ate pairings for abelian varieties, see [LR15]). For the Ate pairing situation, we assume that $E[\ell](\mathbb{F}_q)$ is not empty but that the embedding degree d is greater than 1, thus the full ℓ-torsion is defined over $\mathbb{F}_{q^d}$, and $E[\ell] = \mathbb{G}_1 \oplus \mathbb{G}_2$ where $\mathbb{G}_1 = E[\ell](\mathbb{F}_q)$ is the eigenspace of π_q for its eigenvalue 1, and $\mathbb{G}_2 = \{Q \in E[\ell](\mathbb{F}_q) \mid \pi_q Q = qQ\}$ is the eigenspace for the eigenvalue q.

In pairing based cryptography, we consider the (non reduced) Tate pairing as restricted to $\mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{F}^*_{q^d}/\mathbb{F}^{*,\ell}_{q^d}$, while the (reduced) Ate pairing $a_{\lambda,\ell} : \mathbb{G}_2 \times \mathbb{G}_1 \to \mathbb{F}^*_{q^d}/\mathbb{F}^{*,\ell}_{q^d}$ is given by $a_{\lambda,\ell}(P, Q) = f_{\lambda,P}((Q) - (0))$ for any $\lambda \equiv q \mod \ell$. In the special case when

$\lambda = q$, the Ate pairing $a_{T,q}$ is already reduced, otherwise the reduced Ate pairing is given by $f_{\lambda,P}((Q)-(0))^{(q^d-1)/\ell} \in \mu_\ell$.

Now let $g = (P, Q, g_{P,Q})$ be a biextension element with $P \in \mathbb{G}_2$ and $Q \in \mathbb{G}_1$. Then $\pi_q(g) = (qP, Q, g_{P,Q}^{\pi_q})$ and $g^{\star_{1,}\lambda}$ are both biextension elements above (qP, Q). They need not be equal, and in fact the monodromy between them is precisely the λ-Ate pairing.

Proposition 3.17. *Let $P \in \mathbb{G}_2, Q \in \mathbb{G}_1$, $g_{P,Q}$ any biextension element in $\mathbb{F}_{q^d}$ above (P,Q), and let c be the monodromy such that $g_{P,Q}^{\star_{1,}\lambda} = c\pi_q(g_{P,Q})$. Then c gives the λ-Ate pairing: $a_{\lambda,\ell}(P,Q) = c$.*

Proof. Immediate from the definitions and Porism 3.10. □

Remark 3.18 (Optimal Ate)**.** Write $\ell = \sum c_i q^i$. Then rather than computing $g_{P,Q}^{\star_{1,}\ell}$ as $\prod_{\star_{1,}i} \left(g_{P,Q}^{\star_{1,}c_i}\right)^{\star_{1,}q^i} = e_{T,\ell}(P,Q)$ (the non reduced Tate pairing), we can compute $\prod_{\star_{1,}i}(\pi_q^i)(g_{P,Q}^{\star_{1,}c_i}) = C$. By Proposition 3.17, this will differ from the Tate pairing by a bunch of Ate pairings; and by Porism 3.10 this is exactly the optimal Ate pairing.

We leave to the reader the monodromy interpretation of the twisted Ate and the Eil pairing when given an automorphism α of order dividing d.

Corollary 3.19 (Explicit formulas)**.** *Using the evaluation representation $(P, Q, g_{P,Q}(R_0))$ of biextension elements, the Ate pairing on $\mathbb{G}_2 \times \mathbb{G}_1$ is computed as follows: if $g_{P,Q}$ is represented by (P,Q,c), and $g_{P,Q}^{\star_{1,}\lambda} = (qP, Q, c')$ then $a_{\lambda,\ell}(P,Q) = c'/c^q$.*

If $\ell = \sum a_i q^i$, and $g_{P,Q}^{\star_{1,}a_i} = (a_iP, Q, c_i)$, then the optimal Ate pairing is given by the constant $\prod_{\star_{1,}i}(q^i a_i P, Q, c_i^{q^i})$.

Remark 3.20 (The reduced Ate and Tate pairings)**.** When computing the λ-ate pairing, changing the representative $g_{P,Q}$ by $u \cdot g_{P,Q}$ for some $u \in \overline{\mathbb{F}}_q^*$ changes the value of the λ-Ate pairing by $g_{P,Q}^{\star_{1,}\lambda}/\pi_q(g_{P,Q}) = u^{\lambda-q}$. Hence we recover the fact that the q-Ate pairing is already reduced, and does not depends on the choice of representative.

As an aside, since $\mu_\ell \subset \mathbb{F}_{q^d}$, we saw that the reduced Tate pairing was given by $g_{P,Q}^{\star_{1,}q^d}/g_{P,Q}$ for any $g_{P,Q}$ defined over $\mathbb{F}_{q^d}$. The true definition of the reduced Tate pairing should actually be as the monodromy $g_{P,Q}^{\star_{1,}q^d}/\pi_{q^d}(g_{P,Q})$. In this case, it does not depends on the choice of representative for $g_{P,Q}$ either, even a non rational one, and reduces to the definition given above when $g_{P,Q}$ is defined over $\mathbb{F}_{q^d}$.

Remark 3.21 (The Ate and Tate pairings as Weil-Cartier pairings)**.** It is well known that the reduced Tate pairing $e_{T,\ell}(P,Q)$ is induced by the Weil-Cartier pairing $e_{\pi_q^d-1} : E[\pi_q^d - 1] \times E[\hat{\pi}_q{}^d - 1] \to \mathbb{G}_m$. Furthermore, $\hat{\pi}_q{}^d = \pi_q^d$ when restricted to $E[\ell]$, by definition of the embedding degree.

It is maybe less known, but the q-Ate pairing can also be interpreted as a Weil-Cartier pairing:

$$a_{q,\ell}(P,Q) = e_{\pi_q-1}(P,Q)^q = e_{\pi_q-1}(P,qQ) = e_{\pi_q-1}(\pi_q P, \pi_q Q). \tag{15}$$

We remark that by definition, $\mathbb{G}_1 \subset \operatorname{Ker}\pi_q - 1$ and $\mathbb{G}_2 \subset \operatorname{Ker}\hat{\pi}_q - 1$. Equation (15) can be proven using the relationship between the Ate and Tate pairings (see [Ver10]), and the relationship between the Weil-Cartier pairings of $\pi_q - 1$ and $\pi_q^d - 1 = (\pi_q - 1)(1 + \cdots + \pi_q^{d-1})$ coming from the compatibility of the Weil-Cartier pairings with isogenies.

We will use this to give a monodromy interpretation of the Weil-Cartier pairing associated to $\pi_q^d - 1$ and $\pi_q - 1$.

First as a warm-up, let us recall the monodromy interpretation of the Weil pairing: we take $P, Q \in E[\ell]$, a $g_{P,Q}$ in the biextension above them, and compute the monodromy $g_{P,Q}^{\star_1,\ell} = c_1 \cdot g_{P,Q}$, $g_{P,Q}^{\star_2,\ell} = c_2 \cdot g_{P,Q}$ to get the Weil pairing $e_{W,\ell}(P,Q) = c_1/c_2$. An alternative way is to compute the monodromy $(g_{P,Q}^{\star_1,\ell})^{\star_2,\ell} = c(g_{P,Q}^{\star_2,\ell})^{\star_1,\ell}$, a quick calculation gives $c = c_1/c_2(c_2/c_1)^\ell = c_1/c_2$ (using that $e_{W,\ell}(P,Q) \in \mu_\ell$) so this also gives the Weil pairing (and we recover the commutator interpretation of Example 2.1).

For the α-Weil-Cartier pairing, α an endomorphism, to compute $e_\alpha(P,Q)$ when $P \in \operatorname{Ker}\alpha$ and $Q \in \operatorname{Ker}\hat{\alpha}$, it is natural to see if a possible strategy to compute e_α is to take the quotient of the monodromy of the action of α on $g_{P,Q}$ (any biextension element above (P,Q)) with respect to $\star_1$, and the action of $\hat{\alpha}$ on $g_{P,Q}$ with respect to $\star_2$. Using Proposition 3.17 and the interpretation of the reduced Tate and Ate pairings as Weil-Cartier pairings above, we'll see that an analogous strategy does hold for $\alpha = \pi_q - 1$ and $\alpha = \pi_q^d - 1$..

We first start with the Weil-Cartier pairing for $\pi_q^d - 1$. We first want to compute the action of $\pi_q^d - 1$ on $g_{P,Q}$ with respect to $\star_1$ and compare it with 1; it will be easier to compute π_q^d on $g_{P,Q}$ and compare it with $g_{P,Q}$: $\pi_q^d(g_{P,Q}) = c_1 \cdot g_{P,Q}$. Likewise, rather than computing the action of $\hat{\pi}_q^d - 1$ on $g_{P,Q}$ with respect to $\star_2$ and compare it with 1, we will compute the action of $\hat{\pi}_q^d$ on $g_{P,Q}$ and compare it with $g_{P,Q}$. Now $\hat{\pi}_q^d = q^d\pi_q^{-d}$. We have $\pi_q^{-d}(g_{P,Q}) = 1/\pi_q^{-d}(c_1) \cdot g_{P,Q}$, and $(q^d\pi_q^{-d}) \cdot_{\star_2} (g_{P,Q}) = (1/\pi_q^{-d}(c_1))^{q^d} g_{P,Q}^{\star_2,q^d} = 1/c_1 g_{P,Q}^{\star_2,q^d}$. So the monodromy quotient is given by $g_{P,Q}^{\star_2,q^d}/\pi_q^d(g_{P,Q})$ which is precisely the formula from Remark 3.20 for the reduced Tate pairing. Hence we do have a correct formula for the Weil-Cartier pairing of $\pi_q^d - 1$.

Now we try to find a monodromy approach to compute the Weil-Cartier pairing for $\pi_q - 1$ on $\pi_q(P), \pi_q(Q)$, with $\pi_q(P) \in \mathbb{G}_1, \pi_q(Q) \in \mathbb{G}_2$ (beware that we switched arguments compared to Proposition 3.17). We can assume that we are given an element of the form $g_{\pi(P),\pi(Q)} = \pi_q(g_{P,Q})$ above $(\pi_q(P), \pi_q(Q))$. We want to first compute the quotient of the action (with respect to $\star_2$) of $\hat{\pi}_q = q\pi_q^{-1}$ on $\pi_q(g_{P,Q})$ and $\pi_q(g_{P,Q})$. By the same computation as above, the result is precisely the monodromy $g_{P,Q}^{\star_2,q}/\pi_q(g_{P,Q})$, i.e., the q-Ate pairing (using the symmetry $\star_2 = \star_1$ and the fact that we switched the side of the arguments). Strangely, in this case we do not need to compute a monodromy action of π_q on $g_{\pi(P),\pi(Q)}$ with respect to $\star_1$ and compare it to $g_{\pi(P),\pi(Q)}$.

Remark 3.22 (Twists and automorphisms)**.** Let $\psi : A' \to A$ be an isomorphism, and let $D' = \psi^*D$. Then we have an isomorphism of biextensions $X_D \to X_{D'}, g_{P,Q} \mapsto \psi^* g_{P,Q} = g_{P,Q} \circ \psi = g_{P',Q'}$ where $P' = \psi^{-1}(P)$ and $Q' = \psi^{-1}(Q)$.

In particular, any monodromy information computed from $g_{P,Q}$ via the biextension arithmetic on X_D, can also be recovered via $g_{P',Q'}$ and the biextension arithmetic on $X_{D'}$. This generalises [CLN10] to abelian varieties.

Indeed, in that article, the authors explain that if $E/\mathbb{F}_q$ is an elliptic curve admitting a twist of degree f, and we want to do pairings over $E/\mathbb{F}_{q^d}$ with $f \mid d$, then we can consider the twist $E'/\mathbb{F}_{q^{d/f}}$ of $E/\mathbb{F}_{q^{d/f}}$, which becomes isomorphic via $\psi : E' \to E$ over $\mathbb{F}_{q^d}$. Working with P', Q' over E' rather than with P, Q over E can be helpful in term of the field of definitions of their coordinates (see the paper for more details).

A word of warning: if we start with $g_{P,Q}$ normalised with respect to some uniformiser π_{0_E} for 0_E (typically $\pi_{0_E} = \pm x/y$), then $\psi^* g_{P,Q}$ will be normalised for $\pi^*\pi_{0_E}$, which may not be $\pm x'/y'$. For instance in the situation of [CLN10, Theorem 1], we have $\pi^* x/y = \omega x'/y'$, which explains precisely the factor ω found from the explicit computations in the proof of that Theorem. The biextension point of view thus gives a more conceptual proof of that result (since we have seen above that Miller's algorithm is a particular way of computing the biextension arithmetic).

Let us give more details for the general case of an abelian variety. Recall that a twist A' of A over k corresponds to an element $\xi \in H^1(\mathrm{Gal}(\bar{k}/k), \mathrm{Aut}(A)(\bar{k}))$. If $k = \mathbb{F}_q$ is a finite field, this H^1 is isomorphic to $\mathrm{Aut}(A)(\overline{\mathbb{F}}_q)/(\pi_q - 1)$. Let us assume that $\mathrm{Aut}(A)(\overline{\mathbb{F}}_q) = \mathrm{Aut}(A)(\mathbb{F}_q)$ for simplicity, so that the H^1 is isomorphic to $\mathrm{Aut}(A)$. Take $\alpha \in \mathrm{Aut}(A)$, and let $A' = A_\alpha$ be the twist corresponding to α. Let $\phi : A \to A'$ be an isomorphism (which will be defined over the extension of $\mathbb{F}_q$ of degree e the order of α). We then have $\phi^{-1} \circ \pi_q \circ \phi = \alpha\pi_q$.

Now for the Weil pairing $e_{W,\ell}(P, Q)$, we can take any biextension element $g_{P,Q}$ above P, Q, and compute the Weil pairing through monodromy information. By the argument above, using the isomorphism $X_A \to X_{A'}, g_{P,Q} \mapsto g'_{P',Q'} = \phi^* g_{P,Q}$, we see that $e_{W,\ell}(P, Q) = e_{W,\ell}(P', Q')$ for $P' = \phi(P), Q' = \phi(Q)$, because the pullback of a constant function λ by ϕ is still λ.

For the (reduced) Tate pairing, say over $\mathbb{F}_{q^d}$, things are more subtle: we recover it from the monodromy information $g_{P,Q}^{\star_1, q^d} = \lambda$ only if we take $g_{P,Q}$ rational over $\mathbb{F}_{q^d}$. But in that case, $g'_{P',Q'}$ might not be, so the two Tate pairing could differ. However, if $e \mid d$ (as is usually the case when using twists in pairing based cryptography), then since the isomorphism $X_A \to X_{A'}$ is defined over $\mathbb{F}_{q^e}$ then $g'_{P',Q'}$ will be rational.

Another way to see that is to use Remark 3.20: the Tate pairing is given by the monodromy $g_{P,Q}^{\star_1, q^d} = \lambda\pi_{q^d}(g_{P,Q})$, where this time the result does not depend on the fact that $g_{P,Q}$ is rational. Computing the monodromy on the twist, we get ${g'_{P',Q'}}^{\star_1, q^d} = \lambda'\pi'_{q^d}(g'_{P',Q'})$, where λ' might be different from λ because of the relation $\phi^{-1} \circ \pi_q \circ \phi = \alpha\pi_q$, which implies that $\phi^{-1} \circ \pi_{q^d} \circ \phi = \alpha^d \pi_q$. We recover that when $e \mid d$, $\phi^{-1} \circ \pi'_{q^d} \circ = \pi_{q^d}$, hence $\lambda' = \lambda$. In the general case, if $[\phi]$ is the determinant action of ϕ on the differentials (i.e., its action on $\Lambda^g \Omega^1_{A/\mathbb{F}_q}$) (this depends on the choice of $[\phi]$, not only on the choice of twist A'), then $\lambda' = [\phi]^{q^d - 1}\lambda$.

The same reasoning hold for the ate pairing, where this time we use the monodromy: $g_{P,Q}^{\star_1, q} = \lambda\pi_q(g_{P,Q})$, hence the same considerations as for the Tate pairing above apply with π_q instead of π_{q^d}.

4. Cubical arithmetic

In this section, we introduce the cubical representation of biextension elements, from which we will derive efficient biextension arithmetic on Kummer lines.

We refer to [Bre83; Mor85, Chapitre 1] for the concept of cubical torsors, and notably to [Bre83, Introduction; Mor85, Chapitre 1, § 2, 3] for illuminating discussions on the relationship between squared torsor structures, theta groups, cubical torsor structures and symmetric biextensions (notably: on an abelian variety line bundles admit squared and cubical structures, the cubical structure being induced by the squared one; and theta groups correspond to the squared structures, and symmetric biextensions to the cubical structures).

We first introduce the algebraic Riemann formulas in Section 4.1, which we use in Section 4.2 to define cubical points and an arithmetic on cubical points. We show in Section 4.3 that the cubical arithmetic is a refinement of the biextension arithmetic, and can thus be used to get a representation of biextension elements. We reframe in Section 4.4 pairings in terms of cubical points. In Section 4.5 we introduce the affine lift representation of cubical point, and compare it with the evaluation representation of Section 3.3. We show in Section 4.7 that with small tweaks, these affine lifts also give a good representation of cubical point on a Kummer variety $A/\pm 1$. In Section 4.8, for a complex abelian variety $A/\mathbb{C} = \mathbb{C}^g/\Lambda$, we make explicit the link between the cubical arithmetic and the transcendental/analytic group law on $\mathbb{C}^g$. We then specialize our formulas to the case of elliptic curves in Section 4.9.

4.1. **The algebraic Riemann formulas.** Let D be an ample divisor on an abelian variety. We will assume that D is symmetric up to linear equivalence, ie $[-1]^*D \sim D$. Upon changing D in its algebraic equivalence class, which does not changes the associated polarisation, we can always assume this is the case (possibly over a field extension). Changing D in its linear equivalence class (possibly over a field extension again), we could even assume that D is symmetric.

Proposition 4.1. *Let $P_1, P_2, P_3, P_4 \in A$, $2R = P_1 + P_2 + P_3 + P_4$, and $Q_1 = R - P_1$, $Q_2 = R - P_2$, $Q_3 = R - P_3$, $Q_4 = R - P_4$ (we remark that $2R = Q_1 + Q_2 + Q_3 + Q_4$ and $P_i = R - Q_i$ so the situation is symmetric in the P_i, Q_i). There is a canonical function γ whose divisor is $D_{P_1} + D_{P_2} + D_{P_3} + D_{P_4} - D_{Q_1} - D_{Q_2} - D_{Q_3} - D_{Q_4}$.*

It is convenient to reframe this in terms of line bundles: let $\mathcal{L} = O(D)$ be the associated symmetric line bundle. Denote by $\mathcal{L}_P$ the translate $t_P^*\mathcal{L}$. Then there is a canonical isomorphism

$$\mathcal{L}_{P_1} \otimes \mathcal{L}_{P_2} \otimes \mathcal{L}_{P_3} \otimes \mathcal{L}_{P_4} \simeq \mathcal{L}_{Q_1} \otimes \mathcal{L}_{Q_2} \otimes \mathcal{L}_{Q_3} \otimes \mathcal{L}_{Q_4} \tag{16}$$

We call this isomorphism an algebraic Riemann relation, and use the notation $[P_1, P_2, P_3, P_4; Q_1, Q_2, Q_3, Q_4]$ to denote that the points P_i, Q_i are in Riemann form.

Proof. We have $D_{Q_3} + D_{Q_4} = 2D_R - D_{P_3} - D_{P4} = D_{P_1} + D_{P_2}$. Hence, there exists some (non canonical) function α with divisor $D_{P_1} + D_{P_2} - D_{Q_3} - D_{Q_4}$. Now $[-1]^*\alpha$ has for divisor $([-1]^*D)_{-P_1} + ([-1]^*D)_{-P_2} - ([-1]^*D)_{-Q_3} - ([-1]^*D)_{-Q_4}$, so $t_R^*[-1]^*\alpha = \alpha(R-(\cdot))$ has for divisor $([-1]^*D)_{R-P_1} + ([-1]^*D)_{R-P_2} - ([-1]^*D)_{R-Q_3} - ([-1]^*D)_{R-Q_4} = ([-1]^*D)_{Q_1} + ([-1]^*D)_{Q_2} - ([-1]^*D)_{P_3} - ([-1]^*D)_{P_4}$.

Since D is linearly equivalent to $[-1]^*D$, there exists some function β with divisor $[-1]^*D - D$. It follows that $\alpha(R-(\cdot))\frac{\beta(\cdot+Q_1)\beta(\cdot+Q_2)}{\beta(\cdot+P_3)\beta(\cdot+Q_4)}$ has for divisor $D_{Q_1} + D_{Q_2} - D_{P_3} - D_{P_4}$. Hence the function

$$\gamma = \frac{\alpha(\cdot)}{\alpha(R-(\cdot))}\frac{\beta(\cdot+Q_1)\beta(\cdot+Q_2)}{\beta(\cdot+P_3)\beta(\cdot+Q_4)}$$

has for divisor $D_{P_1} + D_{P_2} - D_{Q_3} - D_{Q_4} - D_{Q_1} - D_{Q_2} + D_{P_3} + D_{P_4}$ as wanted, and it does not depend on our choice of α, β.

Using the language of line bundles, the proof simplifies as follows: fix an arbitrary isomorphism $\alpha : \mathcal{L}_{P_1} \otimes \mathcal{L}_{P_2} \to \mathcal{L}_{Q_3} \otimes \mathcal{L}_{Q_4}$ and use $t_R^*[-1]^*\alpha$ along with any isomorphism $\delta : [-1]^*\mathcal{L} \simeq \mathcal{L}$ (in practice it will be convenient to use the canonical one rigidified at 0) to obtain an isomorphism $\alpha_2 : \mathcal{L}_{Q_1} \otimes \mathcal{L}_{Q_2} \to \mathcal{L}_{P_3} \otimes \mathcal{L}_{P_4}$. We then have an isomorphism $\gamma = \alpha \otimes \alpha_2^{-1} : \mathcal{L}_{P_1} \otimes \mathcal{L}_{P_2} \otimes \mathcal{L}_{P_3} \otimes \mathcal{L}_{P_4} \simeq \mathcal{L}_{Q_1} \otimes \mathcal{L}_{Q_2} \otimes \mathcal{L}_{Q_3} \otimes \mathcal{L}_{Q_4}$ which does not depend on the choice of α. □

Remark 4.2. We remark that if D is symmetric, we can take $\beta = 1$ and the function γ simplifies to $\gamma = \frac{\alpha(\cdot)}{\alpha(R-(\cdot))}$.

For instance, this is the case if $A = E$ is an elliptic curve and $D = (0_E)$; we recall that α is any function with divisor $D_{P_1} + D_{P_2} - D_{Q_3} - D_{Q_4}$. An explicit construction of α is as follows: recall that $Q_3 + Q_4 = R - P_3 + R - P_4 = P_1 + P_2$. Let g_{P_1,P_2} be any function with divisor $D_{P_1+P_2} - D_{P_1} - D_{P_2}$ and g_{Q_3,Q_4} be any function with divisor $D_{Q_3+Q_4} - D_{Q_3} - D_{Q_4}$, then $\alpha = g_{Q_3,Q_4}/g_{P_1,P_2}$ has the correct divisor, and we obtain

$$\gamma(X) = \frac{g_{Q_3,Q_4}(X)g_{P_1,P_2}(R-X)}{g_{P_1,P_2}(X)g_{Q_3,Q_4}(R-X)}.$$

We also remark that γ is constructed from α, whose existence comes from the theorem of the square. On an abelian variety a line bundle has a canonical squared structure which induces the canonical cubical structure.

Example 4.3. We have the following important squared and cubical relations (compare with [Mor85, § I.2.3] and [Mor85, I.(2.4.1)] respectively) as special cases of Proposition 4.1 and Remark 4.2:

- $[P+Q, P-Q, 0, 0; -Q, Q, P, P]$.If D is symmetric, the function γ associated to $[P+Q, P-Q, 0, 0; -Q, Q, P, P]$ is $\gamma = \frac{g_{Q,-Q}(P-(\cdot))}{g_{Q,-Q}(\cdot)}$, with $g_{Q,-Q}$ a function with divisor $D_Q + D_{-Q}$.
- $[P+Q+R, P, Q, R; 0, Q+R, P+R, P+Q]$. If D is symmetric, the function γ associated to $[P+Q+R, P, Q, R; 0, Q+R, P+R, P+Q]$ is $\gamma = \frac{g_{Q,R}(P+Q+R-(\cdot))}{g_{Q,R}(\cdot)}$ with $g_{Q,R}$ a function with divisor $D_{Q+R} - D_Q - D_R$.

A trick to find γ in practice is to remark that γ has for divisor $D_{P+Q} + D_{P-Q} - D_{-Q} - D_Q - 2D_P$ and $D_{P+Q+R} + D_P + D_Q + D_R - D_{Q+R} - D_{P+R} - D_{P+Q}$ respectively, and satisfy the equation $\gamma(\cdot)\gamma(T-\cdot) = 1$ where $T = P$ and $T = P+Q+R$ respectively. (More generally, if $[P_1, P_2, P_3, P_4; Q_1, Q_2, Q_3, Q_4]$ are in Riemann positions, γ has for divisor $D_{P_1} + D_{P_2} + D_{P_3} + D_{P_4} - D_{Q_1} - D_{Q_2} - D_{Q_3} - D_{Q_4}$ and $2T = P_1 + P_2 + P_3 + P_4$.) In particular, if $T = 2T'$, $\gamma(T')^2 = 1$, so γ is determined up to a sign from its divisor and this equation.

4.2. **Cubical structure and cubical arithmetic.** We can use Proposition 4.1 to define a "cubical arithmetic".

4.2.1. *Normalised symmetric isomorphism.* Let $\mathcal{L}$ be a symmetric line bundle on A, and $\alpha : \mathcal{L} \simeq [-1]^*\mathcal{L}$ an isomorphism.

Then if $P \in A$, α induces $\alpha(P) : \mathcal{L}(P) \simeq ([-1]^*\mathcal{L})(P) = \mathcal{L}(-P)$.

Following [Mum66, § 2], we call α normalised if $\alpha(0_A)$ is the identity map; such an isomorphism always exist.

4.2.2. *Rigidifications.* When $\mathcal{L}$ is a line bundle on an abelian variety A, we recall that a rigidification of $\mathcal{L}$ at a point $P \in A$ is the choice of an isomorphism $\mathcal{O}_A(P) \to \mathcal{L}(P) := \mathcal{L} \otimes \mathcal{O}_A(P)$ (we will take our rigidifications to be rational). The rigidifications at P form a torsor under $\mathbb{G}_m$.

We remark that a line bundle is locally trivial for the Zariski topology, so we can always find a local trivialisation around P given by an isomorphism $\mathcal{O}_{A,P} \to \mathcal{L} \otimes \mathcal{O}_{A,P}$. A trivialisation $\phi_P : \mathcal{O}_{A,P} \simeq \mathcal{L} \otimes \mathcal{O}_{A,P}$ induces an isomorphism on the residual fibers: $\phi_P : k(P) \simeq \mathcal{L}(P) := \mathcal{L} \otimes k(P)$, so a rigidification at P.

Conversely, a rigidification, i.e. an isomorphism on the fibers always lift (non uniquely) to a local trivialisation. Looking at the image of $1 \in \mathcal{O}$ under a trivialisation, we see that a choice of trivialisation is the same as a choice of local section s of $\mathcal{L}$ at P which generates

$\mathcal{L}$ locally at P, or equivalently $s(P) \neq 0$ (the value of s at P is not well defined, but the fact that it is zero or not is). Reformulated in terms of a divisor D associated to $\mathcal{L}$, a local section s is a function such that $\mathrm{div}(s) + D$ has no poles at P, and s generates $\mathcal{L}$ locally at P if for all other local sections t, the function t/s does not have a pole at P. Trivialisations are more conveniently expressed in terms of line bundles, which is why we have started to switch to this language in Section 4.1. Two local trivialisations, induced by local sections s_1, s_2, give the same rigidification at P iff $(s_1/s_2)(P) = 1$.

Furthermore if $\mathcal{L} \simeq \mathcal{M}$, then since A/k is proper and integral, $\mathrm{Hom}(\mathcal{L}, \mathcal{M}) \simeq \Gamma(_A) \simeq k$. So a rigidification of $\mathcal{L}, \mathcal{M}$ at P is enough to fix uniquely a global isomorphism $\mathcal{L} \to \mathcal{M}$.

4.2.3. *Cubical torsor structure.* Let $\mathcal{M} \in \mathrm{Pic}^0(A)$, $m : A \times A \to A$ be given by the addition map, and π_1, π_2 the two projections. Consider the line bundle $\Lambda\mathcal{M} = m^*\mathcal{M} \otimes \pi_1^*\mathcal{M}^{-1} \otimes \pi_2^*\mathcal{M}^{-1}$. Since $\mathcal{M}$ is algebraically equivalent to 0, it is translation invariant, so $\Lambda\mathcal{M}$ is fiberwise trivial, hence is trivial by the seesaw theorem; it is completely rigidified by a rigidification of $\mathcal{M}$ at 0_A. In particular, once a rigidification at 0_A of $\mathcal{M}$ is fixed, we have canonical isomorphisms $\mathcal{M}(Q_1 + Q_2) \to \mathcal{M}(Q_1) \otimes \mathcal{M}(Q_2)$, satisfying various natural compatibility relations on $A \times A \times A$. This is the squared structure associated to a line bundle algebraically equivalent to 0.

Now if $\mathcal{L}$ is a line bundle, then $\mathcal{M}_P := t_P^*\mathcal{L} \otimes \mathcal{L}^{-1}$ is algebraically equivalent to 0. Fixing a rigidification of $\mathcal{L}$ at 0_A, we obtain canonical isomorphisms $\mathcal{L}(P + Q_1 + Q_2) \otimes \mathcal{L}(Q_1) \otimes \mathcal{L}(Q_2) \simeq \mathcal{L}(P + Q_1) \otimes \mathcal{L}(P + Q_2) \otimes \mathcal{L}(Q_1 + Q_2)$, which depends on a choice of rigidification of $\mathcal{M}_P$ at 0_A. These isomorphisms are subsumed as follows: let $m_{123} : A \times A \times A \to A, (P_1, P_2, P_3) \to P_1 + P_2 + P_3$, $m_{ij} : A \times A \times A \to A, (P_1, P_2, P_3) \to P_i + P_j$ and $m_i = \pi_i$ the projection map. Consider the line bundle $m_{123}^*\mathcal{L} \otimes m_{12}^*\mathcal{L}^{-1} \otimes m_{23}^*\mathcal{L}^{-1} \otimes m_{13}^*\mathcal{L}^{-1} \otimes m_1^*\mathcal{L} \otimes m_2^*\mathcal{L} \otimes m_3^*\mathcal{L}$. Then the theorem of the cube implies that this torsor is trivial, and globally rigidified by a rigidification of $\mathcal{L}$ at 0_A. This is the cubical structure associated to a line bundle $\mathcal{L}$; by the discussion above this cubical structure can be recovered from the squared structures on the $\mathcal{M}_P$.

4.2.4. *Cubical arithmetic.* On an abelian variety A, fixing a rigidification of $\mathcal{L}$ at P is the same as fixing a rigidification of $\mathcal{L}_P := t_P^*$ at 0_A: a local isomorphism $\phi : \mathcal{O}_A(0_A) \to \mathcal{L}_P(0_A)$ at 0_A induces by pullback a local isomorphism $t_P^*\phi : \mathcal{O}_A(P) \to \mathcal{L}(P)$ at P.

Given $P \in A$, we will denote by $\widetilde{P}$ the choice of a rigidification $\phi_{\widetilde{P}}$ of $\mathcal{L}_P$ (implicitly at 0_A). If $\lambda \in \mathbb{G}_m$, we will denote by $\lambda\widetilde{P}$ the rigidification $\lambda\phi_{\widetilde{P}}$. We will call $\widetilde{P}$ a cubical point; the reason for the terminology and the notation will become clear in Section 4.5 where we introduce a convenient way to represent $\phi_{\widetilde{P}}$: a cubical point $\widetilde{P}$ is a "point" lying above the projective point P, with a cubical arithmetic induced from Section 4.2.3.

The cubical arithmetic may be defined as follow: if we have fixed a rigidification of $\mathcal{L}$ at 0_A and rigidifications of $\mathcal{L}$ at $P_1, P_2, P_3, P_1 + P_2, P_1 + P_3, P_2 + P_3$, then by the cubical structure, using the canonical isomorphism $\mathcal{L}(P_1+P_2+P_3) \otimes \mathcal{L}(P_1) \otimes \mathcal{L}(P_2) \otimes \mathcal{L}(P_3) \simeq \mathcal{L}(P_1 + P_2) \otimes \mathcal{L}(P_1 + P_3) \otimes \mathcal{L}(P_2 + P_3)$, we have a canonical rigidification of $\mathcal{L}$ at $P_1 + P_2 + P_3$.

More generally, we can use the algebraic Riemann relations. Let $[P_1, P_2, P_3, P_4; Q_1, Q_2, Q_3, Q_4]$ be points in Riemann position. By Proposition 4.1, given a rigidification $\widetilde{P_i}, \widetilde{Q_i}$ on seven out of our eight points P_i, Q_i, there is a *canonical* rigidification associated to the last one. We call the corresponding rigidification the one induced by the cubical arithmetic. We then say that the cubical points $[\widetilde{P}_1, \widetilde{P}_2, \widetilde{P}_3, \widetilde{P}_4; \widetilde{Q}_1, \widetilde{Q}_2, \widetilde{Q}_3, \widetilde{Q}_4]$ are in Riemann position.

Example 4.4. By Example 4.3, we have the following special cases of cubical arithmetic. We fix once and for all a choice for $\widetilde{0_A}$.

- Given $\widetilde{P}, \widetilde{Q}, \widetilde{P-Q}$, we have a canonical rigidification $\widetilde{P+Q}$, which we call a cubical differential addition. We denote $\widetilde{P+Q} = \mathrm{DiffAdd}(\widetilde{P}, \widetilde{Q}, \widetilde{P-Q})$. As a special case where $\widetilde{P} = \widetilde{Q}$ and $\widetilde{P-Q} = \widetilde{0_A}$, we have a cubical doubling $\widetilde{2P} = \mathrm{Double}(\widetilde{P})$.
- Given $\widetilde{P}, \widetilde{Q}, \widetilde{R}, \widetilde{P+Q}, \widetilde{P+R}, \widetilde{Q+R}$, we have a canonical rigidification $\widetilde{P+Q+R}$, which we call a cubical three way add. We denote $\widetilde{P+Q+R} = \mathrm{ThreeWayAdd}(\widetilde{P}, \widetilde{Q}, \widetilde{R}, \widetilde{P+Q}, \widetilde{P+R}, \widetilde{Q+R})$.
- More generally, given points $\widetilde{P_i}$ and their two by two sums $\widetilde{P_i+P_j}$, we can compute a multi way addition $\widetilde{P_1+\cdots+P_m}$ by iterating multiple three way additions.

In particular, given $\widetilde{P}$ we can use a ladder cubical differential additions and doublings to compute a cubical scalar multiplication $\widetilde{\ell P}$. Likewise, given $\widetilde{P+Q}, \widetilde{P}, \widetilde{Q}$, we can compute $\widetilde{\ell P+Q}$ though a ladder. We will denote $\widetilde{\ell P+Q} = \mathrm{ScalarMult}(\ell, \widetilde{P+Q}, \widetilde{P}, \widetilde{Q})$ and $\widetilde{\ell P} = \mathrm{ScalarMult}(\ell, \widetilde{P})$.

We can also define the opposite of a point as follows. Since $\mathcal{L}$ is symmetric, we take the normalised isomorphism $\alpha : [-1]^*\mathcal{L} \simeq \mathcal{L}$. If $\widetilde{P}$ is a rigidification of $\mathcal{L}$ at P, $[-1]^*\widetilde{P}$ is a rigidification of $[-1]^*\mathcal{L}$ at $-P$, and we can use the isomorphism above to obtain a rigidification of $\mathcal{L}$ at $-P$, which we denote by $-\widetilde{P}$. Since we use the normalised symmetric isomorphism, we have $-\widetilde{0_A} = \widetilde{0_A}$.

The cubical arithmetic is homogeneous with respect to the action by $\mathbb{G}_m$. (We recall that if $\widetilde{P}$ is a cubical point corresponding to an isomorphism $O_A(P) \to \mathcal{L}(P)$, $\lambda\widetilde{P}$ is the cubical point corresponding to this isomorphism multiplied by λ.) More precisely, we have:

Lemma 4.5.

- *If* $[\widetilde{P}_1, \widetilde{P}_2, \widetilde{P}_3, \widetilde{P}_4; \widetilde{Q}_1, \widetilde{Q}_2, \widetilde{Q}_3, \widetilde{Q}_4]$ *are in Riemann position, then so are* $[\lambda_{P,1}\widetilde{P}_1, \lambda_{P,2}\widetilde{P}_2, \lambda_{P,3}\widetilde{P}_3, \lambda_{P,4}\widetilde{P}_4; \lambda_{Q,1}\widetilde{Q}_1, \lambda_Q$ *whenever* $\lambda_{P,1}\lambda_{P,2}\lambda_{P,3}\lambda_{P,4} = \lambda_{Q,1}\lambda_{Q,2}\lambda_{Q,3}\lambda_{Q,4}$*;*
- $-(\lambda\widetilde{P}) = \lambda(-\widetilde{P})$
- $\mathrm{DiffAdd}(\lambda_P\widetilde{P}, \lambda_Q\widetilde{Q}, \lambda_{PQ}\widetilde{P-Q}, \lambda_0\tilde{0}) = \frac{\lambda_P^2\lambda_Q^2}{\lambda_{PQ}\lambda_0^2}\,\mathrm{DiffAdd}(\widetilde{P}, \widetilde{Q}, \widetilde{P-Q}, \tilde{0})$;
- $\mathrm{ThreeWayAdd}(\lambda_{P_1}\widetilde{P_1}, \lambda_{P_2}\widetilde{P_2}, \lambda_{P_3}\widetilde{P_3}, \lambda_{P_2+P_3}\widetilde{P_2+P_3}, \lambda_{P_1+P_3}\widetilde{P_1+P_3}, \lambda_{P_1+P_2}\widetilde{P_1+P_2}, \lambda_0\tilde{0}) = \frac{\lambda_{P_1+P_2}\lambda_{P_2+P_3}\lambda_{P_1+P_3}\lambda_0}{\lambda_{P_1}\lambda_{P_2}\lambda_{P_3}}\,\mathrm{ThreeWayAdd}(\widetilde{P_1}, \widetilde{P_2}, \widetilde{P_3}, \widetilde{P_2+P_3}, \widetilde{P_1+P_3}, \widetilde{P_1+P_2}, \tilde{0})$
- $\mathrm{ScalarMult}(\ell, \lambda_{PQ}\widetilde{P+Q}, \lambda_P\widetilde{P}, \lambda_Q\widetilde{Q}, \lambda_0\tilde{0}) = \lambda_{PQ}^{\ell}\lambda_P^{\ell(\ell-1)}\lambda_Q^{-(\ell-1)}\lambda_0^{-\ell(\ell-1)}\lambda_P^{\ell^2}\,\mathrm{ScalarMult}(\ell, \widetilde{P+Q}, \widetilde{P}, \widetilde{Q}, \tilde{0})$

We warn that the cubical arithmetic defined in Example 4.4 does not form a group structure (unlike biextensions which do induce compatible groups structures on subsets). The cubical arithmetic shares many analogy with the arithmetic of a Kummer variety. Furthermore, using [Mor85, § I.5], all expected natural relations do hold in the cubical arithmetic. We note also that if P is a point of ℓ-torsion and we fix a rigidification $\widetilde{P}$, then $\widetilde{\ell P}$ is a rigidification of $\mathcal{L}$ at 0_A so differ from our global choice of $\tilde{0}_{A_k}$ by some constant λ: $\widetilde{\ell P} = \lambda\tilde{0}_{A_k}$. In general, we will have $\lambda \neq 1$ (i.e., $\widetilde{P}$ will not be of ℓ-torsion), in fact constants like λ will provide us exactly the monodromy informations which gives the Weil and Tate pairings.

4.2.5. *Cubical nets.* Assume that we are given r points $P_i \in A$, and that we have fixed cubical points $\widetilde{P_i}$ and $\widetilde{P_i+P_j}$ for all $i \neq j$. Then for each $n = (n_1, \ldots, n_r) \in \mathbb{Z}^r$, the cubical arithmetic determines a cubical point $\sum n_i\widetilde{P}_i := \widetilde{\sum n_iP_i}$. (For the link with elliptic nets, we refer to Section 4.9.5).

The compatibility relations for the cubical torsor structure show that we can use whichever Riemann relations to arrive at the end result, they all give the same cubical points. For instance,

we can compute $2P_1 \widetilde{+} 2P_2$ as $\mathrm{Double}(P_1 \widetilde{+} P_2)$, or by computing successively via doublings and differential additions $\widetilde{2P_1}$, $2P_1 \widetilde{+} P_2$, $2P_1 \widetilde{+} 2P_2$, or doing it the other way around: $\widetilde{2P_2}$, $2P_2 \widetilde{+} P_1$, $2P_1 \widetilde{+} 2P_2$. The end result is the same.

Remark 4.6. One should be careful that if P_1 is a point of ℓ_1 torsion, $\widetilde{\sum n_i P_i}$ and $\ell_1 P_1 \widetilde{+} \sum n_i P_i$ will not give the same cubical point in general; they differ by some projective factor λ_{P_1}. This is the whole point of the cubical arithmetic; this monodromy will give pairings.

We can extend Lemma 4.5 by recurrence on the rank r (compare with [Stao8, Proposition 5.1.2, Theorem 6.2.3, Theorem 10.1.1]):

Lemma 4.7. *If we change $\widetilde{P_i}$ into $\lambda_i \widetilde{P_i}$, and $P_i \widetilde{+} P_j$ into $\lambda_{ij} P_i \widetilde{+} P_j$, then $\widetilde{\sum n_i P_i}$ is changed into*

$$\prod_i \lambda_i^{n_i^2} \prod_{i<j} \left(\frac{\lambda_{ij}}{\lambda_i \lambda_j}\right)^{n_i n_j} \widetilde{\sum n_i P_i}.$$

A convenient way to remember Lemma 4.7, which I learnt from [Stao8, § 10.3], is to write $\lambda'_{ii} = \lambda_i, \lambda_{ij} = (\lambda'_{ij})^2 \lambda_i \lambda_j$ (we will see that the choice of square root for λ'_{ij} does not matter). Then if we write a formal square $(\sum n_i x_i)^2 = \sum_i n_i^2 x_i^2 + \sum_{i<j} 2 n_i n_j x_i x_j = \sum_{i \leq j} c_{ij} x_i x_j$, we see that $\widetilde{\sum n_i P_i}$ is changed into $\prod_{i \leq j} {\lambda'_{ij}}^{c_{ij}} \widetilde{\sum n_i P_i}$.

Lemma 4.8 (Periodicity). *Let $\Gamma \subset \mathbb{Z}^r$ be the lattice of relations on the P_i: for all $\gamma = (\gamma_1, \dots, \gamma_r) \in \Gamma$, $\sum \gamma_i P_i = 0$.*

Then for all $n \in \mathbb{Z}^r$, $\gamma \in \Gamma$, the two cubical points $\sum n_i \widetilde{P}_i$ and $\sum (n_i + \gamma_i) \widetilde{P}_i$ lie over the same point $\sum n_i P_i$, so differ by a projective factor $\lambda(\gamma, n)$: $\sum (n_i + \gamma_i) \widetilde{P}_i = \lambda(\gamma, n) \widetilde{\sum n_i P_i}$.

Then λ is quadratic, and affine linear in n: for $\gamma_1, \gamma_2, \gamma_3 \in \Gamma$, $n_1, n_2, n_3 \in \mathbb{Z}^r$,

$$\lambda(\gamma_1+\gamma_2+\gamma_3, n_1+n_2+n_3)\lambda(\gamma_1,n_1)\lambda(\gamma_2,n_2)\lambda(\gamma_3,n_3) = \lambda(\gamma_1+\gamma_2,n_1+n_2)\lambda(\gamma_2+\gamma_3,n_2+n_3)\lambda(\gamma_1+\gamma_3,n_1+$$
$$\lambda(\gamma, n_1+n_2+n_3)\lambda(\gamma,n_3) = \lambda(\gamma,n_1+n_3)\lambda(\gamma,n_2+n_3).$$

Proof. Using Lemma 4.7, the proof is the same as in [Stao8, Theorem 10.2.3] (except that our λ is well defined everywhere, so this simplifies a bit the proof). □

Example 4.9. If P is a point of ℓ-torsion and we fix a cubical point $\widetilde{P}$, then writing $\ell \widetilde{P} = \lambda_0 \tilde{0}$, $(\ell+1)\widetilde{P} = \lambda_1 \widetilde{P}$, with $\lambda_1 = \lambda_0 \lambda'_1$, we have $(u\ell + v)\widetilde{P} = \lambda_0^{u^2} {\lambda'_1}^{uv} v \widetilde{P}$, and writing $2\ell \widetilde{P} = \ell \widetilde{P} + \ell \widetilde{P}$, we obtain $\lambda_0^2 = {\lambda'_1}^{\ell}$, as in [Stao8, Theorem 10.2.2]. So if we let $\alpha, \beta \in \bar{k}$ such that[1] $\beta^\ell = \lambda_0$, $\beta^2 = \lambda'_1$ $\alpha^\ell = \beta$, we have $(u\ell+v)\widetilde{P} = \beta^{\ell u^2 + 2uv} v \widetilde{P} = \alpha^{(u\ell+v)^2 - v^2} v \widetilde{P}$. (There is a small typo in [Stao8, Theorem 10.2.2], the α defined there is our β, but the formula is only correct using the α we define here.)

We will see in Section 4.4 that λ'_1 is the self Tate pairing $e_{T,\ell}(P,P)$. Furthermore, if $\ell = 2\ell'+1$ is odd, then $(\ell'+1)\widetilde{P} = \beta \cdot -\ell' \widetilde{P}$, and if $\ell = 2\ell'+2$ is even, $(\ell'+2)\widetilde{P} = \beta^2 \cdot -\ell' \widetilde{P}$.

Example 4.10. If $\ell_1 P + \ell_2 Q = 0$, and we fix $\widetilde{P}, \widetilde{Q}, \widetilde{P \widetilde{+} Q}$, we have $(u\ell_1 + v_1)\widetilde{P \widetilde{+} (u\ell_2 + v_2)Q} = {\lambda'_P}^{uv_1} {\lambda'_Q}^{uv_2} {\lambda_{PQ}}^{u^2} v_1 \widetilde{P \widetilde{+} v_2 Q}$ where $2P + (\widetilde{\ell_1 P} + \ell_2 Q) = \lambda'_P \widetilde{2P}$, $2Q + (\widetilde{\ell_1 P} + \ell_2 Q) = \lambda'_Q \widetilde{2Q}$, $P + Q + \widetilde{(\ell_1 P + \ell_2 Q)} = \lambda'_P \lambda'_Q \lambda_{PQ} \widetilde{P \widetilde{+} Q}$. Compare with [Stao8, Lemma 10.2.5].

Using that λ is defined everywhere, we can also recover these projective factors as follows: $\ell_1 \widetilde{P \widetilde{+} \ell_2 Q} = \lambda_{PQ} \tilde{0}$, $P + \widetilde{\ell_1 P} + \ell_2 Q = \lambda'_P \lambda_{PQ} \widetilde{P}$, $Q + \widetilde{\ell_1 P} + \ell_2 Q = \lambda'_Q \lambda_{PQ} \widetilde{Q}$.

[1] This is always possible if ℓ is odd; if ℓ is even we will see in Section 6.2.2 that this is possible if and only if $e_{D,\star}(\ell/2P) = 1$.

Example 4.11. Suppose that P_i is of order ℓ_i and that we have fixed $\widetilde{P_i}$, $\widetilde{P_i + P_j}$. Then, letting $\lambda_i = \lambda_{i,i}, \lambda'_i = \lambda'_{i,i}$ and $\lambda_{i,j} = \lambda_{j,i}$ for symmetry, we have $\sum_i \widetilde{(u_i \ell_i + v_i) P_i} = \prod_{i,j} \lambda_{i,j}^{u_i u_j} {\lambda'_{i,j}}^{u_i v_j} \widetilde{(\sum_i v_i P_i)}$.

We can recover the projective coefficients as follows: $\widetilde{\ell_i P_i} = \lambda_{ii} \tilde{0}$, $\ell_i \widetilde{P_i + \ell_j P_j} = \lambda_i \lambda_j \lambda_{i,j}^2 \tilde{0}$, $\ell_i \widetilde{P_i + P_j} = \lambda'_{i,j} \lambda_i \widetilde{P_j}$ (we remark that $\lambda_{i,j}^2$ and $\lambda_{i,i}$ are in the base field). More over, we have $\lambda_i^2 = {\lambda'_i}^{\ell_i}$ and $\lambda_{i,j}^2 = {\lambda'_{i,j}}^{\ell_j} = {\lambda'_{j,i}}^{\ell_i}$.

If we change $\widetilde{P_i}$ and $\widetilde{P_i + P_j}$ by projective factors μ_i, $\mu_i \mu_j \mu_{i,j}$, then we change $\lambda_i, \lambda_{i,j}^2$ into $\lambda_i \mu_i^{\ell_i^2}, \lambda_{i,j}^2 \mu_{i,j}^{\ell_i \ell_j}$, and $\lambda'_{i,j}$ into $\lambda'_{i,j} \mu_{i,j}^{\ell_i}$.

We will see that $\lambda'_{i,j}$ gives the non reduced Tate pairing $e_{T,\ell_i}(P_i, P_j)$, and that if $\ell = d_i \ell_i = d_j \ell_j$, the Weil pairing is given by $e_{W,\ell}(P_i, P_j) = {\lambda'_{i,j}}^{d_i} / {\lambda'_{j,i}}^{d_j}$.

4.2.6. *Action of the theta group on cubical points.* There is an action of the theta group $G(D)$ on cubical points. Let $T \in A[D]$, and $(T, g_T) \in G(D)$ an element of the theta group. By definition, this is the same as a choice of a global isomorphism $\phi_T : \mathcal{L} \to t_T^* \mathcal{L}$. In particular, this isomorphism is completely determined by a rigidification of $t_T^* \mathcal{L}$ at 0, hence a cubical point $\widetilde{T}$ (recall that we have fixed once and for all a cubical point $\tilde{0}$).

In other words, g_T is completely determined by $\widetilde{T}$, and conversely, and by definition of the cubical arithmetic and the arithmetic of the theta group, cubical points above $A[D]$ with their arithmetic are isomorphic to the theta group $G(D)$ (or to reformulate: the cubical arithmetic become trivial over $A[D]$). In particular, given cubical points $\widetilde{T}_1, \widetilde{T}_2$ where $T_1, T_2 \in A[D]$, $\widetilde{T_1 + T_2}$ is well defined.

Now, given a cubical point $\widetilde{P}$, i.e a rigidification of $\mathcal{L}$ at P, we can use ϕ_T to obtain a rigidification of $\mathcal{L}_T$ at P, which we call $g_T \cdot \widetilde{P}$ or $\widetilde{P} + \widetilde{T} := \widetilde{P + T} = g_T \cdot \widetilde{P}$.

Lemma 4.12. *If the cubical points $[\widetilde{P}_1, \widetilde{P}_2, \widetilde{P}_3, \widetilde{P}_4; \widetilde{Q}_1, \widetilde{Q}_2, \widetilde{Q}_3, \widetilde{Q}_4]$ are in Riemann position, and $[\widetilde{S}_1, \widetilde{S}_2, \widetilde{S}_3, \widetilde{S}_4; \widetilde{T}_1, \widetilde{T}_2, \widetilde{T}_3, \widetilde{T}_4]$ are too, with each $S_i, T_i \in A[D]$, then so are $[\widetilde{P_1 + S_1}, \widetilde{P_2 + S_2}, \widetilde{P_3 + S_3}, \widetilde{P_4 + S_4}; \widetilde{Q_1 + T_1}, \widetilde{Q_2 + T_2}, \widetilde{Q_3 + T_3}, \widetilde{Q_4 + T_4}]$.*

Proof. This follows by unraveling the definitions. Anticipating Section 4.8, an analytic proof is also given in [LR22a, Lemma 3.5]. □

Corollary 4.13. *Given cubical points $\widetilde{P_i}$ and $\widetilde{P_i + P_j}$, and cubical points $\widetilde{T_i}$ where $T_i \in A[D]$, we have $\sum n_i (\widetilde{P_i} + \widetilde{T_i}) = \sum n_i \widetilde{P_i} + \sum n_i \widetilde{T_i}$.*

4.2.7. *Functions associated to cubical cycles.* Let D be a divisor on A, $Z = \sum n_i (P_i)$ be a cycle on A such that $s(Z) = \sum n_i P_i = 0$, then there exists a rational function f_Z with divisor $\sum n_i D_{P_i}$.

Switching to the language of line bundles, if we are given cubical points $\widetilde{P}_i$ above each P_i (as a shortcut we could say we have a cubical cycle $\widetilde{Z} = \sum n_i (\widetilde{P_i})$ above Z), then since the line bundle $\otimes_i \mathcal{L}_{P_i}^{\otimes n_i}$ is globally trivial, and each rigidification $\widetilde{P}_i$ on $\mathcal{L}_{P_i}$ induce a rigidification of $\otimes_i \mathcal{L}_{P_i}^{\otimes n_i}$, the cubical cycle $\widetilde{Z}$ induces a well defined global isomorphism $\mathcal{O}_A \simeq \otimes_i \mathcal{L}_{P_i}^{\otimes n_i}$.

There is thus a well defined function $f_{\widetilde{Z}}$ with divisor $\sum n_i D_{P_i}$.

More generally, if $s(Z) \in A[D]$, then there also exists a rational function f_Z with divisor $\sum n_i D_{P_i}$. This rational function is completely determined by $\widetilde{Z}$, and by the choice of a theta group element $(s(Z), g_{s(Z)}) \in G(D)$, which by Section 4.2.6 is the same as the choice of a cubical point $\widetilde{s(Z)}$ (along with the choice of $\tilde{0}$).

We will pursue this construction in Sections 4.5.2 and 4.6.

4.3. **The cubical representation of the biextension.** We can now define the cubical representation of a biextension element $g_{P,Q} \in X_D$. Recall that the function $g_{P,Q}$ has for divisor $D_{P+Q} + D - D_P - D_Q$; in terms of line bundles it corresponds to an isomorphism $\mathcal{L}_P \otimes \mathcal{L}_Q \simeq \mathcal{L}_{P+Q} \otimes \mathcal{L}$ where $\mathcal{L} = O(D)$ is the line bundle associated to D.

We can proceed as in Section 4.2.7 for the cycle $Z = (P+Q) + (0) - (P) - (Q)$. Namely assume that we are given local rigidifications at 0, $\tilde{0}, \widetilde{P}, \widetilde{Q}, \widetilde{P+Q}$, of $\mathcal{L}, \mathcal{L}_P, \mathcal{L}_Q, \mathcal{L}_{PQ}$ respectively. These give a local isomorphism $\mathcal{L}_P \otimes \mathcal{L}_Q \to \mathcal{L}_{P+Q} \otimes \mathcal{L}$ at 0. Since these line bundles are globally isomorphic, this local isomorphism lift to a unique global isomorphism, hence defines a biextension element which we denote by $(P, Q, g_{P,Q}) = [\widetilde{P}, \widetilde{Q}; \tilde{0}, \widetilde{P+Q}]$. (We will often assume $\tilde{0}$ has been fixed once and for all and drop it from our notations). We call this the cubical representation of the biextension. It is convenient to also use a two dimensional notation:

$$(P, Q, g_{P,Q}) = \begin{pmatrix} \tilde{0} & \widetilde{P} \\ \widetilde{Q} & \widetilde{P+Q} \end{pmatrix}.$$

Recall that $\widetilde{P}$ is a notation for a rigidification $\phi_{\widetilde{P}} : O_A(0) \to \mathcal{L}_P(0)$ at 0, so by abuse of notations we denote by $\widetilde{P+Q} \otimes \tilde{0} \otimes \widetilde{P}^{-1} \otimes \widetilde{Q}^{-1}$ (or even $\frac{\widetilde{P+Q}\tilde{0}}{\widetilde{P}\widetilde{Q}}$) the corresponding tensor product of rigidifications $O_A(0) \to \mathcal{L}_{P+Q}(0) \otimes \mathcal{L}(0) \otimes \mathcal{L}_P^{-1}(0) \otimes \mathcal{L}_Q^{-1}(0)$. The rigidification $\frac{\widetilde{P+Q}\tilde{0}}{\widetilde{P}\widetilde{Q}}$ lift to a global trivialisation, which is associated precisely to the biextension function $g_{P,Q}$ given by the cubical representation.

Remark 4.14. This representation is quite similar to the evaluation representation; instead of representing $g_{P,Q}$ through its extended value at 0, we use a rigidification of $\mathcal{L}_{P+Q} \otimes \mathcal{L} \otimes \mathcal{L}_P^{-1} \otimes \mathcal{L}_Q^{-1}$ at 0. We refer to Section 4.5 for the relationship between the two representations. The advantage of the latter representation is that it can be further refined though local rigidifications of $\mathcal{L}_{P+Q}, \mathcal{L}_P, \mathcal{L}_Q \cdots$ at 0 (which do not come from global functions evaluated at 0 since these line bundles are not trivial individually).

In the evaluation representation we could change our evaluation point from 0 to R_0; likewise here we could use instead a representation of the form

$$(P, Q, g_{P,Q}) = \begin{pmatrix} \widetilde{R}_0 & \widetilde{P+R_0} \\ \widetilde{Q+R_0} & \widetilde{P+Q+R_0} \end{pmatrix}.$$

We could even use the cubical three way add introduced below to change the base point R_0 of our representation on the fly; but on the following we will stick to using $R_0 = 0$ for simplicity.

We remark that this representation is redundant: the possible choices of $g_{P,Q}$ form a $\mathbb{G}_m$-torsor, while the choice on the right hand choice consist of four $\mathbb{G}_m$-torsors. From the definition, we have:

Lemma 4.15. *Given $\lambda_0, \lambda_P, \lambda_Q, \lambda_{PQ} \in \mathbb{G}_m$, we have*

$$[\lambda_P\widetilde{P}, \lambda_Q\widetilde{Q}; \lambda_0\tilde{0}, \lambda_{PQ}\widetilde{P+Q}] = \frac{\lambda_{PQ}\lambda_0}{\lambda_P\lambda_Q}[\widetilde{P}, \widetilde{Q}; \tilde{0}, \widetilde{P+Q}],$$

so they define the same biextension element if and only if $\lambda_{PQ}\lambda_0 = \lambda_P\lambda_Q$.

We will see that although it is redundant, the cubical representation give a fast biextension arithmetic. An analogy is using the redundant modified Jacobian coordinates (X, Y, Z, T) to compute the scalar multiplication of an elliptic curve.

Now we explain how to recover the biextension arithmetic in term of our cubical representation and the cubical arithmetic. There are actually several possibilities (the cubical arithmetic is a refinement of the biextension arithmetic), and we will focus on two specific formulas.

Theorem 4.16. *Fix $\tilde{0}$, and let $\widetilde{P_1}, \widetilde{P_2}, \widetilde{P_1 + Q}, \widetilde{P_2 + Q}$ be cubical points, and let $g_{P_1,Q} = [\widetilde{P_1}, \widetilde{Q}; \tilde{0}, \widetilde{P_1 + Q}]$, $g_{P_2,Q} = [\widetilde{P_1}, \widetilde{Q}; \tilde{0}, \widetilde{P_2 + Q}]$ the associated biextension elements. (We can always use Lemma 4.15 to ensure that our representation use the same $\widetilde{Q}$).*

Take $\widetilde{P_1 + P_2}$ an arbitrary cubical point above $P_1 + P_2$ and let

$$P_1 + \widetilde{P_2} + Q = \mathrm{ThreeWayAdd}(\widetilde{P_1}, \widetilde{P_2}, \widetilde{Q}, \widetilde{P_2 + Q}, \widetilde{P_1 + Q}, \widetilde{P_1 + P_2}).$$

Then $g_{P_1,Q} \star_1 g_{P_2,Q} = [\widetilde{P_1 + P_2}, \widetilde{Q}; \tilde{0}, P_1 + \widetilde{P_2} + Q]$. We call this a (cubical) biextension standard addition.

Alternatively, assume that $[\widetilde{P_1 - P_2}, \widetilde{Q}; \tilde{0}, P_1 - \widetilde{P_2} + Q]$ is a cubical representation of $g_{P_1-P_2,Q} = g_{P_1,Q} \star_1 g_{P_2,Q}^{\star_1,-1}$. Let $\widetilde{P_1 + P_2} = \mathrm{DiffAdd}(\widetilde{P_1}, \widetilde{P_2}, \widetilde{P_1 - P_2})$ and $P_1 + \widetilde{P_2} + Q = \mathrm{DiffAdd}(\widetilde{P_1 + Q}, \widetilde{P_2}, P_1 - \widetilde{P_2} + Q)$. Then $g_{P_1,Q} \star_1 g_{P_2,Q} = [\widetilde{P_1 + P_2}, \widetilde{Q}; \tilde{0}, P_1 + \widetilde{P_2} + Q]$. We call this a (cubical) biextension differential addition.

As a particular case, when $P_1 = P_2 = P$, and we have $g_{P,Q} = [\widetilde{P}, \widetilde{Q}; \tilde{0}, \widetilde{P + Q}]$, we can take $g_{0,Q} = 1 = [\tilde{0}, \widetilde{Q}; \tilde{0}, \tilde{0}]$, so if we let $2\widetilde{P} = \mathrm{Double}(\widetilde{P})$ and $\widetilde{2P + Q} = \mathrm{DiffAdd}(\widetilde{P + Q}, \widetilde{P}, \widetilde{Q})$, we have: $g_{P,Q} \star_1 g_{P,Q} = [\widetilde{2P}, \widetilde{Q}; \tilde{0}, \widetilde{2P + Q}]$. We call this a (cubical) biextension differential doubling.

Proof. This is a translation of the fact, explained in [Bre83; Mor85, Chapitre 1], that the cubical structure induce a symmetric biextension, and the various natural compatibilities of the cubical arithmetic, to which we refer to [Mor85, § I.5]. We will also give an analytical argument in Section 4.8.

For the reader who does not like abstract proofs, we will prove the first statement using the explicit formulas of Proposition 4.1 and leave the second as an exercice.

By definition of the cubical representation $[\widetilde{P_i}, \widetilde{Q}; \tilde{0}, \widetilde{P_i + Q}]$, the rigidification $\widetilde{P_i + Q} \otimes \tilde{0} \otimes \widetilde{P_i}^{-1} \otimes \widetilde{Q}^{-1}$ globalises to a global trivialisation, which from the divisor point of view corresponds to a function $g_{P_i,Q}$ with divisor $D_{P_i+Q} - D_{P_i} - D_Q$.

We will assume D symmetric by simplicity. By Example 4.3, the point $P_1 + \widetilde{P_2} + Q$ computed through the cubical arithmetic is such that $P_1 + \widetilde{P_2} + Q \otimes \widetilde{P_1} \otimes \widetilde{P_2} \otimes \widetilde{Q} \otimes \tilde{0}^{-1} \otimes \widetilde{P_2 + Q}^{-1} \otimes \widetilde{P_1 + Q}^{-1} \otimes \widetilde{Q}^{-1}$ corresponds to a global trivialisation induced by the function $g_{P_1,P_2}(P_1 + P_2 + Q - \cdot)/g_{P_1,P_2}(\cdot)$.

It follows that the function associated to the cubical representation $[\widetilde{P_1 + P_2}, \widetilde{Q}; \tilde{0}, P_1 + \widetilde{P_2} + Q]$, is equal to $\frac{g_{P_1,P_2}(P_1+P_2+Q-\cdot)}{g_{P_1,P_2}(\cdot)} g_{P_1,Q} g_{P_2,Q}$. Comparing with Equation (10), we need to check that $g_{P_1,P_2}(P_1 + P_2 + Q - \cdot) = g_{P_1,P_2}(Q + \cdot)$. These two functions have the same divisor (using our assumption that D is symmetric), and their evaluation at a point T such that $2T = P_1 + P_2$ yield the same value $g_{P_1,P_2}(Q + T)$, hence they are equal. We leave the case where D is only linearly symmetric to the reader (we could also invoke flat descent to reduce to the symmetric case). □

Example 4.17. If $g_{P,Q} = [\widetilde{P}, \widetilde{Q}; \tilde{0}, \widetilde{P + Q}]$, then $g_{P,Q}^{\star_1,-1} = [\widetilde{-P}, \widetilde{Q}; \tilde{0}, \widetilde{-P + Q}]$, where $\widetilde{-P + Q} = \mathrm{DiffAdd}(\widetilde{-P}, \widetilde{Q}, \widetilde{P + Q})$.

If we have a cubical representation of $g_{P_1,Q}, g_{P_2,Q}$ as above, and also a cubical representation of $g_{P_1,Q} \star_1 g_{P_2,Q}$: $g_{P_1+P_2,Q} = [\widetilde{P_1 + P_2}, \widetilde{Q}; \tilde{0}, P_1 + \widetilde{P_2} + Q]$, then $g_{P_1-P_2,Q} =$

Input: a biextension element $g_{P,Q} = [\widetilde{P}, \widetilde{Q}, \tilde{0}, \widetilde{P+Q}]$ represented by cubical points

Output: cubical points $\widetilde{\ell P}$, $\widetilde{\ell P + Q}$ such that $g_{P,Q}^{\star_1,\ell} = [\widetilde{\ell P}, \widetilde{Q}, \tilde{0}, \widetilde{\ell P + Q}]$

→ For each bit b_i of ℓ from left to right (skipping the first one), given a cubical representation $\widetilde{nP}, \widetilde{nP+Q}$, of $g_{P,Q}^{\star_1,n}$, where n is the current truncation of ℓ on the leftmost bits, do a cubical biextension double:
   a. $\widetilde{2nP+Q} = \mathrm{DiffAdd}(\widetilde{nP+Q}, \widetilde{nP}, \widetilde{Q})$;
   b. $\widetilde{2nP} = \mathrm{Double}(\widetilde{nP})$;

   And if $b_i = 1$, also do a cubical biextension addition:
   a. Compute $(2n+1)P = 2(nP) + P$ and take an arbitrary cubical lift $\widetilde{(2n+1)P}$ of $(2n+1)P$.
   b. $\widetilde{(2n+1)P + Q} = \mathrm{ThreeWayAdd}(\widetilde{2nP}, \widetilde{P}, \widetilde{Q}, \widetilde{P+Q}, \widetilde{2nP+Q}, \widetilde{(2n+1)P})$;

Algorithm 4.1. Biextension exponentiation via cubical double and add

$[\widetilde{P_1 - P_2}, \widetilde{Q}; \tilde{0}, \widetilde{P_1 - P_2 + Q}]$, is a cubical representation of $g_{P_1,Q} \star_1 g_{P_2,Q}^{\star_1,-1}$, where $\widetilde{P_1 - P_2} = \mathrm{DiffAdd}(\widetilde{P_1}, \widetilde{-P_2}, \widetilde{P_1 + P_2})$ and $\widetilde{P_1 - P_2 + Q} = \mathrm{DiffAdd}(\widetilde{P_1 + Q}, \widetilde{-P_2}, \widetilde{P_1 + P_2 + Q})$.

By Theorem 4.16, we have two algorithms to compute the biextension exponentiation $g_{P,Q}^{\star_1,\ell}$ in the cubical representation $g_{P,Q} = [\widetilde{P}, \widetilde{Q}; \tilde{0}, \widetilde{P+Q}]$. The first one, given in Algorithm 4.1 is a standard double and add algorithm, using standard (cubical) biextension addition, each biextension addition involving one abelian variety addition and one cubical three way addition. We note that by Theorem 4.16, we actually have two ways to compute biextension doubling in a double and add ladder: either we use a biextension normal addition of $g_{P,Q}$ with itself, or we use a cubical biextension differential doubling. In Algorithm 4.1 we use the second method, because it is the faster one for Kummer lines.

The second one is to do a differential ladder, using a cubical differential ladder to compute $\widetilde{\ell P}, \widetilde{\ell P + Q}$ from $g_{P,Q}$. Namely at each step we have $g_{P,Q}^m, g_{P,Q}^{m+1}$, represented via the cubical points $\widetilde{mP}, \widetilde{mP+Q}, \widetilde{(m+1)P}$ (in theory we would also need $Q + \widetilde{(m+1)P}$ to represent $g_{P,Q}^{m+1}$, but we will see we won't need it in the algorithm; and in any case it could be recomputed on the fly through a three way addition). In other words, we use a biextension ladder, each step involving a cubical biextension differential addition and a cubical biextension doubling (here using a normal cubical biextension doubling would not be correct). Each step of this biextension differential ladder then involves a cubical doubling and two cubical differential additions. (It should involve three cubical differential additions but from the remark above the one to compute the $Q + \widetilde{(m+1)P}$ is not used).

In Algorithm 4.2 we present such a ladder algorithm, except we use a ladder of the form $g_{P,Q}^{m-1}, g_{P,Q}^m$ represented by the points $\widetilde{(m-1)P}, \widetilde{mP+Q}, \widetilde{mP}$ instead.

We also need to explain how to work out the action of $A[D]$ on the biextension X_D from Lemma 3.9 via cubical arithmetic. Recall from Section 4.2.6 that we have an action of the theta group $G(D)$ on cubical points.

Lemma 4.18. *If $g_{P,Q} = [\widetilde{P}, \widetilde{Q}; \tilde{0}, \widetilde{P+Q}]$ and $T \in A[D]$, then letting g_T be any function such that $(T, g_T) \in G(D)$, $T \cdot g_{P,Q} = [g_T \cdot \widetilde{P}, \widetilde{Q}; \tilde{0}, g_T \cdot \widetilde{P+Q}] = [\widetilde{P+T}, \widetilde{Q}; \tilde{0}, \widetilde{P+Q+T}]$.*

4.4. **Pairings via the cubical representation.** Using Section 4.3 to represent biextension elements, by Theorems 3.11 and 4.16 we get the following formulas for pairings:

Input: a biextension element $g_{P,Q} = [\widetilde{P}, \widetilde{Q}, \tilde{0}, \widetilde{P+Q}]$ represented by cubical points

Output: cubical points $\ell\widetilde{P}, \widetilde{\ell P+Q}$ such that $g_{P,Q}^{\star_1,\ell} = [\ell\widetilde{P}, \widetilde{Q}, \tilde{0}, \widetilde{\ell P+Q}]$

- Compute $\widetilde{2P} = \mathrm{Double}(\widetilde{P})$, $\widetilde{2P+Q} = \mathrm{DiffAdd}(\widetilde{P+Q}, \widetilde{P}, \widetilde{Q})$ to get a representation of $g_{P,Q}^{\star_1,2}$.
- For each bit b_i of $\ell - 1$ from left to right (skipping the first one), given a cubical representation $\widetilde{nP}$, $\widetilde{(n+1)P}$, $\widetilde{(n+1)P+Q}$ of $g_{P,Q}^{\star_1,n}, g_{P,Q}^{\star_1,n+1}$ where n is the current truncation of $\ell - 1$ on the leftmost bits, compute:
  a. If $b_i = 0$:
     - $\widetilde{(2n+1)P+Q} = \mathrm{DiffAdd}(\widetilde{(n+1)P+Q}, \widetilde{nP}, \widetilde{P+Q})$;
     - $\widetilde{(2n+1)P} = \mathrm{DiffAdd}(\widetilde{(n+1)P}, \widetilde{nP}, \widetilde{P})$;
     - $\widetilde{2nP} = \mathrm{Double}(\widetilde{nP})$;
  b. If $b_i = 1$:
     - $\widetilde{2(n+1)P+Q} = \mathrm{DiffAdd}(\widetilde{(n+1)P+Q}, \widetilde{(n+1)P}, \widetilde{Q})$;
     - $\widetilde{2(n+1)P} = \mathrm{Double}(\widetilde{(n+1)P})$;
     - $\widetilde{(2n+1)P} = \mathrm{DiffAdd}(\widetilde{(n+1)P}, \widetilde{nP}, \widetilde{P})$;

Algorithm 4.2. Biextension exponentiation via a cubical ladder

Theorem 4.19. *Let $g_{P,Q} = [\widetilde{P}, \widetilde{Q}; \tilde{0}, \widetilde{P+Q}]$, and use any of the two cubical biextension exponentiation to get a representation $g_{P,Q}^{\star_1,\ell} = [\ell\widetilde{P}, \widetilde{Q}; \tilde{0}, \widetilde{\ell P+Q}]$.*

Assume that $P \in A[\ell D]$, we can then use the action of $-\ell P$ from Lemma 4.18 to compute $-\ell P \cdot g_{P,Q}^{\star_1,\ell} = [-\ell P \cdot \widetilde{\ell P}, \widetilde{Q}; \tilde{0}, -\ell P \cdot \widetilde{\ell P+Q}]$. Then $-\ell P \cdot \widetilde{\ell P} = \lambda_{0,P}\tilde{0}$, $-\ell P \cdot \widetilde{\ell P+Q} = \lambda_{1,P}\widetilde{Q}$, so $-\ell P \cdot g_{P,Q}^{\star_1,\ell} = \lambda_P$ with $\lambda_P = \lambda_{1,P}/\lambda_{0,P}$. We will also denote $\lambda_P = \frac{(\widetilde{-\ell P})\cdot\widetilde{\ell P+Q}}{\widetilde{Q}} \frac{\tilde{0}}{(\widetilde{-\ell P})\cdot\widetilde{\ell P}}$.

If $Q \in A[\ell D]$ too, then the Weil pairing is (up to a sign) $e_{W,D,\ell}(P,Q) = \lambda_P/\lambda_Q$. If $P \in A[\ell D](k), Q \in A(k)$ and our cubical points are rational, the non reduced Tate pairing is (up to a sign) $e_{T,D,\ell}(P,Q) = \lambda_P$.

If $k = \mathbb{F}_q$, $\mu_\ell \subset \mathbb{F}_q$, and $P \in A[\ell]$, we can also recover the reduced Tate pairing by computing λ'_P such that $g_{P,Q}^{\star_1,q-1} = \lambda'_P$, or alternatively such that $g_{P,Q}^{\star_1,q} = \lambda'_P g_{P,Q}$. We have $\lambda'_P = \frac{(q-\widetilde{1})P+Q}{\widetilde{Q}} \frac{\tilde{0}}{(q-1)P} = \frac{q\widetilde{P}+Q}{\widetilde{P+Q}} \frac{\widetilde{Q}}{\widetilde{qP}}$.

The same remark as in Remark 2.11 still applies.

Remark 4.20 (Refined bilinearity)**.** We can extend Remark 3.14 as follows. Lets assume for simplicity that we are computing the Tate pairing on $\mathbb{G}_1 = E[\ell](\mathbb{F}_q) \times \mathbb{G}_2 \subset E[\ell](\mathbb{F}_{q^d})$, and that ℓ is prime to $q-1$, so that $d > 1$ where d is the embedding degree.

Then given $P \in \mathbb{G}_1$, by Section 4.2.5 there is a unique $\mathbb{F}_q$-rational cubical lift $\widetilde{P}$ which is ℓ-periodic, i.e., such that $(a\ell + b)\widetilde{P} = b\widetilde{P}$ for all $a, b \in \mathbb{Z}$ (we will come back to these canonical cubical points of ℓ-torsion in Sections 6.2 and 6.4.

To compute the Tate pairing $e_{T,\ell}(P,Q)$ with $Q \in \mathbb{G}_2$ (or even any point in $E[\ell](\mathbb{F}_{q^d})$, we can start with $\widetilde{P}$ as above and an arbitrary choice of $\widetilde{Q}, \widetilde{P+Q}$. Since $\ell\widetilde{P} = \tilde{0}$, the (non reduced) Tate pairing is given by the monodromy $e_{T,\ell}(P,Q) = \lambda_P \in \mathbb{F}_{q^d}^*$ with $\widetilde{\ell P+Q} = \lambda_P\widetilde{Q}$. If we look at the associated biextension elements, we have $g_{P,Q}^{\star_1,\ell} = \lambda_P$, so $g_{P,Q}^{\star_1,\ell+1} = \lambda_P g_{P,Q}$, hence since $\widetilde{\ell P+P} = \widetilde{P}$ by assumption on $\widetilde{P}$, we can also recover λ_P as $\ell P + \widetilde{P} + Q = \lambda_P \widetilde{P+Q}$.

Now by Example 4.10 (beware of the change of notations), we have $\ell u_0 P + \widetilde{u_1 P} + vQ = \lambda_P^{u_0 v} u_1 \widetilde{P+vQ}$. In particular, to compute the Tate pairing $e_{T,\ell}(iP,Q)$ (resp. $e_{T,\ell}(P,iQ)$), if we start with $\widetilde{Q}, \widetilde{iP+Q}$ (resp. $\widetilde{iQ}, \widetilde{P+iQ}$) as computed from $\widetilde{P}, \widetilde{Q}, \widetilde{P+Q}$ by the cubical arithmetic, then we have $\widetilde{\ell iP+Q} = \lambda_P^i \widetilde{Q}$ (resp. $\widetilde{\ell P+iQ} = \lambda_P^i \widetilde{iQ}$).

As in Porisms 2.10 and 3.10 we get:

Porism 4.21. *Let $g_{P,Q} = [\widetilde{P}, \widetilde{Q}; \tilde{0}, \widetilde{P+Q}]$, and use any of the two cubical biextension exponentiation to get a representation $g_{P,Q}^{\star_{1,\ell}} = [\widetilde{\ell P}, \widetilde{Q}; \tilde{0}, \widetilde{\ell P+Q}]$.*

Then the function $f_{\ell,P}$ evaluated on the cycle $(x+Q)-(x)$ is given by $\frac{g_{\ell P,Q}}{g_{P,Q}^{\ell}}$.

If furthermore $P \in A[\ell D]$, then $f_{-\ell D_P}((R+Q)-(R)) = \frac{(\widetilde{-\ell P})\cdot\widetilde{\ell P+Q+R}}{(\widetilde{-\ell P})\cdot\widetilde{\ell P+R}\widetilde{Q+R}} \left(\frac{\widetilde{P+R}\widetilde{Q+R}}{\widetilde{P+Q+R}}\right)^{\ell}$.

Here $\frac{\widetilde{P+R}\widetilde{Q+R}}{\widetilde{P+Q+R}}$ is not a monodromy information but a notation for the theta group function $g_{P,Q}$ represented by $\widetilde{P}, \widetilde{Q}$ evaluated at R; we refer to Section 4.5.2 and Equation (17) for more details.

We have a natural Galois action on cubical points: if σ is a Galois group element, $\sigma(\widetilde{P})$ is the rigidification of our rational line bundle $\mathcal{L}$ at $\sigma(P)$ given by applying σ to the rigidification of $\mathcal{L}$ at P.

In the context of the Ate and optimal Ate pairings, Proposition 3.17 and Corollary 3.19 then become:

Proposition 4.22 (Ate and optimal Ate pairings in the cubical representation)**.** *Let $A/\mathbb{F}_q$ be an abelian variety with embedding degree $d>1$ with respect to ℓ: $\mu_\ell \subset \mathbb{F}_{q^d}$. Let $\mathbb{G}_1, \mathbb{G}_2$ denotes the subspace of $A[\ell]$ where the Frobenius π_q has eigenvalues 1 and q respectively. Let $P \in \mathbb{G}_2$ and $Q \in \mathbb{G}_1$. Take any cubical biextension representation $(P,Q,g_{P,Q}) = [\widetilde{P}, \widetilde{Q}; \tilde{0}, \widetilde{P+Q}]$. Let $\lambda \equiv q \mod \ell$.*

Then $(P,Q,g_{P,Q})^{\star_{1,\lambda}} = [\lambda\widetilde{P}, \widetilde{Q}; \tilde{0}, \lambda\widetilde{P+Q}]$ and $\pi_q((P,Q,g_{P,Q})) = [\pi_q(\widetilde{P}), \pi_q(\widetilde{Q}); \pi_q(\tilde{0}), \pi_q(\widetilde{P+Q})]$ both represent biextension elements above (qP,Q). They differ by a monodromy constant which gives the (non reduced, except if $\lambda = q$) λ-Ate pairing.

Explicitly, assuming that $\widetilde{Q}, \tilde{0}$ are chosen to be $\mathbb{F}_q$-rational for simplicity,

$$a_{\lambda,\ell}(P,Q) = \frac{\widetilde{qP+Q}}{\widetilde{\pi_q(P+Q)}} \frac{\pi_q(\widetilde{P})}{\widetilde{qP}}.$$

If $\ell = \sum c_i q^i$, we can compute two cubical points $\widetilde{\ell P+Q}, \widetilde{\ell P}$ as follows:

1. *Compute the $\widetilde{c_i P+Q}, \widetilde{c_i P}$ via a cubical biextension exponentiation;*
2. *Compute $\widetilde{c_i q^i P+Q}, \widetilde{c_i q^i P}$ by applying π_q^i to $\widetilde{c_i P+Q}, \widetilde{c_i P}$;*
3. *Compute iteratively $Q + \widetilde{c_i q^i P + \sum_j c_j q^j P}$, $\widetilde{c_i q^i P + \sum_j c_j q^j P}$ by taking an arbitrary choice for $\widetilde{c_i q^i P + \sum_j c_j q^j P}$ and then using a cubical three way addition to compute $\widetilde{Q + c_i q^i P + \sum_j c_j q^j P}$ from $\widetilde{Q}, \widetilde{c_i q^i P}, \widetilde{\sum_j c_j q^j P}$ and the two by two sums $\widetilde{c_i q^i P + \sum_j c_j q^j P}$, $\widetilde{Q + c_i q^i P}, \widetilde{Q + \sum_j c_j q^j P}$.*

Then $[\widetilde{\ell P}, \widetilde{Q}; \tilde{0}, \widetilde{\ell P+Q}]$ is a biextension element above $(0,Q)$, hence represents a constant equal to $\frac{\widetilde{\ell P+Q}}{\tilde{0}} \frac{\widetilde{Q}}{\widetilde{\ell P}}$ which is the optimal Ate pairing.

Remark 4.23 (Cubical arithmetic versus biextension arithmetic)**.** The cubical arithmetic is a refinement of the biextension arithmetic, since we can recover the biextension arithmetic from it. On the other hand, since for pairings we only need the biextension arithmetic, and different cubical points can represent the same biextension points by Lemma 4.15, we have some leeways: even if we don't compute the correct cubical arithmetic, as long as the underlying biextension arithmetic is still correct, our pairings will be correct.

We have seen an example already when using the cubical biextension double and add to compute $g_{P,Q}^{\star_1,\ell}$. The points $\widetilde{\ell P}, \widetilde{\ell P + Q}$ we obtain in this way are not the correct one from the cubical point of view, but they differ from the correct ones by the same factor λ, which mean that the associated biextension element is still the correct one by Lemma 4.15.

An important example of the difference between the two arithmetic is with respect to the multiplication by elements $\lambda \in k$. In the biextension (or theta groups), we have $(\lambda \cdot g_{P,Q})^{\star_1,\ell} = \lambda^\ell \cdot g_{P,Q}^{\star_1,\ell}$. By contrast, we have $\ell(\lambda \cdot \widetilde{P}) = \lambda^{\ell^2}\widetilde{P}$.

One can use Lemma 4.5 to check directly that the Weil pairing and the class of the non reduced Tate pairing as computed in Theorem 4.19 does not depends on the choice of $\widetilde{P}, \widetilde{Q}, \widetilde{P+Q}$. We can also use that $\lambda \cdot [\widetilde{P}, \widetilde{Q}; \widetilde{P+Q}] = [\widetilde{P}, \widetilde{Q}; \tilde{0}, \widetilde{\lambda P + Q}]$ to recover that $g_{P,Q}^{\star_1,\ell} = \lambda^\ell \cdot g_{P,Q}^{\star_1,\ell}$.

Finally we remark that there are two special cases where we can do a biextension exponentiation faster using the cubical arithmetic.

The first one is when $\ell = 2^m$, in which case the biextension exponentiation consists entirely of biextension doublings, so we can use a cubical biextension ladder consisting entirely of cubical biextension doublings, i.e. a cubical doubling and a cubical differential addition at each step.

The other more subtle case is for self pairings, i.e. when we want to compute $g_{P,P}^{\star_1,\ell}$. Since we are allowed to choose any $g_{P,P}$ above (P,P) for pairings, we will take one of special form, given by $[\widetilde{P}, \widetilde{P}, \tilde{0}, \widetilde{2P}]$. In other words, we take for $\widetilde{2P}$ the cubical doubling $2\widetilde{P}$ rather than an arbitrary cubical points. Changing $\widetilde{P}$ to $\lambda \cdot \widetilde{P}$ changes $2\widetilde{P}$ to $\lambda^4 \cdot 2\widetilde{P}$, hence $g_{P,P}$ to $\lambda^2 \cdot g_{P,P}$. In other words, these special $g_{P,P}$ form a torsor under the squared action of $\mathbb{G}_m$ rather than under $\mathbb{G}_m$. Then we can compute $\ell\widetilde{P}, (\ell+1)\widetilde{P}$ via a cubical ladder, which involves one cubical doubling and one cubical differential addition by step, to get $g_{P,P}^{\star_1,\ell} = [\widetilde{P}, \ell\widetilde{P}, \tilde{0}, (\ell+1)\widetilde{P}]$. Here it is important to use the cubical arithmetic.

A last remark is that when using the cubical representation to compute different pairings $e(P,Q_i)$ with the same base point P, i.e. biextension exponentiation of elements of the form $g_{P,Q_i} = [\widetilde{P}, \widetilde{Q_i}; \tilde{0}, \widetilde{P+Q_i}]$, then for the representation of each $g_{P,Q_i}^{\star_1,\ell} = [\ell\widetilde{P}, \widetilde{Q_i}; \tilde{0}, \widetilde{\ell P + Q_i}]$, we can of course use the same $\ell\widetilde{P}$ and share the computation. This is also a well known pairing trick.

4.5. **The affine lift representation of cubical points.** To use the cubical arithmetic in order to obtain the biextension arithmetic in order to obtain our pairings, we need to find a convenient representation of our cubical points $\widetilde{P}$.

4.5.1. *Cubical coordinates.* In this section we introduce cubical coordinates for cubical points, which we also call the affine coordinates representation (or affine representation for short) because it lifts the projective representation of abelian varieties points. Fix $X_1, \dots, X_m \in \Gamma(\mathcal{L})$ global sections of $\mathcal{L} = O_A(D)$. We have a partial map $\varphi : A \to \mathbb{P}^{m-1}, P \mapsto (X_1(P) : \cdots, X_m(P))$, which is not defined at the base points of $\mathcal{L}$.

The element $X_i(P)$ is in the fiber $\mathcal{L}(P)$ of the line bundle $\mathcal{L}$, and $\widetilde{P}$ gives a rigidification $O_A(P) = \kappa(P) \to \mathcal{L}(P)$. Using this isomorphism, we obtain an element $X_i(\widetilde{P}) \in \kappa(P)$. This allows to interpret X_i as an affine coordinate on cubical points.

More explicitly, our cubical point $\widetilde{P}$ can be lifted to a local trivialisation ϕ_P of $\mathcal{L}$ at P, which is the same as a choice of a local generator s at P. So we can write $X_i = s_i \cdot s$, for some $s_i \in O_A$. We define the affine representation of our cubical point to be $\widetilde{\varphi}(\widetilde{P}) = (X_1(\widetilde{P}), X_m(\widetilde{P})) := (s_1(P), \dots, s_m(P)) \in \mathbb{A}^m$; it is easy to check that it depends only on $\widetilde{P}$. If P is not a base point, $\widetilde{\phi}(\widetilde{P})$ is an affine point lying above the projective point $\varphi(P)$. Furthermore, in this case not all $s_i(P)$ are 0, and our local section s is completely determined from any non zero coordinate of $\widetilde{\phi}(\widetilde{P})$: $\widetilde{P}$ is completely determined by $(P, \widetilde{\varphi}(\widetilde{P}))$, or even by $(P, X_i(\widetilde{P}))$ for any i such that $s_i(P) \neq 0$. If furthermore $\mathcal{L}$ is very ample, P is completely determined by $\phi(P)$, so $\widetilde{P}$ is completely determined by $\widetilde{\varphi}(\widetilde{P})$, in which case we will often denote it by $\widetilde{P}$ too.

We thus obtain a convenient representation as affine points of the cubical points. For instance, in Theorem 4.19, the value $\lambda_{1,P}$ is simply given as $\lambda_{1,P} = X_i(\widetilde{\ell P + Q})/X_i(\widetilde{Q})$ for any i such that $X_i(Q) \neq 0$. Furthermore, if $\widetilde{P}$ is represented by $(X_1(\widetilde{P}), \dots, X_m(\widetilde{P}))$, the Galois action $\sigma(\widetilde{P})$ of a Galois element σ on the rigidification $\widetilde{P}$ of $\mathcal{L}$ at P is given by $\sigma(\widetilde{P}) = (\sigma(X_1(\widetilde{P})), \dots, \sigma(X_m(\widetilde{P})))$. If $P \in A[\mathcal{L}]$, and $(P, g_P) \in G(\mathcal{L})$ is an element of the theta group above P, then the action of the theta group element g_P on $\widetilde{P}$ is given by $g_P \cdot \widetilde{P} = ((g_P \cdot X_1)(\widetilde{P}), \dots, (g_P \cdot X_m)(\widetilde{P}))$ where $g_P \cdot X_i$ is the natural action of $G(\mathcal{L})$ on sections $X_i \in \Gamma(\mathcal{L})$ from Equation (2).

We will call the cubical doublings, differential additions and three way additions in the affine lift representation of cubical points the affine doublings, affine differential additions and affine three way additions respectively.

Remark 4.24. There is a global $\mathbb{G}_m$-ambiguity when using affine coordinates X_i to represent cubical points. We'll illustrate this in the case of an elliptic curve: take a short Weierstrass equation $y^2 = x^3 + ax + b$, with associated Weierstrass coordinates x, y. The projective equation is $Y^2Z = X^3 + aXZ^2 + bZ^3$, and fixing the coordinate $Z \in \Gamma(3(0_E))$ fixes X, Y via $X = xZ, Y = yZ$. However, the projective curve equation does not change if we replace Z by λZ, so we could also work with $\lambda X, \lambda Y, \lambda Z$. Now, via the affine representation of cubical points by the coordiantes X, Y, Z, there is no difference between keeping the same cubical points and changing the coordinates X, Y, Z to $\lambda X, \lambda Y, \lambda Z$, or keeping the same coordinates X, Y, Z and changing all cubical points $\widetilde{P}$ into $\lambda\widetilde{P}$.

Let $\mathcal{L} = O_E((0_E))$ be the line bundle associated to the canonical principal polarisation on E. We will often say that we normalize $\tilde{0}$ so that $(Z/(x/y))(\tilde{0}) = 1$. This means that if we lift $\tilde{0}$ to a local trivialisation of $\mathcal{L}$ at 0_E, i.e. a local choice of section s; the section $Z \in \Gamma(\mathcal{L})$ can then be written locally as $Z = gs$ for some function $g \in k(E)$, and we require that $(g/(x/y))(0) = 1$. We can thus interpret this normalisation condition in two different ways: the first one is to fix a global section Z, and then ask to take $\tilde{0}$ such that $(Z/(x/y))(\tilde{0}) = 1$; the second one is to fix $\tilde{0}$ first and then take Z such that we have this same equality.

4.5.2. *Cubical functions for pairings computations.* Let $[\widetilde{P}_1, \widetilde{P}_2, \widetilde{P}_3, \widetilde{P}_4; \widetilde{Q}_1, \widetilde{Q}_2, \widetilde{Q}_3, \widetilde{Q}_4]$ be cubical points in Riemann relation. By definition and Proposition 4.1 they define a canonical function γ, which only depends on $[P_1, P_2, P_3, P_4; Q_1, Q_2, Q_3, Q_4]$.

We can recover γ in terms of the affine coordinates X_i as follows. Let D_i be the divisor of zeroes of X_i; since $X_i \in \Gamma(\mathcal{L})$ is a global section, the associated line bundle $O_A(D_i)$ is isomorphic to $\mathcal{L}$ and all the D_i are linearly equivalent. Since the points $[P_1, P_2, P_3, P_4; Q_1, Q_2, Q_3, Q_4]$.

are in Riemann position, they define canonical functions γ_i for each i. And we have, by definition of the affine coordinates, $\gamma_i(0) = \frac{X_i(\widetilde{P}_1)X_i(\widetilde{P}_2)X_i(\widetilde{P}_3)X_i(\widetilde{P}_4)}{X_i(\widetilde{Q}_1)X_i(\widetilde{Q}_2)X_i(\widetilde{Q}_3)X_i(\widetilde{Q}_4)}$.

Take $R \in A$, and consider the points $[P_1+R, P_2+R, P_3+R, P_4+R; Q_1+R, Q_2+R, Q_3+R, Q_4+R]$, they are still in Riemann position, and the associated function is simply $t_R^*\gamma_i$. So if $[\widetilde{P_1+R}, \widetilde{P_2+R}, \widetilde{P_3+R}, \widetilde{P_4+R}; \widetilde{Q_1+R}, \widetilde{Q_2+R}, \widetilde{Q_3+R}, \widetilde{Q_4+R}]$ are in Riemann position, we obtain:

$$\gamma_i(R) = \frac{X_i(\widetilde{P_1+R})X_i(\widetilde{P_2+R})X_i(\widetilde{P_3+R})X_i(\widetilde{P_4+R})}{X_i(\widetilde{Q_1+R})X_i(\widetilde{Q_2+R})X_i(\widetilde{Q_3+R})X_i(\widetilde{Q_4+R})}.$$

We can go further: fix cubical points $\tilde{0}, \widetilde{P}, \widetilde{Q}, \widetilde{P+Q}$, since the cycle $Z = (P+Q) + (0) - (P) - (Q)$ satisfy $s(Z) = 0$, these cubical points defines uniquely a function f_Z as in Section 4.2.7, which in this case is a biextension element $g_{P,Q} \in X_{\mathcal{L}}$ in the biextension associated to $\mathcal{L}$.

To each coordinate X_i, with zero divisor D_i, we let $g_{i,P,Q}$ be the function with divisor $D_{i,P+Q} - D_{i,P} - D_{i,Q}$ associated to $g_{P,Q}$ via the isomorphism $X_{\mathcal{L}} \simeq X_{D_i}$.

If $R \in A$, we can fix arbitrary cubical points $\widetilde{R}, \widetilde{R+P}, \widetilde{R+Q}$, and look at the unique cubical point $\widetilde{R+P}+Q$ given by the three way addition, i.e. such that $[\widetilde{R+P}+Q, \widetilde{R}, \widetilde{P}, \widetilde{Q}; \tilde{0}, \widetilde{P+Q}, \widetilde{R+Q}, \widetilde{P+Q}]$ are in Riemann position. By Theorem 4.16 combined with the same reasoning as above, we have

$$g_{i,P,Q}(R) = \frac{X_i(\widetilde{P+\widetilde{Q}+R})X_i(\widetilde{R})}{X_i(\widetilde{P+R})X_i(\widetilde{Q+R}),} \tag{17}$$

and so

$$g_{i,P,Q}((R)-(0)) = \frac{X_i(\widetilde{P+\widetilde{Q}+R})X_i(\widetilde{R})X_i(\widetilde{P})X_i(\widetilde{Q})}{X_i(\widetilde{P+R})X_i(\widetilde{Q+R})X_i(\widetilde{P+Q})X_i(\tilde{0})}.$$

In particular, $g_{i,P,Q}$ does not depend on the choice of $\widetilde{R}, \widetilde{R+P}, \widetilde{R+Q}$, as long as $\widetilde{R+P}+Q$ is computed through a three way addition, as can be checked directly by homogeneity.

This allows to write the genuine function $g_{i,P,Q}$ as a product of "cubical functions".

If $T \in A[D_i]$, we have also seen in Section 4.2 how a choice of cubical point $\widetilde{T}$ gives a canonical theta group element $(T, g_{i,T}) \in G(D_i)$. It can be described as follows: if $R \in A$, take an arbitrary cubical point $\widetilde{R}$, and let $\widetilde{R+T}$ be given by the action of $\widetilde{T}$ on $\widetilde{R}$. Then by definition of this action, $g_{i,T}(R) = X_i(\widetilde{R+T})/X_i(\widetilde{R})$.

For a degree 0 cycle Z such that $s(Z) \in \operatorname{Ker}\Phi_{\mathcal{L}}$, and a cubical lift $\widetilde{Z}$ of Z, we can combine the two methods to iteratively reduce $\widetilde{Z}$ to a cycle $\widetilde{s(Z)} - \tilde{0}$ and express the function $f_{\widetilde{Z}}$ as product of cubical functions.

4.5.3. *The affine cubical biextension representation as evaluation of cubical functions.* Let $(P, Q, g_{P,Q}) \in X_D$ be a biextension element. In the evaluation representation, we represent $g_{P,Q}$ via (the extended value) $g_{P,Q}(0)$ at the base point 0. Let $X_1, \dots, X_m \in \Gamma(D)$, and assume that X_1 has D for divisor of zeroes, X_i has D_i for divisor of zeroes, with $D_i = D + \operatorname{div}(X_i/X_1)$. We have a function $g_{i,P,Q} = g_{P,Q}^{\frac{X_i}{X_1}((\cdot+P+Q)+(\cdot)-(\cdot+P)-(\cdot+Q))}$ coming from the biextension isomorphism $X_D \simeq X_{D_i}$ (see Remark 3.8). We can then define a multievaluation representation, representing $g_{P,Q}$ via the (extended) evaluations $g_{i,P,Q}(0)$. It is often the case that the functions X_i are determined from X_1 via translation by some elements of torsion T (or more precisely via the action of some $(T, g_T) \in G(D)$ on X_1 in the theta group). This is for instance the case for theta functions of level n (where the T are points of n-torsion), or for the Montgomery model of the Kummer line where $T = (0:1)$ and

$g_T \cdot (X, Z) = (Z, X)$. In this case the multievaluation representation is simply the evaluation of $g_{P,Q}$ not only at the base point $R_0 = 0$, but also at the points $R_0 + T$.

Now assume that we are given a cubical representation $g_{P,Q} = [\widetilde{P}, \widetilde{Q}; \tilde{0}, \widetilde{P+Q}]$. By Section 4.5.2, given $R \in A$, and any choice of $\widetilde{R}$ above R, $\widetilde{R+P}$, $\widetilde{R+Q}$, computing $P + \widetilde{Q} + R$ via a cubical three way add, we have

$$g_{i,P,Q} : R \mapsto \frac{X_i(P + \widetilde{Q} + R)X_i(\widetilde{R})}{X_i(\widetilde{P+R})X_i(\widetilde{Q+R})}. \tag{18}$$

Now, although the function $g_{i,P,Q}$ is a genuine function on our abelian variety, its individual members $\widetilde{R} \mapsto X_i(P + \widetilde{Q} + R), X_i(\widetilde{R}), X_i(\widetilde{P+R}), X_i(\widetilde{Q+R})$ only make sense as virtual cubical functions, whose associated divisor of zeroes are $t^*_{P+Q}D_i$, D_i, $t^*_P D_i$ and $t^*_Q D_i$ respectively. And the affine cubical representation is precisely the evaluation of these virtual functions at 0.

When doing a biextension exponentiation to compute $g^{\star_1,\ell}_{P,Q}$, in the multievaluation representation we obtain functions $g_{i,\ell P,Q}$ such that by Porism 3.10 $g_{i,\ell P,Q}/g^\ell_{i,P,Q}$ is the function $f_{i,\ell,P}$ evaluated on the cycle $(x+Q)-(x)$, where $f_{i,\ell,P}$ has for divisor $D_{i,\ell P} + (\ell-1)D_i - \ell D_{i,P}$. The multievaluation representation is given by the evaluation of the functions $g_{i,\ell P,Q}$ at 0, and so $g_{i,\ell P,Q}(0)/g^\ell_{i,P,Q}(0)$ is the value of $f_{i,\ell,P}$ at the cycle $(Q) - (0)$.

Now, the cubical representation also allows to write

$$g_{i,\ell P,Q}(R) = \frac{X_i(\ell P + \widetilde{Q} + R)X_i(\widetilde{R})}{X_i(\widetilde{\ell P + R})X_i(\widetilde{Q+R})}. \tag{19}$$

In particular, we have that the evaluation of $f_{i,\ell,P}$ at the cycle $(R + Q) - (R)$ is given by

$$\frac{X_i(\ell P + \widetilde{Q} + R)X_i(\widetilde{R})}{X_i(\widetilde{\ell P + R})X_i(\widetilde{Q+R})} \left(\frac{X_i(\widetilde{P+R})X_i(\widetilde{Q+R})}{X_i(P + \widetilde{Q} + R)X_i(\widetilde{R})} \right)^\ell, \tag{20}$$

(compare with Example 4.29) and if $P \in A[\ell D]$, and $\widetilde{\ell P}$ is an arbitrary cubical point above ℓP,

$$f_{\ell D_{i,P}}((R+Q)-(R)) = \frac{X_i(-\widetilde{\ell P} \cdot \ell P + \widetilde{Q} + R)X_i(\widetilde{R})}{X_i(-\widetilde{\ell P} \cdot \widetilde{\ell P + R})X_i(\widetilde{Q+R})} \left(\frac{X_i(\widetilde{P+R})X_i(\widetilde{Q+R})}{X_i(P + \widetilde{Q} + R)X_i(\widetilde{R})} \right)^\ell. \tag{21}$$

And, although the function $g_{i,\ell P,Q}$ is a genuine function on our abelian variety, its individual members $(\widetilde{R}, \widetilde{P+R}, \widetilde{Q+R}) \mapsto X_i(\ell P + \widetilde{Q} + R), X_i(\widetilde{R}), X_i(\widetilde{\ell P + R}), X_i(\widetilde{Q+R})$ only make sense as cubical functions, with divisors of zeroes given by $t^*_{\ell P+Q}D_i$, D_i, $t^*_{\ell P}D_i$ and $t^*_Q D_i$ respectively.

Thus the cubical representation is a way to decompose the functions $g_{i,\ell P,Q}$ as a product of cubical functions, and the affine lift representation is simply the evaluation of these cubical functions at the cubical point $\tilde{0}$, hence is a way to decompose the functions evaluations $g_{i,\ell P,Q}(0)$ as a product of cubical evaluations.

And Porism 4.21 shows that the cubical arithmetic is an efficient way to embed the Miller functions $f_{\ell,P}$ in the affine coordinates of $\widetilde{\ell P}, \widetilde{\ell P + Q}$. Explicitly, evaluating Equations (20) and (21) at $\tilde{0}$:

Porism 4.25. *Let D_i be the divisor of zeroes of X_i (since X_i is a section of D, $D_i \sim D$).*

Let $f_{i,\ell,P}$ be a function with divisor $D_{i,\ell P} - \ell D_{i,P}$. Then the function $f_{i,\ell,P}$ evaluated on the cycle $(Q) - (0)$ is given by $\frac{X_i(\ell\widetilde{P+Q})X_i(\tilde{0})}{X_i(\widetilde{\ell P})X_i(\widetilde{Q})}\left(\frac{X_i(\widetilde{P})X_i(\widetilde{Q})}{X_i(\widetilde{P+Q})X_i(\tilde{0})}\right)^{\ell}$.

If $P \in A[\ell D], f_{\ell D_{i,P}}((Q) - (0)) =$ $\frac{X_i(-\widetilde{\ell P}\cdot\ell\widetilde{P+Q})X_i(\tilde{0})}{X_i(-\widetilde{\ell P}\cdot\widetilde{\ell P})X_i(\widetilde{Q})}\left(\frac{X_i(\widetilde{P})X_i(\widetilde{Q})}{X_i(\widetilde{P+Q})X_i(\tilde{0})}\right)^{\ell}$.

Remark 4.26 (Cubical coordinates for monodromy)**.** In Theorem 4.19, we compute $\lambda_P = \frac{(\widetilde{-\ell P})\cdot\ell\widetilde{P+Q}}{\widetilde{Q}}\frac{\tilde{0}}{(\widetilde{-\ell P})\cdot\widetilde{\ell P}}$, which is a quotient of two monodromy information: $\widetilde{-\ell P}\cdot\widetilde{\ell P} = \lambda_{0,P}\tilde{0}$, $\widetilde{-\ell P}\cdot\ell\widetilde{P+Q} = \lambda_{1,P}\widetilde{Q}$, $\lambda_P = \lambda_{1,P}/\lambda_{0,P}$.

We can use different cubical coordinates to compute these two monodromy informations; if for instance $X_0(0) \neq 0$ and $X_1(Q) \neq 0$, we have:

$$\lambda_P = \frac{X_1((\widetilde{-\ell P})\cdot\ell\widetilde{P+Q})}{X_1(\widetilde{Q})}\frac{X_0(\tilde{0})}{X_0((\widetilde{-\ell P})\cdot\widetilde{\ell P})}.$$

Remark 4.27. When using Porism 4.25 and Theorem 4.19 to evaluate the extended Tate pairing as in Remark 2.11 when $D = mD_1$, one need to be careful about the choice of the coordinate X_i.

Indeed, using X_i amount to computing the extended Tate pairing associated to D_i, but we have seen in Remark 2.11 that for the extended Tate pairing we need to be sure to choose D_i such that $D_i = mD_1'$, with D_1' rational. This won't be the case in general for the zero divisor D' of an arbitrary section $X \in \Gamma(O_A(D))$.

On the other hand, if X_1 has zero divisor $D = mD_1$, then we have seen in Remark 2.11 that we do not need to use the correcting factor $g_{P,Q}^{\ell}$ to correct the monodromy information λ_P when $g_{P,Q}$ comes from a m-th tensor power of a rational biextension element in X_{D_1}. An easy way to ensure this is to start with $g_{P,Q} = [\widetilde{P}, \widetilde{Q}, \tilde{0}, \widetilde{P+Q}]$ where the cubical points are normalised via $X_1(\widetilde{P}) = X_1(\widetilde{Q}) = X_1(\tilde{0}) = X_1(\widetilde{P+Q}) = 1$. Indeed, if Y_1 is a section of $O_A(D_1')$, then $X_1' = Y_1^m$ is a m-th power. Our X_1 is of the form μX_1 for some scalar factor, but this scalar factor gives the same $g_{P,Q}$ by Lemma 4.15.

For the convenience of the reader, let us summarise the whole discussion and reformulate Theorem 4.19 using Porism 4.25; this is our last reformulation. We remark that, compared to Theorem 4.19, this the cubical function approach of pairings hides somewhat the monodromy interpretation, but by Remark 4.26 this monodromy interpretation is still useful to change cubical coordinate on the fly.

Theorem 4.28. *Let $\mathcal{L}$ be a symmetric line bundle, and $X \in \Gamma(\mathcal{L})$ a section with zero divisor D.*

We fix a cubical point $\tilde{0}$ above 0_A once and for all, and for $P, Q, R \in A$, we fix cubical points $\widetilde{P}, \widetilde{Q}, \widetilde{R}, \widetilde{P+Q}, \widetilde{P+R}, \widetilde{Q+R}$. Then if $g_{P,Q} \in X_D$ is the biextension function with divisor $D_{P+Q} - D_P - D_Q$ encoded by $[\widetilde{P}, \widetilde{Q}; \tilde{0}, \widetilde{P+Q}]$, we have

$$g_{P,Q}(R) = \frac{X(P+\widetilde{Q}+R)X(\widetilde{R})}{X(\widetilde{P+R})X(\widetilde{Q+R})},$$

where $P+\widetilde{Q}+R$ is given by the cubical three way addition.

Using the cubical arithmetic to compute $\ell\widetilde{P}$, $\ell P+\widetilde{Q}+R$, $\ell\widetilde{P+R}$, if $g_{\ell P,Q} = g_{P,Q}^{\star_{1,\ell}}$, we also have

$$g_{\ell P,Q}(R) = \frac{X(\ell P+\widetilde{Q}+R)X(\widetilde{R})}{X(\ell\widetilde{P+R})X(\widetilde{Q+R})}.$$

Input: $P, Q \in A[\ell D]$
Output: The Weil pairing $e_{W,\ell D}(P,Q)$

- ➔ Take arbitrary affine cubical lifts $\widetilde{P}, \widetilde{Q}, \widetilde{P+Q}$ of $P, Q, P+Q$ (along with a choice of $\tilde{0}$).
- ➔ Use Algorithm 4.2 or Algorithm 4.1 to compute $\ell\widetilde{P+Q}, \widetilde{\ell P}$.
- ➔ Take an arbitrary lift $\widetilde{-\ell P}$, and compute the theta group action $\widetilde{-\ell P} \cdot \ell\widetilde{P+Q}, \widetilde{-\ell P} \cdot \widetilde{\ell P}$.
- ➔ Compute the monodromy $\widetilde{-\ell P} \cdot \ell\widetilde{P+Q} = \lambda_{1,P}\widetilde{Q}$, $\widetilde{-\ell P} \cdot \widetilde{\ell P} = \lambda_{0,P}\tilde{0}$.
- ➔ Similarly, compute the monodromy $\widetilde{-\ell Q} \cdot \ell\widetilde{Q+P} = \lambda_{1,Q}\widetilde{P}$, $\widetilde{-\ell Q} \cdot \widetilde{\ell Q} = \lambda_{0,Q}\tilde{0}$.
- ➔ Return $e_{\ell D}(P,Q) = \frac{\lambda_{1,P}}{\lambda_{0,P}}\frac{\lambda_{0,Q}}{\lambda_{1,Q}}$

Algorithm 4.3. Weil pairing via cubical arithmetic

And the function

$$f_{\ell,P}(R) = \frac{X(\ell\widetilde{P+R})X(\widetilde{R})^{\ell-1}}{X(\widetilde{P+R})^{\ell}}$$

has for divisor $D_{\ell P} - \ell D_P$.

All these functions only depend on $\widetilde{P}, \widetilde{Q}, \widetilde{P+Q}$, *not on the choices of* $\widetilde{R}, \widetilde{P+R}, \widetilde{Q+R}$.

If $P \in A[\ell D]$ *and* $f_{\ell D_P}$ *is any function with divisor* ℓD_P, *fixing a cubical point* $\widetilde{-\ell P}$, *we have*

$$f_{\ell D_P}((Q+R)-(R)) = \frac{X(\widetilde{-\ell P} \cdot \ell\widetilde{P} + \widetilde{Q+R})}{X(\widetilde{Q+R})} \frac{X(\widetilde{R})}{X(\widetilde{-\ell P} \cdot \ell\widetilde{P+R})} \left(\frac{X(\widetilde{P})X(\widetilde{Q})}{X(\widetilde{P+Q+R})X(\widetilde{R})}\right)^{\ell};$$

this does not depend on any choice of cubical points.

If $D = mD_1$, $P \in A[\ell D](\mathbb{F}_q)$, $Q \in A(\mathbb{F}_q)$, *the Tate pairing is given by* $e_{T,\ell m D_1}(P,Q) = f_{\ell D_P}((Q+R)-(R))$ *for any* $R \in A(\mathbb{F}_q)$.

If $D = mD_1, P, Q \in A[\ell D]$, *the Tate pairing is given by* $e_{W,\ell m D_1}(P,Q) = \frac{f_{\ell D_P}((Q+R)-(R))}{f_{\ell D_Q}((Q+R)-(R))}$ *for any* $R \in A$.

Theorem 4.28 give Algorithms 4.3 to 4.6 for the Weil pairing, the Tate pairing, the Ate pairing and the optimal Ate pairing respectively.

For the Weil and Tate pairing we treat the general case on how, when $D = mD_1$, we can use the theta group action of $G(D)$ to recover the pairings of level ℓm with respect to D_1 while working with the cubical arithmetic for D_1.

For the Ate and optimal Ate pairing we describe the easier case $P \in \mathbb{G}_2 \subset A[\ell]$; but we could use the functions constructed in Theorem 4.28 to treat the general case too. In practice the Ate and optimal Ate pairings are usuall for ℓ odd, and for pairings in the Montgomery model we have $D = 2(0_E)$ so $m = 2$ is prime to ℓ, so there is no need to treat this general case.

4.5.4. *The naturality of cubical arithmetic.* We can give the following philosophical motivation for using the affine representation to compute pairings. By Theorem 4.19, we need to compute cubical points of the form $\widetilde{\ell P}, \ell\widetilde{P+Q}$. In particular, we need to compute the standard points $\ell P, \ell P + Q$. For this last computation, we want to use projective coordinates to avoid divisions. But for the algorithm, the way we represent the projective coordinates $(X_1 : \ldots : X_m)$ is via affine coordinates $(X_1, \ldots, X_m)$ anyway. If the way we do the projective arithmetic in terms of affine coordinates is already close to the cubical arithmetic, we can

Input: $P \in A[\ell D](\mathbb{F}_q), Q \in A(\mathbb{F}_q), X \in \Gamma(D)$ with divisor of zeroes equal to mD_1
Output: The non reduced Tate pairing of level ℓm $e_{T,\ell m D_1}(P,Q)$

- Take arbitrary rational affine cubical lifts $\widetilde{P}, \widetilde{Q}, \widetilde{P+Q}$ of $P, Q, P+Q$ (along with a choice of $\tilde{0}$).
- Use Algorithm 4.2 or Algorithm 4.1 to compute $\ell\widetilde{P+Q}, \widetilde{\ell P}$.
- Take an arbitrary lift $\widetilde{-\ell P}$, and compute the theta group action $\widetilde{-\ell P} \cdot \ell\widetilde{P+Q}, \widetilde{-\ell P} \cdot \widetilde{\ell P}$.
- Compute the monodromy $\widetilde{-\ell P} \cdot \ell\widetilde{P+Q} = \lambda_{1,P}\widetilde{Q}, \widetilde{-\ell P} \cdot \widetilde{\ell P} = \lambda_{0,P}\tilde{0}$.
- Return $e_{\ell m D_1}(P,Q) = \frac{X(\widetilde{P})X(\widetilde{Q})}{X(\tilde{0})X(\widetilde{P+Q})}\frac{\lambda_{1,P}}{\lambda_{0,P}}$

Algorithm 4.4. Tate pairing via cubical arithmetic

Input: $P \in \mathbb{G}_2, Q \in \mathbb{G}_1, \lambda \equiv q \mod \ell$
Output: The non reduced Ate pairing (of level m) $a_{\lambda,\ell}(P,Q)$

- Take arbitrary rational affine cubical lifts of level m $\widetilde{P}, \widetilde{Q}, \widetilde{P+Q}$ of $P, Q, P+Q$ (along with a choice of $\tilde{0}$).
- Use Algorithm 4.2 or Algorithm 4.1 to compute $\lambda\widetilde{P+Q}, \widetilde{\lambda P}$.
- Compute the monodromy $\lambda\widetilde{P+Q} = \lambda_{1,P}\pi_q(\widetilde{P+Q}), \widetilde{\lambda P} = \lambda_{0,P}\pi_q(\widetilde{P})$.
- Return $a_{\lambda,\ell}(P,Q) = \frac{\lambda_{1,P}}{\lambda_{0,P}}$

Algorithm 4.5. Ate pairing via cubical arithmetic

Input: $P \in \mathbb{G}_2, Q \in \mathbb{G}_1, \ell = \sum c_i q^i$
Output: The optimal Ate pairing (of level m) $a_{\sum c_i q^i}(P,Q)$

- Take arbitrary rational affine cubical lifts of level m $\widetilde{P}, \widetilde{Q}, \widetilde{P+Q}$ of $P, Q, P+Q$ (along with a choice of $\tilde{0}$).
- Use Algorithm 4.2 or Algorithm 4.1 to compute the $c_i\widetilde{P+Q}, \widetilde{c_i P}$.
- Apply Frobenius to compute the $\pi_q^i(c_i\widetilde{P+Q}), \pi_q^i(\widetilde{c_i P})$.
- Use cubical Three Way Additions to compute the $\sum \pi_q^i(c_i\widetilde{P+Q}), \sum \pi_q^i(\widetilde{c_i P})$.
- Compute the monodromy $\sum \pi_q^i(c_i\widetilde{P+Q}) = \lambda_{1,P}\widetilde{Q}, \sum \pi_q^i(\widetilde{c_i P}) = \lambda_{0,P}\tilde{0}$.
- Return $a_{\sum c_i q^i}(P,Q) = \frac{\lambda_{1,P}}{\lambda_{0,P}}$

Algorithm 4.6. Optimal ate pairing via cubical arithmetic

easily correct our algorithm to do a cubical exponentiation rather than a projective point exponentiation. Our exponentiation $\ell P, \ell P + Q$, then gives "for free" our cubical points $\widetilde{\ell P}, \widetilde{\ell P + Q}$, hence our pairings. Now in principle, there is no reason that the random affine arithmetic we use when computing $\ell P, \ell P + Q$ has any reason to be close to the cubical arithmetic; after all at each step we could multiply all our coordinates by some random constant since this does not change the projective point. But in practice, we use efficient algorithms (which do not involve doing random multiplications at each step). And the amazing thing is

that the ladder algorithms we already use, for the Theta, Montgomery, and short Weierstrass models, are already (almost) the correct cubical ones already, as we will see in Section 5. One explanation for this is the unicity of biextensions from Theorem 3.1, it appears that efficient formulas are sufficiently functorial in nature to satisfy the biextension arithmetic, or are close to.

In particular, in Section 5, we will look at the cubical representation on a Kummer line $E/\pm 1$ resulting from affine lifts of sections $X, Z \in \Gamma(2(0_E))$. A biextension element will be given by $[\widetilde{P}, \widetilde{Q}, \tilde{0}, \widetilde{P+Q}]$ and the biextension exponentiation will be determined by $\widetilde{\ell P}$, $\widetilde{\ell P+Q}$ which are both described by two affine coordinates.

In Miller standard algorithm, one also compute the multiples ℓP of P (projectively, so with two projective coordinates $(X(\ell P) : Z(\ell P))$ if $P \in E/\pm 1$ is a Kummer point), and store the Miller evaluations $f_{\ell,P}(Q)$ as a numerator and denominator separately, i.e., as an element of $\mathbb{P}^1$.

In fine, the affine lift representation is very similar: we store $X(\widetilde{\ell P})$, $Z(\widetilde{\ell P})$, $X(\widetilde{\ell P+Q})$, $Z(\widetilde{\ell P+Q})$ as affine coordinates, in a way such that $f_{2(0_E),\ell,P}$ is encoded by (see Porism 4.25) $f_{2(0_E),\ell,P}((Q)-(0)) = \frac{Z(\widetilde{\ell P+Q})Z(\tilde{0})}{Z(\widetilde{\ell P})Z(\widetilde{Q})}\left(\frac{Z(\widetilde{P})Z(\widetilde{Q})}{Z(\widetilde{P+Q})Z(\tilde{0})}\right)^{\ell}$. A word on how to interpret this last equality. The function $f_{2(0_E),\ell,P}$ has for divisor $2(\ell P)+2(\ell-1)(0_E)-2\ell(P)$, so its evaluation on the cycle $(Q)-(0)$ gives a pole of order $2(\ell-1)$ at 0. On the other hand $0_E = (1:0)$ so $Z(0_E) = 0$ is a zero of order two, and the right hand side also gives a pole of order $2(\ell-1)$ at 0. The equality above makes sense by dividing both sides by $\pi_{0_E}^{2(\ell-1)}$ for any uniformizer π_{0_E} at 0_E; for instance divide both members by $Z^{\ell-1}$.

4.6. **Cubical functions.** Although we won't need this for pairings, we can generalize the construction of the cubical functions from Section 4.5.2 as follows, using the strategy of [Stao8, § 10.3].

Recall from Section 4.2.7 that if $Z = \sum n_i(Q_i)$ is a degree zero cycle such that $s(Z) = \sum n_i Q_i = 0$, then the choice of a cubical cycle $\widetilde{Z} = \sum n_i(\widetilde{Q_i})$ is enough to competlely determine a function $f_{\widetilde{Z}}$ with divisor $\sum n_i D_{Q_i}$.

A way to specify $\widetilde{Z}$ is as follows: we consider a cycle of cycles: $\mathcal{Z} = \sum n_i(Z_i)$, where $Z_i = \sum_j n_{ij}(P_{ij})$, and we will explain how to construct a cubical cycle $\widetilde{Z}$ above the cycle $Z = \sum n_i Z_i = \sum_{i,j} n_i n_{ij}(P_{ij})$, provided that Z is of degree 0 with $s(Z) = 0$ (in fact we will need a stronger condition $S(Z) = 0$ to be introduced below).

We suppose that we have fixed once and for all cubical points $\widetilde{P_{ij}}$, $P_{ij}\widetilde{+P_{i'j'}}$. By Section 4.2.5, this completely determines cubical points $\sum n_{ij}\widetilde{P_{ij}}$, hence cubical points $\widetilde{Q_i}$ where $Q_i = s(Z_i)$, hence a cubical cycle $\widetilde{Z} = \sum n_i\widetilde{Q_i}$. Hence we have a well defined function $f_{\widetilde{Z}}$, which depends on the choices of $\widetilde{P_{ij}}$, $P_{ij}\widetilde{+P_{i'j'}}$ above. (The whole discussion could extends to when $s(Z) \in A[D]$ provided we have fixed $\widetilde{s(Z)}$; as a notation we could use $\mathcal{Z} = \sum n_i\widetilde{T_i}\cdot Z_i$ to encode the cycle $Z = \sum n_i\widetilde{T_i}\cdot\widetilde{Q_i}$.)

We have $s(Z) = \sum n_i Q_i = \sum_{i,j} n_i n_{i,j} P_{i,j} \in A$. We let $S(\mathcal{Z}) = \sum_{i,j} n_i n_{i,j} P_{i,j}$ as seen in the free abelian group generated by the $P_{i,j}$. We have of course $S(Z) = 0 \Rightarrow s(Z) = 0$.

When $\mathcal{Z}$ is of degree 0 with $S(\mathcal{Z}) = 0$, we can compute the function associated to $\widetilde{Z}$ as follows. For a point R, take any cubical point $\widetilde{R}$, $R\widetilde{+P_{ij}}$. These choices completely determine

$R + \widetilde{\sum_j} n_{ij}P_{ij}$, hence cubical points $R\widetilde{+Q}_i$. We have:

$$f_{D_m,\widetilde{Z}}(R) = \prod_i X_m(R + \widetilde{\sum_j} n_{ij}P_{ij})^{n_i}.$$

As a reformulation, $f_{D_m,\widetilde{Z}}(R)$ is given by the coordinate X_m evaluated on the cubical cycle $\widetilde{Z}_R$ encoded by $t_{R,*}\mathcal{Z} = \sum_i n_i(R + \sum_j n_{i,j}P_{i,j})$.

Since $\mathcal{Z}$ is a cycle of degree 0 with $S(Z) = 0$, by Section 4.2.5, the value $f_{D_m,\widetilde{Z}}(R)$ only depends on $\widetilde{P_{ij}}$, $P_{ij}\widetilde{+P_{i'j'}}$, but not on the choices of $\widetilde{R}$, $R\widetilde{+P_{ij}}$.

More generally, let $Y = \sum_l m_l(R_l)$ be a cycle. Then we can consider the cycle of cycles $t_{Y_*}\mathcal{Z} = \sum_{i,l} n_i m_l(R_l + Z_i)$, and apply the same strategy to obtain:

$$f_{D_m,\widetilde{Z}}(Y) = \prod_{i,l} X_m(R_l + \widetilde{\sum_j} n_{ij}P_{ij})^{n_i m_l}.$$

This only depends on the choice of $\widetilde{P_{ij}}$, $P_{ij}\widetilde{+P_{i'j'}}$, but not on the choices of $\widetilde{R_l}$, $R_l\widetilde{+P_{ij}}$. And by Lemma 4.7, if $\deg(Y) = 0$ then this does not depend on any choices.

Example 4.29. Let X be a section of $\mathcal{L}$ with divisor D.

- The functions $g_{P,Q}$ with divisor $D_{P+Q} - D_P - D_Q$ from Section 4.5.3 corresponds to $\mathcal{Z} = ((P) + (Q)) + ((0)) - ((P)) - ((Q))$.
- We can also build functions $g_{\ell P,Q}$ with divisors $D_{\ell P+Q} - D_{\ell P} - D_Q$ by considering $\mathcal{Z} = (\ell(P) + (Q))) + ((0)) - (\ell(P)) - ((Q))$, we have seen in Section 4.5.3 that this is exactly $g^{\star_1,\ell}_{i,P,Q}$.
- The cycle of cycles $\mathcal{Z} = (\ell(P)) + (\ell-1)((0)) - \ell((P))$ gives a function with divisor $D_{\ell P} - \ell D_P$.

  If $P \in A[\ell D]$, then fixing a cubical point $\widetilde{\ell P}$ gives a function $g_{\ell P}$ with divisor $D_{\ell P}$, such that $g_{\ell P}(R) = X(\frac{\widetilde{\ell P}+\widetilde{R}}{\widetilde{R}})$ (by the definition of the action of $G(D)$ on cubical points), and we could say that $\mathcal{Z} = (\ell(P)) + (\ell-1)((0)) - \ell((P)) - (\widetilde{\ell P})$ gives a function with divisor $-\ell D_P$.
- If $P \in A[\ell]$, $\mathcal{Z} = \ell((P)) - \ell((0))$ encodes a cubical cycle $\widetilde{Z} = \ell\widetilde{P} - \ell\tilde{0}$, but while $s(Z) = 0$, $S(\mathcal{Z}) = \ell P \neq 0$ in the free group generated by P, so we cannot simply construct the associated function as $X(R\widetilde{+P})^\ell / X(\widetilde{R})^\ell$; this would depend on the choices of $\widetilde{R}$ and $R\widetilde{+P}$.

  We really need to encode the fact that P is of ℓ-torsion in $\mathcal{Z}$ itself, i.e. to rewrite it as $\mathcal{Z} = (\ell(P)) + (\ell-1)((0)) - \ell((P))$, as above.

Example 4.30. Sometimes, the function $f_{\mathcal{Z}}$ does not even depend on the choices of $\widetilde{P_{ij}}$, $P_{ij}\widetilde{+P_{i'j'}}$, and we can use Section 4.2.5 to conveniently check if that is the case.

- Lets consider $\mathcal{Z} = ((P) + (Q)) + ((0)) - ((P)) - ((Q))$ from Example 4.29 again. We have $(x_P + x_Q)^2 - x_P^2 - x_Q^2 = 2x_P x_Q$, so changing $\widetilde{P}$ to $\lambda_P\widetilde{P}$, $\widetilde{Q}$ to $\lambda_Q\widetilde{Q}$, $P\widetilde{+Q}$ to $\lambda_P\lambda_Q\lambda'_{PQ}P\widetilde{+Q}$, changes the resulting function $f_{\mathcal{Z}}$ by a factor λ'_{PQ}. In fact we saw in Example 4.29 that we recover exactly the way to associate a biextension function $g_{P,Q}$ to our cubical points $\widetilde{P}, \widetilde{Q}, P\widetilde{+Q}$.
- If $\mathcal{Z} = ((P) + (Q) + (R)) + ((P)) + ((Q)) + ((R)) - ((P) + (Q)) - ((P) + (R)) - ((Q) + (R))$, then we have $(x_P + x_Q + x_R)^2 + x_P^2 + x_Q^2 + x_R^2 - (x_P + x_Q)^2 - (x_P + x_R)^2 - (x_Q + x_R)^2 = 0$, so $f_{\mathcal{Z}}$ does not depend on our choice of

cubical points. In fact, this is precisely the function associated to cubical three way addition.

- If $\mathcal{Z} = ((P)+(Q))+((P)-(Q))+((0))+((0))-(-(Q))-((Q))-((P))-((P))$, then since $(x_P+x_Q)^2+(x_P-x_Q)^2-(-x_Q)^2-(x_Q)^2-(x_P)^2-(x_P)^2$, then $f_{\mathcal{Z}}$ also does not depend on the choice of cubical points; this is the function associated to differential additions.
- If $[P_1,P_2,P_3,P_4;Q_1,Q_2,Q_3,Q_4]$ are in Riemann position, so that $Q_i = T - P_i$ with $2T = P_1+P_2+P_3+P_4$, we can always find (possibly over a field extension) points P_i' such that $2P_i' = P_i$ and $T = P_1'+P_2'+P_3'+P_4'$, so that $Q_1 = -P_1'+P_2'+P_3'+P_4'$ and so on.

  Consider $\mathcal{Z} = (2(P_1'))+(2(P_2'))+(2(P_3'))+(2(P_4'))-(-(P_1')+(P_2')+(P_3')+(P_4'))-((P_1')-(P_2')+(P_3')+(P_4'))-((P_1')+(P_2')-(P_3')+(P_4'))-((P_1')+(P_2')+(P_3')-(P_4'))$. Then $f_{\mathcal{Z}}$ also does not depend on the choice of cubical points, and actually gives the general Riemann relation on cubical points, because $(2x_{P_1'})^2+(2x_{P_2'})^2+(2x_{P_3'})^2+(2x_{P_4'})^2+(-x_{P_1'}+x_{P_2'}+x_{P_3'}+x_{P_4'})^2-(x_{P_1'}-x_{P_2'}+x_{P_3'}+x_{P_4'})^2-(x_{P_1'}+x_{P_2'}-x_{P_3'}+x_{P_4'})^2-(x_{P_1'}+x_{P_2'}+x_{P_3'}-x_{P_4'})^2=0$.

  The remarkable fact about the Riemann relations is that since the divisor of $f_{\mathcal{Z}}$ only involves $P_1,P_2,P_3,P_4,Q_1,Q_2,Q_3,Q_4$, we can compute it without involving the P_i' (hence a field extension). We can reframe the previous invariance under the choice of cubical points as the computation $(x_{P_1})^2+(x_{P_2})^2+(x_{P_3})^2+(x_{P_4})^2-(x_T-x_{P_1})^2-(x_T-x_{P_2})^2-(x_T-x_{P_3})^2-(x_T-x_{P_4})^2=0$ where $x_T := \frac{1}{2}(x_{P_1}+x_{P_2}+x_{P_3}+x_{P_4})$.
- Consider $\mathcal{Z} = (3(P))-9((P))$ (this example comes from [Stao8, Equation 10.13]). We have $(3x_P)^2-9x_P^2=0$, and indeed $X(3\widetilde{P})/X(\widetilde{P})^9$ does not depend ont he choice of $\widetilde{P}$. However, $s(\mathcal{Z}) = 3P-9P = -6P \neq 0$, so $\mathcal{Z}$ is not associated to a function: we can check that $(x_R+3x_P)^2-9(x_R+x_P)^2 = -8x_R^2-12x_Px_R$, hence the value of $X(\widetilde{R}+3\widetilde{P})/X(\widetilde{R}+\widetilde{P})^9$ does depends on the choices of $\widetilde{R}$ and $\widetilde{R+P}$. In particular, replacing $\widetilde{R}$ by $\lambda_R\widetilde{R}$, and $\widetilde{R+P}$ by $\lambda_R\lambda'_{RP}\widetilde{R+P}$ changes this value by $\lambda_R^{-8}{\lambda'_{RP}}^{-6}$.

In summary, using a cycle of cycles $\mathcal{Z} = \sum n_i(\sum n_{i,j}P_{i,j})$ we can describe a cubical function $\widetilde{R} \mapsto X(t_{R,*}\mathcal{Z}) = \prod_i X(R+\widetilde{\sum n_{i,j}P_{i,j}})^{n_i}$; which in general depend not only on the choices of the $\widetilde{P_i}, \widetilde{P_i+P_j}$, but also on the choice of $\widetilde{R}, \widetilde{R+P_i}$.

When $\mathcal{Z}$ is of degree 0 and $s(\mathcal{Z}) = 0$, $\mathcal{Z}$ encodes a genuine function which descends on the abelian variety. But to compute it in practice as the evaluation of some coordinate X on the cycle $t_{R,*}\mathcal{Z}$ we need the stronger condition that $S(\mathcal{Z}) = 0$. In this case, the function only depends on the choices of $\widetilde{P_i}, \widetilde{P_i+P_j}$. This makes it very practical to build functions with prescribed divisors, such as the ones used in Theorem 4.28.

Finally, for certain well chosen cycles of cycles (e.g., coming from the algebraic Riemann relations), the function is canonical and does not depend on the choice of any cubical points. These canonical functions can then be used to compute the cubical arithmetic.

Example 4.31. Let $\phi : E_1 \to E_2$ be an isogeny of odd degree ℓ with kernel $K = \langle P\rangle$ between Montgomery Kummer lines. We assume furthermore that if $T = (0:1)$ on E_1, $T' = (0:1)$ on E_2, we have $\varphi(T) = T'$. The coordinate x' on E_2 has for divisor $2(T')-2(0_{E_2})$ hence ϕ^*x' has for divisor $\sum_{i=0}^{\ell-1}(2(T+iP)-2(iP))$. Since Z has for zero divisor $2(0_E)$, for a $R \in E_1$, making arbitrary choices for $\widetilde{R}, \widetilde{R+P}$, we consider the following product of cubical functions: $\Phi(R) = \prod_{i=0}^{\ell-1} Z(R+\widetilde{iP}+T)Z(\widetilde{R+iP})^{-1}$. If we take $\widetilde{T}$ such that the induced theta group

action is $\widetilde{T}\cdot(X,Z)=(Z,X)$, we obtain $\Phi(R)=\prod_{i=0}^{\ell-1}\frac{X(\widetilde{R+iP})}{Z(\widetilde{R+iP})}=\prod_{i=0}^{\ell-1}x(R+iP)$, which indeed has the correct divisor and evidently does not depend on any choice of cubical points.

4.7. **Cubical arithmetic and pairings on Kummer varieties.**

4.7.1. *Biextension additions on Kummer varieties.* Let $(A,\mathcal{L})$ be a principally polarised abelian variety, if $\mathcal{L}$ is indecomposable then $\mathcal{L}^2$ gives an embedding of the Kummer variety $A/\pm 1$.

Let $(X_1,\dots,X_m)\in\Gamma(\mathcal{L})$ be the global sections of $\mathcal{L}$ (we will call these level 2 coordinates). We then have by Section 4.5 the affine lift representation $\widetilde{P}=(X_1(\widetilde{P}),\dots,X_m(\widetilde{P}))$. The projective coordinates $(X_1(P):\dots:X_m(P))$ only allows to recover $\pm P$, so in this case our affine lift representation for $\widetilde{P}$ only recovers $\pm\widetilde{P}$.

However, when using the affine lift representation for the cubical points in the cubical biextension representation $g_{P,Q}=[\widetilde{P},\widetilde{Q};\tilde{0},\widetilde{P+Q}]$, then by [LR16] the points P,Q are determined up to the *same sign*. The reason is that, if Q is not of 2-torsion, the map $A\to A/\pm 1\times A/\pm 1, P\mapsto(\pm P,\pm(P+Q))$ is an embedding (and if both P,Q are of 2-torsion we can use the action of the theta group $G(\mathcal{L}^2)$ instead). Since pairings are bilinear, this representation of $g_{P,Q}$ is enough to recover the pairings $e(P,Q)$ exactly.

We just need to explain how to do a cubical biextension exponentiation using affine coordinates of level 2. The cubical ladder from Algorithm 4.2 works as is; we just need affine doublings and differential additions that lift the standard doublings and differential additions on the Kummer variety to the cubical arithmetic. However, for the cubical double and add algorithm from Algorithm 4.1, the algorithm requires a cubical biextension addition, which requires to lift an abelian variety standard addition.

We cannot do a standard addition on a Kummer variety. However, in the context of a cubical biextension addition, we are given representations $g_{P_1,Q}=[\widetilde{P_1},\widetilde{Q};\tilde{0},\widetilde{P_1+Q}]$ and $g_{P_2,Q}=[\widetilde{P_2},\widetilde{Q};\tilde{0},\widetilde{P_2+Q}]$ and we want to compute a representation $g_{P_1,Q}\star_1 g_{P_2,Q}=[\widetilde{P_1+P_2},\widetilde{Q};\tilde{0},\widetilde{P_1+P_2}+Q]$. We note that our affine lift representation, when interpreted as projective coordinates, give us the coordinate of $\pm P_1,\pm P_2,\pm Q,\pm(P_1+Q),\pm(P_2+Q)$ on the Kummer variety. This data is enough to compute a compatible addition $\pm(P_1+P_2)=\mathrm{CompatibleAdd}(\pm P_1,\pm P_2;\pm Q,\pm(P_1+Q),\pm(P_2+Q))$ as explained in [LR16]. Take any affine lift $\widetilde{P_1+P_2}$ of $\pm(P_1+P_2)$, and then proceed to compute $\widetilde{P_1+P_2}+Q$ via a three way addition like in the usual case. This gives Algorithm 4.7.

We obtain the following algorithm to compute the pairings associated to $\mathcal{L}^2$ on an abelian variety A. Let $P,Q\in A$, compute $P+Q\in A$, and then compute the projective coordinates $X_i(P),X_i(Q),X_i(P+Q)$ to get the points $\pm P,\pm Q,\pm(P+Q)$ on $A/\pm 1$. Take arbitrary lifts of these projective coordinates to get a representation of $g_{P,Q}=[\widetilde{P},\widetilde{Q};\tilde{0},\widetilde{P+Q}]$.

The pairings then requires to compute biextension exponentiations which can be done using Algorithms 4.2 and 4.7, and eventually the action of the theta group $G(\mathcal{L}^2)$ (to compute pairings with ℓ even) or the action of π_q (for the optimal Ate and optimal Ate pairings) which can be expressed naturally as in Section 4.5 on the level 2 affine coordinates X_i.

4.7.2. *Pairings on Kummer varieties.* The computation of $\pm(P+Q)$ (via its coordinates $X_i(\pm(P+Q))$) requires to start with $P,Q\in A$. If we want to compute pairings which genuinely lie on the Kummer variety, we then have only $\pm P,\pm Q$. The best we can compute from these points is the degree two étale subscheme $\pm(P\pm Q)$ of $A/\pm 1$. This subscheme is isomorphic to $\mathrm{Spec}\,R$, with $R=k[X]/\mathfrak{P}(X)$, $\mathfrak{P}$ a polynomial of degree 2. Typically, if X_1 is a separating coordinate, we can take $\mathfrak{P}(X)=(X-X_1(P+Q))(X-X_1(P-Q))$ and express the isomorphism between $\mathrm{Spec}\,R$ and $\pm(P\pm Q)$ by giving linear relations between

Input: a biextension element $g_{P,Q} = [\widetilde{P}, \widetilde{Q}, \tilde{0}, \widetilde{P+Q}]$ represented by cubical points

Output: cubical points $\widetilde{\ell P}, \widetilde{\ell P+Q}$ such that $g_{P,Q}^{\star_1,\ell} = [\widetilde{\ell P}, \widetilde{Q}, \tilde{0}, \widetilde{\ell P+Q}]$

→ For each bit b_i of ℓ from left to right (skipping the first one), given a cubical representation $\widetilde{nP}, \widetilde{nP+Q}$, of $g_{P,Q}^{\star_1,n}$, where n is the current truncation of ℓ on the leftmost bits, do a cubical biextension double:
   a. $\widetilde{2nP+Q} = \mathrm{DiffAdd}(\widetilde{nP+Q}, \widetilde{nP}, \widetilde{Q})$;
   b. $\widetilde{2nP} = \mathrm{Double}(\widetilde{nP})$;

   And if $b_i = 1$, also do a cubical biextension addition:
   a. Compute $(2n+1)P = \mathrm{CompatibleAdd}(2nP, P; 2nP+Q, P+Q)$ and take an arbitrary cubical lift $\widetilde{(2n+1)P}$ of $(2n+1)P$.
   b. $\widetilde{(2n+1)P+Q} = \mathrm{ThreeWayAdd}(\widetilde{2nP}, \widetilde{P}, \widetilde{Q}, \widetilde{P+Q}, \widetilde{2nP+Q}, \widetilde{(2n+1)P})$;

Algorithm 4.7. Biextension exponentiation on Kummer varieties via cubical double and add

$X_i(P \pm Q)$ and $X_1(P \pm Q)$. One can then consider the abelian and Kummer schemes A_R/R, $(A_R/\pm 1)/R$ over R. The coordinate $X \mod \mathfrak{P}$ encodes $P+Q$ and $P-Q$. More formally, we have a canonical R point $\mathrm{Spec}\, R \to A_R/\pm 1$, and as, eventually over a degree 2 extension k' of k, R splits as $R_{k'} = k' \oplus k'$, the isomorphism $\mathrm{Spec}\, R \simeq \pm(P \pm Q)$ splits as $\mathrm{Spec}\, k' \oplus \mathrm{Spec}\, k' \simeq \pm(P+Q) \times \pm(P-Q)$, so the canonical R point splits as the point $\pm(P+Q)$ and the point $\pm(P-Q)$. Anyway, working over R rather than over k, we can do our pairing computations as before, except that in the end we obtain a representation of the degree two scheme $e(P,Q)^{\pm 1}$ of $\mathbb{G}_m$. Typically this representation is given by a degree two polynomial $\mathfrak{Q}(X) = (X - e(P,Q))(X - e(P,Q)^{-1}) = X^2 - (e(P,Q) + e(P,Q)^{-1})X + 1$, and so we can recover the trace $(e(P,Q) + e(P,Q)^{-1})$, which we will call the symmetric pairing of P, Q.

Alternatively we can see $e(P,Q)^{\pm 1}$ as a point of $\mathbb{G}_m/\pm 1$, and if $x^{\pm 1} \in \mathbb{G}_m/\pm 1$, the trace $x^{\pm 1} \mapsto x + 1/x$ is a convenient representation of $x^{\pm 1}$. We can still do arithmetic on $\mathbb{G}_m/\pm 1$ (see [Rob21a, § 2.12.2]), notably compute exponentiation via squarings and differential multiplications. This allows us to compute reduced symmetric pairings from non reduced symmetric pairings. This strategy to compute pairings is well known, see [GL08] for Kummer lines and [LR15] for the case of Kummer varieties in the theta model.

4.8. **Analytic cubical points and analytic theta functions.** Our goal is now to find formulas for the cubical arithmetic using the affine lift representation. It will be convenient to work out formulas over $\mathbb{C}$.

Since we know that the cubical arithmetic is algebraic, working over the universal abelian scheme there certainly exists algebraic formulas over $\mathbb{C}$. By standard arguments (see [Rob21b, § 2.3.6]), if we find through analytic means algebraic formulas defiend over $\mathbb{Q}$, we know that they give the correct arithmetic formula over any field k of characteristic p, as long as the formulas have good reduction modulo p.

Now, from the algebraic Riemann relations on line bundles from Proposition 4.1, taking sections s_i of $\mathcal{L}$ we know that there should exists linear realations between suitable translated tensor products of the s_i, i.e. Riemann relations on sections. These have been worked out in the case where the s_i are the algebraic theta functions by Mumford in [Mum66, p. 333–335].

Now let $A/\mathbb{C}$ be an abelian variety, it is known that it is a (polarisable) torus $A = V/\Lambda$ with V a $\mathbb{C}$-vector space of dimension g, and that the analytic addition law on $V \simeq \mathbb{C}^g$

induces the algebraic addition law on A. Fix a very ample line bundle $\mathcal{L}$ on A. It is also known that there is a theory of analytic theta functions on A with respect to $\mathcal{L}$, which are given by analytic functions on V [Mum83; Mum84].

Given these theta functions θ_i, we have two representations of a point $P \in A$. First we have the projective representation: take any $z_P \in V$ above P, and let $\theta(P) = (\theta_1(z_P) : \ldots : \theta_m(z_P))$. The theta functions are automorphic with respect to Λ (with the same factor of automorphy), so changing z_P does not change the projective point $\theta(P)$. But we also have an affine representation $\theta(z_P) = (\theta_1(z_P), \ldots, \theta_m(z_P))$, which is an affine point above $\theta(P)$.

We remark the similarity with the discussion of Section 4.5. In fact, since V is simply connected, the pullback of any line bundle $\mathcal{L}$ on A to V is trivial, so essentially the choice of z_P above P induces a choice of rigidification of $\mathcal{L}$ at P.

If we know the projective points $\theta(P), \theta(Q)$, we can compute $\theta(P+Q)$. However, if we know $\theta(z_P), \theta(z_Q)$, we cannot recover the analytic addition $\theta(z_P + z_Q)$. However, we can use the analytic Riemann relations [Mum83], so that whenever we have $[z_1, z_2, z_3, z_4; z'_1, z'_2, z'_3, z'_4]$ in Riemann position, and we know all but one of the $\theta(z_i)$ we can recover exactly the last one. In particular, as in Example 4.3, we can do analytic differential additions and analytic three way additions.

Now the analytic Riemann relations on analytic theta functions are exactly the same as the algebraic Riemann relations on algebraic theta functions. Since the analytic Riemann relations are induced by the analytic group law on V on one hand, and the algebraic Riemann relations give the cubical structure, we deduce that the analytic addition law on V induces the cubical algebraic structure. See also [Bre83, p. 41, 42]. In fact Breen uses the algebraic cubical theory to give an alternative definition of algebraic theta functions compared to Mumford (which uses the Heisenberg group representation and a choice of isomorphism between the Heisenberg group and a theta group).

In summary, the cubical arithmetic encodes the algebraic information which can be extracted from the analytic group law of V above the algebraic group law of A. Analytically, a cubical point $\widetilde{P}$ corresponds to an analytic point z_P above P, and cubical arithmetic like $\widetilde{mP + nQ}$ (when we have enough information to compute it) corresponds to the analytic point $mz_P + nz_Q$. This allows us to naturally check the various compatibility relations on the cubical arithmetic by checking it through the analytic group law.

The first consequence, is that the affine version of the theta Riemann relations, since they are over $\mathbb{C}$ given by the analytic theta Riemann relations coming from the analytic group law, give explicit formulas for the cubical arithmetic expressed in terms of affine lifts of theta points. In particular, all the arithmetic on affine lift of theta points as developed in [LR16; LR10; LR15], was actually a cubical arithmetic in disguise, as should be clear when comparing Theorem 4.19 with [LR16; LR15]. This explain how to do the cubical arithmetic in theta models of abelian varieties. This allows to have a nice intuition on the operations allowed under the cubical arithmetic: any algebraic relations that can be derived from the transcendantal addition law $z_i + z_j + \cdots + z_k$ via the analytic theta functions (so for instance: we cannot recover the $\theta_i(z_1 + z_2)$ from the $\theta_i(z_1)$, $\theta_i(z_2)$, but we can is we also know the $\theta_i(z_1 - z_2)$) can be rewritten as some cubical operation.

Furthermore, the automorphic factor associated to theta functions gives an explicit trivialisation of the theta groups on $A = V/\Lambda$ pulled back to V, and of biextensions on $A \times A$ pulled back to $V \times V$. This allows to give an analytic proof of the link between cubical points and biextensions.

We conclude this section by an example of how the analytic interpretation of the cubical law makes it very easy to prove certain of its properties:

Lemma 4.32. *Let M be an 8×8 matrix of cubical points. Assume that all lines give cubical points in Riemann position, and 7 out of 8 columns also give cubical points in Riemann position. Then the last column also gives cubical poitns in Riemann position.*

Proof. Let $[\widetilde{P}_1, \widetilde{P}_2, \widetilde{P}_3, \widetilde{P}_4; \widetilde{Q}_1, \widetilde{Q}_2, \widetilde{Q}_3, \widetilde{Q}_4]$ be analytic cubical points in Riemann position. This means that we have an affine representation via analytic coordinates $\widetilde{P}_i = (\theta_j(z_i))$, $\widetilde{Q}_i = (\theta_j(z'_i))$, where $z_i, z'_i \in \mathbb{C}^g$ and $[z_1, z_2, z_3, z_4; z'1, z'_2, z'_3, z'_4]$ are analytic points in Riemann position.

To our matrix M of cubical points, corresponds a matrix M_z of elements $z_{ij} \in \mathbb{C}^g$, such that each line and 7 out of 8 columns are in Riemann position. Then by linear algebra the last column has to be in Riemann position too.

In other words: going from the $\widetilde{P}_i$ to the analytic z_i is a way to trivialize the cubical arithmetic; for an algebraic proof, we can use [Mor85, § I.5] instead. □

Corollary 4.33. *If $[\widetilde{P}_1, \widetilde{P}_2, \widetilde{P}_3, \widetilde{P}_4; \widetilde{Q}_1, \widetilde{Q}_2, \widetilde{Q}_3, \widetilde{Q}_4]$ are cubical points in Riemann position, so are $[m\widetilde{P}_1, m\widetilde{P}_2, m\widetilde{P}_3, m\widetilde{P}_4; m\widetilde{Q}_1, m\widetilde{Q}_2, m\widetilde{Q}_3, m\widetilde{Q}_4]$ for any $m \in \mathbb{Z}$.*

4.9. **Cubical arithmetic on elliptic curves and Kummer lines.** In this section, we give explicit formulas for the arithmetic of cubical elliptic points in Sections 4.9.1 to 4.9.3, and for cubical points on elliptic Kummer lines in Section 4.9.4. Finally in Section 4.9.5 we make the link between elliptic nets and cubical points.

4.9.1. *Weierstrass coordinates.* We will use Section 4.8 to derive the cubical arithmetic of an elliptic curve $(E, (0_E))$ in Weierstrass equation $y^2 = x^3 + a_2x^2 + a_4x + a_6$.

It is well known by Riemann Roch that the line bundle $\mathcal{L}$ associated to (0_E) has one global section Z_1. The line bundle $\mathcal{L}^2$ has two sections, X_2, Z_2 with $Z_2 = Z_1^2$. And finally the line bundle $\mathcal{L}^3$ has three sections X_3, Y_3, Z_3, with $X_3 = X_2Z_1, Z = Z_1^3$.

To simplify notations, we let $X, Y, Z = X_3, Y_3, Z_3$. We can work on the elliptic curve via the projective coordinates $(X : Y : Z)$, or via the affine coordinates $x = X/Z = X_2/Z_2, y = Y/Z$ which are defined everywhere except at 0_E. (It is also standard in elliptic curve cryptography to use Jacobian coordinates which are given by $(X_2 : Y_3 : Z_1)$, with projective weights $(2 : 3 : 1)$.)

Since 0_E is of multiplicity 1 in the divisor (0_E), by the proof of [Mum66, Proposition 2 p.307], we have that $Z_1(-\widetilde{P}) = -Z_1(\widetilde{P})$. So Z_1 is odd, and since x is odd and y even, X is even and Y is even.

The affine representation of a cubical point $\widetilde{P}$ above $P = (X(P) : Y(P) : Z(P))$, with respect to the divisor $D = 3(0_E)$, then corresponds to a choice of affine lift $\widetilde{P} = (X(\widetilde{P}), Y(\widetilde{P}), Z(\widetilde{P}))$ above the projective point P.

If we want to work with the cubical arithmetic associated to (0_E) (hence a principal polarisation), since it is not very ample, to represent a cubical point $\widetilde{P}$, we need to fix both a representation of P itself, e.g. via the Weierstrass coordinates $(X(P) : Y(P) : Z(P))$ and a choice of $Z_1(P)$, which we will denote by $Z_1(\widetilde{P})$. Our representation is thus $\widetilde{P} = (P, Z_1(\widetilde{P}))$. It is well defined except when $P = 0_E$ because the neutral point is a base point of (0_E) and $Z_1(0_E) = 0$.

Remark 4.34. As discussed in Remark 4.24, we want to normalize $\tilde{0}$ such that $(Z_1/(x/y))(\tilde{0}) = 1$. With this normalisation, we have $X_2(\tilde{0}) = (X_2/(Z_1/(x/y))^2)(\tilde{0}) = (X_2x^2/(Z_1^2y^2))(\tilde{0}) = (x^3/y^2)(\tilde{0}) = 1$, and $Y_3(\tilde{0}) = (Y_3/(Z_1/(x/y))^3)(\tilde{0}) = (Y_3x^3/(Z_1^3y^3))(\tilde{0}) = (x^3/y^2)(\tilde{0}) = 1$.

We will call sections of the divisor $n(0_E)$ coordinates of level n, so for instance Z_1 is of level 1, X_2, Z_2 of level 2, and X_3, Y_3, Z_3 of level 3. To specify a cubical point $\widetilde{P}$ with respect to the divisor $n(0_E)$, we need to specify P using some projective coordinates of level m: $(X_{m,i}(P))$, and one or several affine coordinates of level n: $X_{n,i}(\widetilde{P})$. If $m = n$, then the projective coordinates are subsumed by the affine coordinates, but we can take m different from n.

For instance the representation $\widetilde{P} = (X(\widetilde{P}), Y(\widetilde{P}), Z(\widetilde{P}))$ uses level 3 affine coordinates (so gives cubical points associated to $3(0_E)$); and our normalisation of the neutral point is $\tilde{0} = (0, 1, 0)$. The representation $\widetilde{P} = ((X(P) : Y(P) : Z(P)), Z_1(\widetilde{P}))$ uses level 3 projective coordinates with a level 1 affine coordinate (so cubical points associated to (0_E)). This times, $Z_1(0_E) = 0$ so to define our neutral point we need to add the extra condition that $(Z_1/(x/y))(\tilde{0}) = 1$.

In Section 4.9.4 we will use level 2 affine coordinates $\widetilde{P} = (X_2(\widetilde{P}), Z_2(\widetilde{P}))$. Projectively, these only allow to recover $(X_2(P) : Z_2(P))$, i.e. $x(P)$, so they cannot distinguish between P and $-P$. Our neutral point is $\tilde{0} = (1, 0)$.

On a twisted Edwards curve, the completed Edward coordinates are given by $\{(X : Z), (Y : T) \in \mathbb{P}^1 \times \mathbb{P}^1 \mid aX^2T^2 + Y^2Z^2 = Z^2T^2 + dX^2Y^2\}$, and the Segre embedding $\mathbb{P}^1 \times \mathbb{P}^1 \to \mathbb{P}^3$, $(X : Z), (Y : T) \mapsto (XT : YZ : ZT : XY)$ gives the extended Edwards coordinates (which are of level 4) [BBLP13, § 2.7]. Let M be the Montgomery model which is birationnally equivalent to E, and X_M, Z_M its level two coordinates, and $T = (1 : 0)$ the canonical point of 4 torsion on the Kummer line. Then up to a linear change of variable, the completed Edwards coordinates correspond precisely to the embedding $M \to \mathbb{P}^1 \times \mathbb{P}^1, P \mapsto ((X_M(P) : Z_M(P)), (X_M(P + T) : Z_M(P + T)))$. This was first remarked (implicitly) in [Koh11b; Koh11a, §8.1]; see also [FK22, § 3] and [LR16] for generalisations to higher dimension. This means we could use a mix of affine level 2 and projective level 4 coordinates to describe cubical points of level 2 in $\mathbb{A}^2 \times \mathbb{P}^1$ via $\widetilde{P} = ((X_M(\widetilde{P}), Z_M(\widetilde{P})), (X_M(P + T) : Z_M(P + T)))$, and compute pairings using a mix of Kummer line cubical arithmetic as in Section 5 and Edwards additions. We leave that for future work.

More generally, on an abelian variety A, with projective embedding $A \to \mathbb{P}^N$ given by a very ample divisor E, and a divisor D, we just need an explicit version of the theorem of the square for D, i.e., for $P, Q \in A$, represented by their projective coordinates $X_i(P), X_i(Q)$, $X_i \in \Gamma(E)$, build a function $g_{P,Q}$ with divisor $D_{P+Q} - D_P - D_Q$, such that we are able to evaluate it on $R = (X_i(R))$, to be able to work out the cubical arithmetic, hence biextension arithmetic, associated to D, while working in the model associated to E. It is not required to choose $E = D$.

4.9.2. *Analytic cubical elliptic points.* Analytically, the Weierstrass sigma function σ is a theta function (up to some exponential factor) which has zeroes of order 1 exactly at the lattice Λ, it can thus play the role of our Z_1 (see Remark 4.35). It is also well known that analytically, $x = \wp, y = \wp'$.

The cubical arithmetic with respect to σ is then given by the Frobenius Stickelberger relations (see [Stao8, Lemma 5.1.3; Sil86, Exercice 6.3; Bre83, Eq.(3.13.4) p 42]):

$$\frac{\sigma(z+w)\sigma(z-w)}{\sigma(z)^2\sigma(w)^2} = \rho(w)-\rho(z)$$
$$\frac{\sigma(2z)}{\sigma(z)^4} = -\rho'(z)$$
$$\frac{\sigma(x+y+z)\sigma(x)\sigma(y)\sigma(z)}{\sigma(x+y)\sigma(y+z)\sigma(x+z)} = \frac{-1}{2}\begin{vmatrix}1 & \wp(x) & \wp'(x)\\ 1 & \wp(y) & \wp'(y)\\ 1 & \wp(z) & \wp'(z)\end{vmatrix}\frac{1}{(\wp(x)-\wp(y))(\wp(y)-\wp(z))(\wp(z)-\wp(x))}.$$

These allows to compute the cubical doubling, cubical differential additions and cubical three way additions on a Weierstrass model. There also exists Frobenius–Stickelberger relations for multiway relations (which allows to compute them directly rather than through three way additions), but we won't need them.

Remark 4.35 (The normalised neutral point)**.** Our cubical analytic neutral point $\tilde{0}$ is the one corresponding to $0 \in \mathbb{C}$. The normalisation is given by $(\sigma/(\wp/\wp'))(0) = -2$ by [OKUO11, Lemma 1]. Indeed $(\wp\sigma^2)(0) = 1$ and $(\wp'\sigma^3)(0) = -2$. But the Weierstrass equation is $\wp'^2 = \wp^3 - g_2\wp - g_3$, so for a Weierstrass equation of the form $y^2 = x^3 + ax + b$, via $x = \wp, y = \wp'/2$, $\tilde{0}$ corresponds to the normalisation $(\sigma/(-x/y))(0) = 1$, which is the opposite of the normalisation we use in Remark 4.24 (alternatively: $\sigma = -Z_1$).

4.9.3. *Algebraic elliptic cubical points.* Our analytic cubical laws translate in the following algebraic law (taking into account the change of variable from Remark 4.35 between analytic Weierstrass coordinates and algebraic Weierstrass coordinates): given $\widetilde{P} = (x(P), y(P), Z_1(\widetilde{P}))$, $\widetilde{Q} = (x(Q), y(Q), Z_1(\widetilde{Q}))$ and $\widetilde{P-Q} = (x(P-Q), y(P-Q), Z_1(\widetilde{P-Q}))$, we have the cubical doubling

$$\widetilde{2P} = (x(2P), y(2P), Z_1(\widetilde{P})^4 2y(P)),$$

and the cubical differential addition is $\widetilde{P+Q} = (x(P+Q), y(P+Q), Z_1(\widetilde{P+Q}))$ with

$$Z(\widetilde{P+Q})Z(\widetilde{P-Q}) = Z_1(\widetilde{P})^2 Z_1(\widetilde{Q})^2 (x(Q)-x(P)).$$

And the three way addition reads:

$$\frac{Z_1(\widetilde{P+Q+R})Z_1(\widetilde{P})Z_1(\widetilde{Q})Z_1(\widetilde{R})}{Z_1(\widetilde{P+Q})Z_1(\widetilde{Q+R})Z_1(\widetilde{P+R})} = \begin{vmatrix}1 & x(P) & y(P)\\ 1 & x(Q) & y(Q)\\ 1 & x(R) & y(R)\end{vmatrix}\frac{1}{(x(Q)-x(P))(x(R)-x(P))(x(R)-x(Q))}$$
$$= \frac{l_{P,Q}(R)}{(x(R)-x(P))(x(R)-x(Q))},$$

where $l_{P,Q} : y - \alpha x - \beta$ is the equation of the line going through P and Q.

Finally, for general cubical points in Riemann position $[\widetilde{P}_1, \widetilde{P}_2, \widetilde{P}_3, \widetilde{P}_4; \widetilde{Q}_1, \widetilde{Q}_2, \widetilde{Q}_3, \widetilde{Q}_4]$, if we let $P_1 = P'_1 + P'_2$, $P_2 = P'_1 - P'_2$, $P_3 = P'_3 + P'_4$, $P_4 = P'_3 - P'_4$, then $2P'_1 = P_1 + P_2$, $2P'_2 = P_1 - P_2$, $2P'_3 = P_3 + P_4$, $2P'_4 = P_3 - P_4$, and we have: $Q_1 = P'_3 - P'_2$, $Q_2 = P'_3 + P'_2$, $Q_3 = P'_1 - P'_4$, $Q_4 = P'_1 + P'_4$. We can then use the cubical differential additions above to obtain

$$\frac{Z_1(\widetilde{P_1})Z_1(\widetilde{P_2})Z_1(\widetilde{P_3})Z_1(\widetilde{P_4})}{Z_1(\widetilde{Q_1})Z_1(\widetilde{Q_2})Z_1(\widetilde{Q_3})Z_1(\widetilde{Q_4})} = \frac{(x(P'_2)-x(P'_1))(x(P'_4)-x(P'_3))}{(x(P'_2)-x(P'_3))(x(P'_4)-x(P'_1))}.$$

For the reader who do not like analytic arguments, let us give a purely algebraic proof of the cubical arithmetic for elliptic curves in Weierstrass form. Recall from Remark 4.24 that for an elliptic curve in Weierstrass form $y^2 = x^3 + a_2x^2 + a_4x + a_6$, we take $\tilde{0}$ corresponding to the uniformiser $\pi_{0_E} = x/y$, i.e. $(Z_1/(x/y))(\tilde{0}) = 1$

By Example 4.3 and Section 4.5.2, for a differential addition, we have $\frac{Z_1(P\widetilde{+Q}+R)Z_1(P\widetilde{-Q}+R)Z_1(\widetilde{R})Z_1(\tilde{(}R)}{Z_1(-\widetilde{Q}+R)Z_1(\widetilde{Q+R})Z_1(\widetilde{P+R})Z_1(\widetilde{P+R})} = \gamma(R)$. Here by Example 4.3, $\gamma = \frac{g_{Q,-Q}(\cdot)}{g_{Q,-Q}(P-(\cdot))}$.

For a generic Q, we can take $g_{Q,-Q} = x - x(Q)$ and so $\gamma(R) = \frac{x(P-R)-x(Q)}{x(R)-x(Q)}$. We also have $Z_1(-Q) = -Z_1(Q)$ and even $Z_1(-\widetilde{Q}) = -Z_1(\widetilde{Q})$.

Since our choice of base point evaluation to represent cubical functions is $\tilde{0}$, we have $Z_1(\widetilde{P+Q})Z_1(\widetilde{P-Q}) = Z_1(\widetilde{P})^2Z_1(\widetilde{Q})Z_1(-\widetilde{Q})Z_1(\tilde{0})^{-2}\gamma(0_E)$.

And since $Z_1/(x/y)(\tilde{0}) = 1$, we have $(\gamma/Z_1^2)(\tilde{0}) = (\gamma y^2/x^2)(0_E)$. Hence, we have $Z_1(\widetilde{P+Q})Z_1(\widetilde{P-Q}) = Z_1(\widetilde{P})^2Z_1(\widetilde{Q})^2 \cdot -(\gamma y^2/x^2)(0_E)$.

The numerator of γ evaluated at $R = 0_E$ gives $x(P) - x(Q)$. The denominator of γ multiplied by $(x/y)^2$ is $x(R)^3/y(R)^2 - x(R)^2x(Q)/y(R)^2$. The term $x(R)^3/y(R)^2$ is equal to 1 at $R = 0_E$ because of the curve equation, while the second term gives 0. So $(\gamma y^2/x^2)(0_E) = x(P) - x(Q)$. In the end, we have $Z_1(\widetilde{P+Q})Z_1(\widetilde{P-Q}) = (x(Q) - x(P))Z_1(\widetilde{P})^2Z_1(\widetilde{Q})^2$, as expected.

We let the reader work out the case of doublings and three way additions.

4.9.4. *Cubical arithmetic on Kummer lines.* We could use $Z_2 = Z_1^2$ to compute the cubical arithmetic with respect to the divisor $D = 2(0_E)$: we would represent $\widetilde{P}$ by $(x(P), y(P), Z_1^2(P))$. This combines level 3 projective coordinates with one level 2 affine coordinates.

Instead, we will use level 2 affine coordinates only, at the cost of moving to the Kummer line. The projective coordinates with respect to $2(0_E)$ are $(X_2 : Z_2)$, and so the associated affine representation from Section 4.5 is $\widetilde{P} = (X_2(\widetilde{P}), Z_2(\widetilde{P})) = (x(P)Z_2(\widetilde{P}), Z_2(\widetilde{P}))$. It is enough to recover P up to a sign, hence we can interpret this latter representation as a cubical representation over the Kummer line.

The cubical arithmetic in this affine representation then becomes, for cubical doublings, $\widetilde{2P} = (x(2P)Z_2(\widetilde{2P}), Z_2(\widetilde{2P}))$ with $Z_2(\widetilde{2P}) = 4y(P)^2Z(\tilde{P})^4$, and we can use the Weierstrass equation to express $y(P)^2$ in terms of $x(P)$. And the cubical differential addition is $\widetilde{P+Q} = (x(P+Q)Z_2(\widetilde{P+Q}), Z_2(\widetilde{P+Q}))$ with $Z_2(\widetilde{P+Q})Z_2(\widetilde{P-Q}) = Z_2(\widetilde{P})^2Z_2(\widetilde{Q})^2(x(Q) - x(P))^2$.

The three way addition is more complex. First from $x(P_1)$, $x(P_2)$, $x(P_1 + Q)$, $x(P_2 + Q)$ we can use a compatible addition to recover $x(P_1 - P_2)$ or $x(P_1 + P_2 + Q)$, and then use a projective differential addition to recover $x(P_1 + P_2)$. Now we can combine the level 1 cubical differential additions formulas for $Z_1(P_1 + \widetilde{P_2} + Q)Z_1(P_1 - \widetilde{P_2} + Q)$, $Z_1(P_1 + \widetilde{P_2} + Q)Z_1(P_2 - \widetilde{P_1} + Q)$, $Z_1(P_1 - \widetilde{P_2} + Q)Z_1(P_2 - \widetilde{P_1} + Q)$, and $Z_1(P_1 \widetilde{+} P_2)Z_1(P_1 \widetilde{-} P_2)$, to obtain:

$$\begin{aligned} Z_2(P_1 + \widetilde{P_2} + Q) &= \frac{Z_2(P_1\widetilde{+Q})Z_2(P_2\widetilde{+Q})Z_2(\widetilde{P_1})Z_2(\widetilde{P_2})}{Z_2(P_1\widetilde{-}P_2)Z_2(\widetilde{Q})} \frac{(x(P_2) - x(P_1 + Q))(x(P_1) - x(P_2 + Q))}{(x(P_1 - P_2) - x(Q))} \\ &= \frac{(X_2(\widetilde{P_2})Z_2(P_1\widetilde{+Q}) - X_2(P_1\widetilde{+Q})Z_2(\widetilde{P_2}))(X_2(\widetilde{P_1})Z_2(P_2\widetilde{+Q}) - X_2(P_2\widetilde{+Q})Z_2(\widetilde{P_1}))}{(X_2(P_1\widetilde{-}P_2)Z_2(\widetilde{Q}) - Z_2(P_1\widetilde{-}P_2)X_2(\widetilde{Q}))}. \end{aligned}$$

Since compatible additions give $x(P_1 + P_2 + Q)$, knowing $Z_2(P_1 + \widetilde{P_2} + Q)$ is enough to recover $X_2(P_1 + \widetilde{P_2} + Q)$, hence $P_1 + \widetilde{P_2} + Q$. (We will see a more direct formula

for $X_2(P_1 + \widetilde{P_2} + Q)$ in the Montgomery model in Section 5.2). Likewise, we can compute $P_1 \widetilde{+} P_2 = \mathrm{DiffAdd}(\widetilde{P_1}, \widetilde{P_2}, P_1 \widetilde{-} P_2)$, then the biextension element corresponding to $[P_1 \widetilde{+} P_2, \widetilde{Q}; \tilde{0}, P_1 + \widetilde{P_2} + Q$ is equal to $g_{P_1,Q} \star_1 g_{P_2,Q}$.

For the compatible additions, let $\kappa_{00} = X(P_1+P_2)X(P_1-P_2), \kappa_{01} = X(P_1+P_2)Z(P_1-P_2)+X(P_1-P_2)Z(P_1+P_2), \kappa_{11} = Z(P_1+P_2)Z(P_1-P_2)$, so that $(X(P_1+P_2) : Z(P_1+P_2)), (X(P_1-P_2) : Z(P_1-P_2))$ are solutions of the homogeneous system $P(X,Z) = \kappa_{11}X^2 - \kappa_{01}XZ + \kappa_{00}Z^2 = 0$. By symmetry, we can always write the κ as biquadratic polynomials in $(X(P_1) : Z(P_1)), (X(P_2) : Z(P_2))$. Define similarly κ' for $P'_1 = P_1 + Q$, $P'_2 = P_2 + Q$ to obtain a polynomial $P'(X,Z)$. The point $(X(P_1-P_2) : Z(P_1-P_2))$ is a root of both $P(X,Z)$ and $P'(X,Z)$, and can be written as $(\kappa'_{01}\kappa_{00} - \kappa_{01}\kappa'_{00} : \kappa'_{11}\kappa_{00} - \kappa_{11}\kappa'_{00})$.

We can use the $\mathbb{G}_m$-action from Lemma 4.15 to get rid of divisions: the cubical points $[\lambda \cdot P_1 \widetilde{+} P_2, \widetilde{Q}; \tilde{0}, \lambda \cdot P_1 + \widetilde{P_2} + Q]$ and $[P_1 \widetilde{+} P_2, \widetilde{Q}; \tilde{0}, P_1 + \widetilde{P_2} + Q]$ represent the same biextension elements. This allows to replace divisions by multiplication.

We recall that a biextension point is then represented via $g_{P,Q} = [\widetilde{P}, \widetilde{Q}; \tilde{0}, \widetilde{P+Q}]$. We could imagine a mix of the $Z_2 = Z_1^2$ representation and the affine lift representation: when computing $g_{P,Q}^{\star_1,\ell} = [\ell\widetilde{P}, \widetilde{Q}; \tilde{0}, \ell\widetilde{P+Q}]$, compute $(X_2(\ell\widetilde{P}), Z_2(\ell\widetilde{P}))$ via a cubical biextension ladder to represent $\ell\widetilde{P}$, and use $Z_2(\ell\widetilde{P+Q}) = Z_2(\ell\widetilde{P+Q})$ to represent $\ell\widetilde{P+Q}$. Indeed, to recover $X_2(\ell\widetilde{P+Q})$, given $Z_2(\ell\widetilde{P+Q})$ (and provided it is not zero), we just need to compute $x_2(\ell P + Q)$. Since we are doing a ladder, we have $(\ell-1)P$ and ℓP, and we can recover $\ell P + Q$ via a compatible addition $\mathrm{CompatibleAdd}(\ell P, Q; (\ell-1)P, P+Q)$.

So $Z_2(\ell\widetilde{P+Q})$ is enough to compute the cubical arithmetic. Can we find fast formulas using Z_2 only without recovering X_2? This is related to the question of denominator elimination in Section 6.1.

4.9.5. *Elliptic nets.* Elliptic nets are another way to compute the cubical arithmetic of an elliptic curve (or abelian variety), associated to the principal polarisation (0_E). In elliptic nets, the cubical point $\widetilde{P} = (P, \sigma(\widetilde{P}))$ is only represented through $\sigma(\widetilde{P})$. This is not enough to determine $\widetilde{P}$ let alone do cubical arithmetic.

The key insight of Stange is the following: while $\sigma(\widetilde{P})$ alone is not enough, the data of $\sigma(m\widetilde{P})$ for small values of m is enough to recover any $\sigma(n\widetilde{P})$ through a recurrence formula. Likewise, the values of $\sigma(n\widetilde{P+mQ})$ for small values of n, m allow to compute all values $\sigma(\ell_1\widetilde{P+\ell_2}Q)$ through a more complicated recurrence formula.

The resulting recurrence formulas allows to build iteratively the so called elliptic nets. Since elliptic nets compute the cubical arithmetic, they can be used to compute pairings: compare [Sta08, Theorems 17.2.1, 17.2.2] with Theorem 4.19 and [OKUO11, Theorems 4, 5] with Proposition 4.22. A somewhat annoying thing with the elliptic net representation is that it cannot represent $\widetilde{0_E}$ since $\sigma(0_E) = 0$, so when using it to compute pairings through the bixtension monodromy, it requires to compare $g_{P,Q}^{\ell+1}$ with $g_{P,Q}$ rather than $g_{P,Q}^{\ell}$ with the constant 1.

Elliptic nets extend to abelian varieties using analytic thetas instead of the Weierstrass σ function, see [Tra14, Chapitre 3] (which also gives formula for analytic thetas of higher level than $n = 2$). Although the result is stated there for Jacobians of hyperelliptic curves, the proof is valid for any abelian variety. The idea is that the differential addition formula deduced from Riemann relations give a bilinear relation between products of the form $\theta_i(z_1+z_2)\theta_j(z_1-z_2)$ and products of the form $\theta_i(z_1)\theta_i(z_2)$ and $\theta_i(z_2)\theta_j(z_2)$. The associated matrix to these bilinear relations is not of full rank and taking its determinant give the recurrence relation sought for.

We now compare elliptic nets with our approach. First the strategy of the proof is slightly different: Stange uses the analytic σ function to derive algebraic formulas for elliptic nets over $\mathbb{C}$, which she uses to give algebraic formulas for elliptic nets over any field. Here we use the fact that we already know that the cubical arithmetic is algebraic to give an algebraic interpretation of the analytic function σ as defining a cubical point, so that we can define an algebraic σ over any field. We could then recover the algebraic elliptic net recurrence from this algebraic σ.

But the main difference is in terms of our choice of representation of cubical point: representing them only by $\sigma(\widetilde{P})$ loses a lot of information and the elliptic net representation needs to work with many values. In Section 4.9, we give cubical formulas for the representation $(P, \sigma(\widetilde{P}))$ where we keep track of the underlying point on the elliptic curve together with $\sigma(\widetilde{P})$. In other words: the cubical points are the intrinsic object, and since $\sigma(\widetilde{P})$ alone can not recover $\widetilde{P}$, it makes working with the elliptic nets $\sigma(\sum n_i\widetilde{P}_i)$ directly harder than working with the cubical points $\sum n_i\widetilde{P}_i$ themselves.

And we will see in Section 5 that the sweet spot for efficient cubical arithmetic seems to be to use the affine representation associated to the divisor $2(0_E)$ and the global sections X, Z, i.e. represent $\widetilde{P}$ via the level two affine coordinates $X(\widetilde{P}), Z(\widetilde{P})$ rather than by only one affine coordinate $\sigma(\widetilde{P})$. Via this representation, as explained in Section 4.5 we will be able to leverage the efficient Kummer arithmetic to an efficient cubical arithmetic.

The only drawback is that the polarisation associated to $2(0_E)$ is not principal anymore, so as explained in Remark 2.11 we will need to use the full power of Theorem 2.9 rather than Corollary 2.5 to correctly handle the case ℓ even.

5. Pairings on Kummer lines

We specialize in Section 5.1 our whole framework of cubical arithmetic from Section 4 to the case of Kummer lines. Then we specialize further in Section 5.2 to the case of the Montgomery model of Kummer lines, which allows us to prove Theorem 1.1.

5.1. **Algorithms for a Kummer line.** We will represent our cubical points through the affine lift representation associated to the sections $X, Z \in \Gamma(2(0_E))$ of the divisor $D = 2(0_E)$. We will always take $\tilde{0} = (1, 0)$ as our choice of affine lift of $0_E = (1 : 0)$. Indeed, we have seen in Remark 4.34 that normalizing $\tilde{0}$ in level 2 through $(Z_2/(x/y)^2)(\tilde{0}) = 1$, makes $X_2(\tilde{0}) = 1$, so $\tilde{0} = (1, 0)$ in level 2 corresponds to $Z_2(\tilde{0}) = 1$ being normalised with respect to $(x/y)^2$.

The algorithm to compute the non reduced Tate pairing is given in Algorithm 5.1. To compute the Weil pairing, it suffices to apply Algorithm 5.1 again with the arguments P, Q reversed, but making the *same choices* for $\widetilde{P}, \widetilde{Q}, \widetilde{P+Q}$ (or if making different choices for whatever reason, ensuring the associated biextension element $g_{Q,P}$ is equal to $\iota(g_{P,Q})$), to obtain λ_Q. The Weil pairing is $e_{W,2(0_E),\ell}(P, Q) = \lambda_P/\lambda_Q$.

This is the (non reduced) Tate and Weil pairings associated to $2(0_E)$, so the square of the usual non reduced Tate and Weil pairing. When ℓ is odd, and if the characteristic $p > 0$, to recover the usual reduced Tate pairing it suffices to adjust the final exponentiation from $(q-1)/\ell$ to $(q-1)/(2\ell)$. In any case, having the squares of the usual pairings on μ_ℓ is not a problem in practice if ℓ is odd (one could always take a square root if needed). This becomes a problem if ℓ is even, in which case we use the strategy of Remark 2.11 to compute the standard (non reduced) Tate and Weil pairing associated to (0_E). In this case, $\ell = 2m$, and rather than computing $\widetilde{\ell P}, \widetilde{\ell P+Q}$, we compute $\widetilde{mP}, \widetilde{mP+Q}$.

By assumption, $T = mP$ is a point of two torsion, and we can now use the action of an element of the theta group $g_T \in G(2(0_E))$ above T as explained in Section 4.5.

Input: $(X(P) : Z(P))$ a rational point of ℓ-torsion, $(X(Q) : Z(Q))$ a rational point, $(X(P+Q) : Z(P+Q))$
Output: The non reduced Tate pairing $e_{T,2(0_E),\ell}(P,Q)$

.

- ➔ Take arbitrary affine lifts $\widetilde{P} = (X(\widetilde{P}), Z(\widetilde{P}))$, $\widetilde{Q} = (X(\widetilde{Q}), Z(\widetilde{Q}))$, $\widetilde{P+Q} = (X(\widetilde{P+Q}), Z(\widetilde{P+Q}))$.
- ➔ Compute $\widehat{\ell\widetilde{P}}, \widehat{\ell\widetilde{P+Q}}$ using either Algorithms 4.2 and 4.7.
- ➔ Write $\widehat{\ell\widetilde{P}} = \lambda_{0,P}\tilde{0}$, $\widehat{\ell\widetilde{P+Q}} = \lambda_{1,P}\widetilde{Q}$.
- ➔ Return $\lambda_P = \lambda_{1,P}/\lambda_{0,P}$.

ALGORITHM 5.1. The non reduced Tate pairing associated to $2(0_E)$ on a Kummer line

Concretely, the translation by T is given by a projective matrix, i.e. an action of the form $(X,Z) \mapsto (aX+bZ : cX+dZ)$. Take any choice of affine lift $(X,Z) \mapsto (aX+bZ, cX+dZ)$ of the translation by T on (X,Z) coordinates. In other words, we lift the projective 2×2 matrix of translation to a standard matrix; this is the same as making a choice of theta group element $g_T \in G(2(0_E))$ above T and looking at the action of g_T on (X,Z). The choice of an affine point $\widetilde{T}$ is a way to encode this choice: g_T should then be the unique element such that $g_T \cdot \tilde{0} = \widetilde{T}$. We call the action of a g_T corresponding to $\widetilde{T}$ on a cubical point $\widetilde{P}$ the translation of $\widetilde{P}$ by $\widetilde{T}$ and denote it $\widetilde{P} + \widetilde{T} = \text{Translate}(\widetilde{P}, \widetilde{T})$. Then we can compute $\widehat{\ell\widetilde{P}} = \widehat{m\widetilde{P}} + \widetilde{T}$, $\widehat{\ell\widetilde{P+Q}} = \widehat{m\widetilde{P+Q}} + \widetilde{T}$.

Since $T = mP$, we could use in practice the choice of $\widetilde{T} = \widehat{m\widetilde{P}}$ which just computed; but we remark that a different choice of $\widetilde{T}$ would still get the correct value as long as we use the same choice of $\widetilde{T}$ for both the translation action to compute $\widehat{\ell\widetilde{P}}$ and $\widehat{\ell\widetilde{P+Q}}$, since changing $\widetilde{T}$ to $\lambda\widetilde{T}$ would change $\widehat{\ell\widetilde{P}}$ to $\lambda\widehat{\ell\widetilde{P}}$ and $\widehat{\ell\widetilde{P+Q}}$ to $\lambda\widehat{\ell\widetilde{P+Q}}$, so they would still induce the same biextension element by Lemma 4.15.

We will also denote by $\text{Translate}(\widetilde{P}, T)$ an algorithm which makes a choice of $\widetilde{T}$ depending on T and then call $\text{Translate}(\widetilde{P}, \widetilde{T})$; as long as the choice of $\widetilde{T}$ is the same for every call with the same T, the computation will be correct. This can save us some arithmetic operations, for instance on a Montgomery model if $\widehat{m\widetilde{P}} = (0,a)$, we prefer to take $\widetilde{(0:1)} = (0,1)$ rather than $(0,a)$. Indeed in the former case, the translation action by $(0,1)$ is simply $(X,Z) \mapsto (Z,X)$, while the translation action of $(0,a)$ would be $(X,Z) \mapsto (aZ,aX)$, so we save multiplications.

The algorithm for the non reduced Tate pairing for even ℓ is given in Algorithm 5.2, and like for Algorithm 5.1 it gives an algorithm for the Weil pairing by calling it a second time with (P,Q) swapped and computing $e_{W,(0_E),\ell}(P,Q) = \lambda_P/\lambda_Q$.

By Remark 4.27, it is important in Algorithm 5.1 that we normalize our cubical points via $Z=1$. Indeed, the divisor of zeroes of Z is $2(0_E)$, so it satisfies the condition of Algorithm 5.1, and normalising all points with $Z=1$ is a convenient way to be sure that the associated biextension element $g_{P,Q}$ is a rational tensor square. (We do have $Z(\tilde{0}) = 0$, but $\tilde{0}$ is normalised for Z with respect to $(x/y)^2$ which is a square too.)

We could allow an arbitrary normalisation of $\widetilde{P}, \widetilde{Q}, \widetilde{P+Q}$, but we would then need to adjust λ_P by $\left(\frac{Z(\widetilde{P+Q})}{Z(\widetilde{P})Z(\widetilde{Q})}\right)^{\ell}$.

Input: $(X(P) : Z(P))$ a rational point of ℓ-torsion, $(X(Q) : Z(Q))$ a rational point, $(X(P+Q) : Z(P+Q))$
Output: The non reduced Tate pairing $e_{T,(0_E),\ell}(P,Q)$

.

- Take affine lifts $\widetilde{P} = (X(\widetilde{P}), Z(\widetilde{P}) = 1)$, $\widetilde{Q} = (X(\widetilde{Q}), Z(\widetilde{Q}) = 1)$, $\widetilde{P+Q} = (X(\widetilde{P+Q}), Z(\widetilde{P+Q}) = 1)$.
- Compute $\widetilde{mP}, \widetilde{mP+Q}$ using either Algorithms 4.2 and 4.7.
- Let $T = mP$, and compute $\widetilde{\ell P} = \text{Translate}(\widetilde{mP}, T)$ and $\widetilde{\ell P+Q} = \text{Translate}(\widetilde{mP+Q}, T)$.
- Write $\widetilde{\ell P} = \lambda_{0,P}\widetilde{0}$, $\widetilde{\ell P+Q} = \lambda_{1,P}\widetilde{Q}$.
- Return $\lambda_P = \lambda_{1,P}/\lambda_{0,P}$.

ALGORITHM 5.2. The non reduced Tate pairing associated to (0_E) on a Kummer line for $\ell = 2m$ even

Input: $(X(P) : Z(P)) \in \mathbb{G}_2$, $(X(Q) : Z(Q)) \in \mathbb{G}_1$, $(X(P+Q) : Z(P+Q)), \lambda \equiv q \mod \ell$
Output: The non reduced Ate pairing $a_{\lambda,2(0_E),\ell}(P,Q)$

.

- Take arbitrary affine lifts $\widetilde{P} = (X(\widetilde{P}), Z(\widetilde{P}))$, $\widetilde{Q} = (X(\widetilde{Q}), Z(\widetilde{Q}))$, $\widetilde{P+Q} = (X(\widetilde{P+Q}), Z(\widetilde{P+Q}))$.
- Compute $\widetilde{mP}, \widetilde{mP+Q}$ using either Algorithms 4.2 and 4.7.
- Write $\widetilde{\lambda P} = \lambda_{0,P}\pi_q(\widetilde{P})$, $\widetilde{\lambda P+Q} = \lambda_{1,P}\pi_q(\widetilde{P+Q})$.
- Return $\lambda_P = \lambda_{1,P}/\lambda_{0,P}$.

ALGORITHM 5.3. The non reduced Ate pairing associated to $2(0_E)$ on a Kummer line

We can also use Proposition 4.22 to compute the (non reduced) Ate pairing (with respect to $2(0_E)$, so the square of the usual Ate pairing). We reuse the notations of Section 3.4 and Proposition 4.22, and let $m \equiv \ell \mod q$. We recall that $\pi_q((X(\widetilde{P}), Z(\widetilde{P}))) = (\pi_q(X(\widetilde{P})), \pi_q(Z(\widetilde{P})))$.

A summary is given by:

Theorem 5.1. *Let $E/\mathbb{F}_q$ be an elliptic curve, and $X = X_{2(0_E)}$ the biextension associated to the divisor $2(0_E)$.*

When ℓ is odd, to compute the square of the non reduced Tate pairing (resp. of the Weil pairing), we need to compute one (resp. two) biextension exponentiation by ℓ.

When ℓ is even, to compute the usual non reduced Tate pairing (resp. of the Weil pairing), we need to compute one (resp. two) biextension exponentiation by $\ell/2$, followed by an affine translation by a point of two torsion.

To compute the square of the usual m-Ate pairing (with $m \equiv q \mod \ell$), we need to compute one biextension exponentiation by m.

Furthermore, using the cubical representation of biextension elements, and representing cubical points $\widetilde{P}$ by the affine lifts $\widetilde{P} = (X(\widetilde{P}), Z(\widetilde{P}))$, where X, Z are sections of $2(0_E)$, a biextension exponetiation then costs:

- *Using the cubical biextension ladder from Algorithm 4.2, one affine doubling and two affine differential additions by bits. (One could replace an affine differential addition by a three way addition instead.)*
  As special cases:
  - *When $\ell = 2^m$ or for self pairing $e(P,P)$, one affine doubling and one affine differential addition by bits*
  - *For batch Tate or Ate pairings computations $e(P,Q_i)$ with the same base point P, after the first biextension exponentiation the follow up biextension exponentiation only cost one differential addition (alternatively, one three way addition) by bits*
- *Using the cubical biextension double and add from Algorithm 4.1, one affine doubling and one differential addition (or alternatively one three way addition) for each doubling, and one compatible addition and one three way addition for each addition.*

5.2. **The Montgomery model.** As explained in Section 5.1, to compute pairings using the formulas from Theorem 4.19 and Proposition 4.22, we need to compute cubical biextension exponentiations using either Algorithm 4.2 or Algorithm 4.7.

5.2.1. *Cubical ladder on the Montgomery model.* For the first one, we just need to explain how to compute affine doublings and affine differential additions in X, Z coordinates. For the second one, we also need to give algorithms for compatible additions and three way additions in X, Z coordinates.

In this section, we thus concentrate on the Montgomery model of Kummer lines. We use Section 4.9 to derive our cubical arithmetic. We start with a Montgomery curve $By^2 = x^3 + \mathcal{A}x^2 + 1$. x-only additions have a particularly nice form on the Montgomery model:

$$x(2P) = \frac{(x(P)^2-1)^2}{4x(P)(x(P)^2+\mathcal{A}x(P)+1)} = \frac{(X(P)^2-Z(P)^2)^2}{4X(P)Z(P)(X(P)^2+\mathcal{A}X(P)Z(P)+Z(P)^2)},$$

and

$$x(P+Q)x(P-Q) = \left(\frac{x(P)x(Q)-1}{x(P)-x(Q)}\right)^2 = \left(\frac{X(P)X(Q)-Z(P)Z(Q)}{X(P)Z(Q)-Z(P)X(Q)}\right)^2.$$

On the other hand, by Section 4.9.4, we have

$$Z(2\widetilde{P}) = 4X(\widetilde{P})Z(\widetilde{P})(X(\widetilde{P})^2+\mathcal{A}X(\widetilde{P})Z(\widetilde{P})+Z(\widetilde{P})^2),$$

and

$$Z(\widetilde{P+Q})Z(\widetilde{P-Q}) = \left(X(\widetilde{Q})Z(\widetilde{P})-X(\widetilde{P})Z(\widetilde{Q})\right)^2.$$

In particular, combining both equations, we see that the natural way to write $x(2P)$, $x(P+Q)x(P-Q)$ as rational functions in terms of the projective coordinates $X(P), Z(P), X(Q), Z(Q)$ already gives us the correct cubical arithmetic by taking the numerator for $X(2\widetilde{P})$, $X(\widetilde{P+Q})X(\widetilde{P-Q})$ and the denominator for $Z(2\widetilde{P})$, $Z(\widetilde{P+Q})Z(\widetilde{P-Q})$ respectively! We remark that we can also directly recover the formulas for $X(\widetilde{2P})$ and $X(\widetilde{P+Q})X(\widetilde{P-Q})$ from the ones for Z, by using that if $\widetilde{T} = (1,0)$, $X(\widetilde{P}) = Z(\widetilde{P+T})$. We obtain Algorithms 5.4 and 5.5 for the cubical arithmetic.

Some comments are in order: in these algorithms we take $\tilde{0} = (1,0)$. Taking a different affine lift of $\tilde{0}$, we would need to adjust the doubling and differential addition formulas accordingly. Also the Montgomery coefficients $\mathcal{A}$ or $\frac{\mathcal{A}+2}{4}$ are often represented in projective

Input: $\widetilde{P} = (X(P), Z(P))$
Output: $\widetilde{2P} = (X(2P), Z(2P)) = \mathrm{Double}(\widetilde{P})$

.

- $a = (X(P) + Z(P))^2$
- $b = (X(P) - Z(P))^2$
- $c = a - b$
- $X(2P) = ab$
- $Z(2P) = c(b + \frac{A+2}{4}c)$

ALGORITHM 5.4. Affine cubical doubling in the Montgomery model

Input: $\widetilde{P} = (X(P), Z(P)), \widetilde{Q} = (X(Q), Z(Q)), \widetilde{P-Q} = (X(P-Q), Z(P-Q))$
Output: $\widetilde{P+Q} = (X(P+Q), Z(P+Q)) = \mathrm{DiffAdd}(\widetilde{P}, \widetilde{Q}, \widetilde{P-Q})$

.

- $u = (X(P) + Z(P))(X(Q) - Z(Q))$
- $v = (X(P) - Z(P))(X(Q) + Z(Q))$
- $4X(P+Q)X(P-Q) = (u+v)^2$
- $4Z(P+Q)Z(P-Q) = (u-v)^2$

ALGORITHM 5.5. Affine cubical differential addition in the Montgomery model

coordinates $\frac{A+2}{4} = (A_{24} : C_{24})$. Here I gave the cubical formulas where C_{24} is normalised to 1. (Alternatively, using the notations from Algorithm 5.4, a C_{24} not normalised to 1 corresponds to doubling formulas given by $X(2P) = C_{24}ab, Z(2P) = c(C_{24}b + A_{24}c)$, and associated to a different normalisation of $\tilde{0}$, but then we would need to update the cubical differential additions for this different normalisation.) However, as explained in Remark 4.23, for pairings we only need the biextension arithmetic, which is more flexible than the cubical arithmetic. In particular, the extra factor 4 in the affine differential addition formulas do not matter for pairings (it does for other applications though): in the case of the Weil pairing they are compensated in the quotient λ_P/λ_Q, and in the case of the Tate pairing it is killed by the final exponentiation, as long as we are not over the base field $\mathbb{F}_p$.

To compute pairings for ℓ even, we also need to make a choice of affine lift of translation by points of 2-torsion (i.e, fix once and for all an element $(T, g_T) \in G(2(0_E))$ represented via its action on the (X, Z) coordinates). This is done in Algorithm 5.6. We remark also that for Montgomery Kummer lines, we could apply Algorithm 5.2 by normalising our points with respect to X rather than to Z, because the divisor of zeroes of X is $2(T)$, with $T = (0 : 1)$, so by Remark 4.27, X is also a suitable coordinate for normalisation.

Example 5.2. Let P be a point of 2-torsion. We want to compute the Tate pairing $e_{T,2}(P, Q)$ with respect to (0_E) rather than with respect to $2(0_E)$ (since the later would be trivial). By Remark 2.11, we can use either the X or Z coordinate, since the divisor of Z is $2(0_E)$ and the divisor of X is $2(T)$, with $T = (0 : 1)$, so they are both twice a rational divisor.

Let M_P be the matrix of translation by P in (X, Z) coordinates (formally this is the matrix of some theta group element $g_P \in G(2(0_E))$ above P action on X, Z). There are two approaches to this pairing computation:

Input: $\widetilde{P} = (X(P), Z(P))$ and $T = (X(T) : Z(T))$ a point of 2-torsion
Output: $\widetilde{P+T} = (X(P+T), Z(P+T)) = \text{Translate}(\widetilde{P}, T)$

.

- If $T = 0_E = (1:0)$, return $(X(P), Z(P))$
- If $T = (0:1)$, return $(Z(P), X(P))$
- Else $T = (X(T) : Z(T))$, we fix once and for all $\widetilde{T} = (X(T), Z(T))$ and return $(X(T)X(P) - Z(T)Z(P), Z(T)X(P) - X(T)Z(P))$

Algorithm 5.6. Affine cubical translation by a point of 2-torsion in the Montgomery model

- We start with $\widetilde{P}, \widetilde{Q}, \widetilde{P+Q}$ normalised to have $Z(\widetilde{P}) = Z(\widetilde{Q}) = Z(\widetilde{P+Q}) = 1$ (alternatively normalized by their X coordinates). We compute the monodromy actions $M_P \cdot \widetilde{P+Q} = \lambda_{P,1}\widetilde{Q}$, and $M_P \cdot \widetilde{P} = \lambda_{P,0}\tilde{0}$, and return the Tate pairing as $e_{T,2}(P,Q) = \lambda_{P,1}/\lambda_{P,0}$.
- We start with $\widetilde{Q}$ arbitrary, but we use $\widetilde{P} = M_P \cdot \tilde{0}$, $\widetilde{P+Q} = M_P \cdot \widetilde{Q}$. Since P is of two torsion, M_P^2 is a constant times the identity matrix: $M_P^2 = \lambda_P \text{Id}$. We have $M_P \cdot \widetilde{P+Q} = \lambda\widetilde{Q}$, $M_P \cdot \widetilde{P} = \lambda\tilde{0}$, so the monodromy $\lambda/\lambda = 1$ is trivial. However, since our points $\widetilde{P}, \widetilde{Q}, \widetilde{P+Q}$ are no longer normalized, we need to keep track of their coordinates to get the correct Tate pairing: $e_{T,2}(P,Q) = \frac{Z(\widetilde{P})Z(\widetilde{Q})}{Z(\widetilde{P+Q})}$.

It is not hard to see that both methods give the same result; this is a special case of the more general formula from Remark 2.11 (which handle the case where the pairing is recovered from both a non trivial monodromy action and a non trivial normalisation).

We stress that these two methods work for Z (or X), because their zero divisors are twice a divisor; it would not work for a random linear combination $U = u_0X + u_1Z$.

In practice, applying the second method, if $\widetilde{P} = (X(P), Z(P))$, $\widetilde{Q} = (X(Q), Z(Q))$, then $\widetilde{P+Q} = (X(P)X(Q) - Z(P)Z(Q), Z(P)X(Q) - X(P)Z(Q))$. We obtain $e_{T,2}(P,Q) = \frac{Z(\widetilde{P})Z(\widetilde{Q})}{Z(\widetilde{P+Q})} = \frac{Z(P)Z(Q)}{Z(P)X(Q)-X(P)Z(Q)} = \frac{1}{x(Q)-x(P)}$.

This is indeed the correct formula for the non reduced Tate pairing (recall that we take the opposite sign convention than the usual one): the normalised function $\mathbf{f}_{2,P} = x - x(P)$, so its evaluation at Q is $x(Q) - x(P)$.

Using the X coordinate instead, we have $e_{T,2}(P,Q) = \frac{X(\widetilde{P})X(\widetilde{Q})}{X(\widetilde{P+Q})X(\tilde{0})} = \frac{X(P)X(Q)}{X(P)X(Q)-Z(P)Z(Q)} = \frac{1}{1-\frac{1}{x(P)x(Q)}}$. And indeed, we have $\mathbf{f}_{2(P+T)-2(T)}((Q)-(0)) = \mathbf{f}_{2(P)-2(0)}((Q+T)-(T)) = \frac{x(Q+T)-x(P)}{x(T)-x(P)} = \frac{1/x(Q)-x(P)}{-x(P)} = 1 - \frac{1}{x(P)x(Q)}$.

Remark 5.3 (Complete arithmetic laws)**.** We remark that Algorithm 5.4 is always well defined, even for doubling $\tilde{0}$, and Algorithm 5.5 is also always well defined, except in the case where $\widetilde{P-Q}$ is above a point $P - Q = 0_E = (0:1)$ or $P - Q = T = (1:0)$. In the first case, we have $P = Q$, and so we can first compute $2\widetilde{Q} = \text{Double}(\widetilde{Q})$ and then adjust the result depending on the scalar thus that $\widetilde{P} = \lambda_P \cdot \widetilde{Q}$, $\widetilde{P-Q} = \lambda_0\tilde{0}$: $\text{DiffAdd}(\widetilde{P}, \widetilde{Q}, \widetilde{P-Q}) = \frac{\lambda_P^2}{\lambda_0}\text{Double}(\widetilde{Q})$. In the second case, we first lift the translation action by T by taking $\widetilde{T} = (1,0) = g_T \cdot \tilde{0}$, with $g_T \in G(2(0_E))$ satisfies $g_T \cdot (X,Z) = (Z,X)$. Now by Lemma 4.12,

$\mathrm{DiffAdd}(g_T \cdot \widetilde{Q}, \widetilde{Q}, \widetilde{T}) = g_T \cdot \mathrm{Double}(\widetilde{Q})$. We can thus compute $\mathrm{Double}(\widetilde{Q})$, apply g_T on it (i.e. switch the X and Z coordinates). Then we need to adjust by the scalars such that $\widetilde{P} = \lambda_P \cdot g_T \cdot \widetilde{Q}, \widetilde{P-Q} = \lambda_0 \widetilde{T}$. Then $\mathrm{DiffAdd}(\widetilde{P}, \widetilde{Q}, \widetilde{P-Q}) = \frac{\lambda_P^2}{\lambda_0} g_T \cdot \mathrm{Double}(\widetilde{Q})$.

This means that, adapting our formulas for these particular case, we have complete cubical arithmetic laws on Montgomery Kummer lines. In practice, for a pairing computation $e(P,Q)$, we can always use Algorithms 5.4 and 5.5, except when P, Q or $P+Q$ is equal to the two torsion point $T = (0:1)$. We might as well treat these cases globally, or use Corollary 4.13.

Still, it is an interesting exercice to unravel the cubical formulas for these special cases, and check we recover the correct pairings. Let $T = (1:0)$, we fix $\widetilde{T} = (1,0)$, since $\tilde{0} = (0,1)$ this means that the corresponding theta group element $g_T \in G(2(0_E))$ is the one that acts on (X,Z) by $(X,Z) \mapsto (Z,X)$.

If $P+Q = T$, then $Q = T - P$, so by bilinearity $e(P,Q) = e(P,T)e(P,-P)$, so we only need to treat the case $P = T$ and $Q = T$. Furthermore, these cases are only interesting when $\ell = 2m$ is even, otherwise the pairings are trivial.

Let us first work out the case $P = T$. We can take $\widetilde{P} = \widetilde{T} = (1,0)$, fix $\widetilde{Q} = (X(Q), Z(Q))$ arbitrarily, and take $\widetilde{P+Q} = \widetilde{T} \cdot \widetilde{Q} = (Z(Q), X(Q))$. Then $m\widetilde{P} = (1,0)$ if m is odd, and $(0,1)$ if m is even. But in the odd case, for the monodromy we need to act by $-m\widetilde{P} = (1,0)$, so $-m\widetilde{P} \cdot m\widetilde{P} = \tilde{0}$. By Corollary 4.13, $\widetilde{mP+Q} = \widetilde{Q}$ if m is even, and is equal to $\widetilde{P+Q}$ if m is odd. In both cases $-m\widetilde{P} \cdot \widetilde{mP+Q}$ is equal to $\widetilde{Q}$. So the monodromy is trivial in this case. Since the divisor is $2(0_E)$, this monodromy gives the square of the Tate pairing $e_{T,2}(P,Q)$, which is indeed trivial since P is of 2-torsion. By Remark 2.11, if we want to recover $e_{T,2}(P,Q)$, we need to correct the monodromy information by the coordinates $\left(\frac{Z(\widetilde{P+Q})}{Z(\widetilde{P})Z(\widetilde{Q})}\right)^m = (X(Q)/Z(Q))^m = x(Q)^m$. We indeed have $\mathbf{f}_{2m,P}((Q)-(0)) = \mathbf{f}_{2,P}((Q)-(0))^m = (x(Q)-x(P))^m = x(Q)^m$.

Now for the case $Q = T$; we take $\widetilde{Q} = (1,0)$, and $\widetilde{P+Q} = \widetilde{Q} \cdot \widetilde{P} = (Z(P), X(P))$. We compute $m\widetilde{P}$ and $\widetilde{mP+Q}$, then we act by $\widetilde{-mP}$ which we can take to be equal to $-m\widetilde{P}$. By Corollary 4.13, $\widetilde{mP+Q} = \widetilde{Q} \cdot \widetilde{mP}$. If g_Q, g_{-mP} are the theta group elements encoded by $\widetilde{Q}, -m\widetilde{P}$, it follows that the monodromy information we get is precisely the commutator pairing between g_Q, g_{-mP}, hence by Example 2.1 is equal to $e_{W,2}(-mP,Q)$. In particular, for the Weil pairing $e_{W,2m}(P,Q) = e_{W,2}(mP,Q)$, since the monodromy information on the preceding paragraph was trivial, it follows that the monodromy we compute with respect of both P,Q do give the Weil pairing. For the Tate pairing however, the monodromy is not enough, and we need to correct by the cubical function $\left(\frac{Z(\widetilde{P+Q})}{Z(\widetilde{P})Z(\widetilde{Q})}\right)^m = (X(P)/Z(P))^m = x(P)^m$. The end result we obtain for the Tate pairing via cubical arithmetic is $e_{T,2m}(P,Q) = x(P)^m e_{W,2}(mP,Q)$. Comparing with the usual formula, we indeed have $e_{T,2m}(P,Q) = \mathbf{f}_{2m,P}((Q)-(0)) = e_{W,2m}(P,Q)\mathbf{f}_{2m,Q}((P)-(0)) = e_{W,2}(mP,Q)x(P)^m$.

We obtain the following complexity, which complements Theorem 5.1, for pairings in the Montgomery model of Kummer lines:

Theorem 5.4. *Given the Kummer line Montgomery coordinates $(X(P):Z(P))$ of $P, Q, P+Q$, first do a batch inversion to compute $x(P) = X(P)/Z(P)$, $x(Q)$, $x(P+Q)$ and their inverses, and also compute $\frac{A+2}{4}$. Take for cubical points $\tilde{0} = (1,0)$, $\widetilde{P} = (x(P),1)$, $\widetilde{Q} = (x(Q),1)$, $\widetilde{P+Q} = (x(P+Q),1)$.*

Then during the cubical biextension ladder, each affine doubling costs $2M + 2S + 1m_0$ and each affine differential addition costs $3M + 2S$.

Input: $P_1 = (X(P_1) : Z(P_1)), P_2 = (X(P_2) : Z(P_2), P_1 + Q = (X(P_1 + Q) : Z(P_1 + Q)), P_2 + Q = (X(P_2 + Q) : Z(P_2 + Q))$
Output: $P_1 + P_2 = (X(P_1 + P_2) : Z(P_1 + P_2)$

.

- $\kappa_{00} = (X(P_1)X(P_2) - Z(P_1)Z(P_2))^2$
- $\kappa_{11} = (X(P_1)Z(P_2) - X(P_2)Z(P_1)^2$
- $\kappa_{01} = 2((X(P_1)X(P_2) + Z(P_1)Z(P_2))(X(P_1)Z(P_2) + X(P_2)Z(P_1)) + 2\mathcal{A}X(P_1)X(P_2)Z(P_1)Z(P_2))$
- $\kappa'_{00} = (X(P_1 + Q)X(P_2 + Q) - Z(P_1 + Q)Z(P_2 + Q))^2$
- $\kappa'_{11} = (X(P_1 + Q)Z(P_2 + Q) - X(P_2 + Q)Z(P_1 + Q)^2$
- $\kappa'_{01} = 2((X(P_1 + Q)X(P_2 + Q) + Z(P_1 + Q)Z(P_2 + Q))(X(P_1 = Q)Z(P_2 + Q) + X(P_2 + Q)Z(P_1 + Q)) + 2\mathcal{A}X(P_1 + Q)X(P_2 + Q)Z(P_1 + Q)Z(P_2 + Q))$
- $X(P_1 - P_2) = \kappa'_{01}\kappa_{00} - \kappa_{01}\kappa'_{00}$
- $Z(P_1 - P_2) = \kappa'_{11}\kappa_{00} - \kappa_{11}\kappa'_{00}$
- $X(P_1 + P_2) = \frac{\kappa_{00}}{X(P_1 - P_2)}$
- $Z(P_1 + P_2) = \frac{\kappa_{11}}{Z(P_1 - P_2)}$

Algorithm 5.7. Compatible additions in the Montgomery model

Theorem 1.1 follows from Theorems 5.1 and 5.4.

5.2.2. *Double and add biextension algorithm for the Montgomery model.* For the double and add algorithm, we need to give the formulas for the compatible addition. Using the notations of Section 4.9.4, we have

$$\kappa_{00} = (X(P_1)X(P_2) - Z(P_1)Z(P_2))^2,$$

$$\kappa_{11} = (X(P_1)Z(P_1) - Z(P_1)X(P_2)),$$

by the differential addition formulas from above. Lastly,

$$\kappa_{01} = 2((X(P_1)X(P_2)+Z(P_1)Z(P_2))(X(P_1)Z(P_2)+X(P_2)Z(P_1))+2\mathcal{A}X(P_1)X(P_2)Z(P_1)Z(P_2))$$

by homogenisation of [BDLS20, Example 4.4].

For the three way addition, Section 4.9.4 already gives the formula for Z. We can obtain $X(P_1 + \widetilde{P_2} + Q)$ from the formula for Z simply by replacing Q by $Q + T$:

$$X(P_1 + \widetilde{P_2} + Q) = \frac{(X(\widetilde{P_2})X(P_1\widetilde{+}Q) - Z(P_1\widetilde{+}Q)Z(\widetilde{P_2}))(X(\widetilde{P_1})X(P_2\widetilde{+}Q) - Z(P_2\widetilde{+}Q)Z(\widetilde{P_1}))}{(X(P_1\widetilde{-}P_2)X(\widetilde{Q}) - Z(P_1\widetilde{-}P_2)Z(\widetilde{Q}))}.$$

Algorithms 5.7 and 5.8 give the algorithms for the compatible addition and the three way cubical addition in the Montgomery model respectively.

We can combine Algorithms 5.7 and 5.8 to compute the addition step of Algorithm 4.7. We compute $P_1 + P_2$ by a compatible addition, then take an arbitrary lift $P_1\widetilde{+}P_2$ and then apply the three way addition to obtain $P_1 + \widetilde{P_2} + Q$. We remark that in Algorithm 5.8, we already have $P_1\widetilde{-}P_2$ from Algorithm 5.4. Furthermore, while we describe these algorithms with divisions for simplicity, we can use Lemma 4.15 to remove all divisions: we can multiply $P_1\widetilde{+}P_2, P_1 + \widetilde{P_2} + Q$ by the same projective factor λ and still get the same biextension element $g_{P_1+P_2,Q}$.

Input: $\widetilde{P}_1 = (X(P_1), Z(P_1)), \widetilde{P}_2 = (X(P_2), Z(P_2), P_1\widetilde{+}Q = (X(P_1+Q), Z(P_1+Q)),$
$P_2\widetilde{+}Q = (X(P_2+Q) : Z(P_2+Q)), P_1\widetilde{+}P_2 = (X(P_1+P_2), Z(P_1+P_2))$
Output: $P_1 + \widetilde{P_2} + Q = (X(P_1 + P_2 + Q), Z(P_1 + P_2 + Q)) =$
$\mathrm{ThreeWayAdd}(\widetilde{P}_1, \widetilde{P}_2, \widetilde{Q}, P_1\widetilde{+}Q, P_2\widetilde{+}Q, P_1\widetilde{+}P_2)$

.

➔ Compute $(X(P_1\widetilde{-}P_2), Z(P_1\widetilde{-}P_2)) = \mathrm{DiffAdd}(\widetilde{P}_1, \widetilde{P}_2, P_1\widetilde{+}P_2)$
➔ $X(P_1 + \widetilde{P_2} + Q) = \frac{(X(P_2)X(P_1+Q)-Z(P_1+Q)Z(P_2))(X(P_1)X(P_2+Q)-Z(P_2+Q)Z(P_1))}{X(P_1-P_2)X(Q)-Z(P_1-P_2)Z(Q)}$
➔ $Z(P_1 + \widetilde{P_2} + Q) = \frac{(X(P_2)Z(P_1+Q)-X(P_1+Q)Z(P_2))(X(P_1)Z(P_2+Q)-X(P_2+Q)Z(P_1))}{X(P_1-P_2)X(Q)-Z(P_1-P_2)Z(Q)}$

ALGORITHM 5.8. Three way cubical additions in the Montgomery model

In practice, this means the we compute the numerators n and denominators d of $X(P_1 + \widetilde{P_2} + Q)$ and $Z(P_1 + \widetilde{P_2} + Q)$ separately, and use the formulas

$$\begin{aligned}
X(P_1 + P_2) &= \kappa_{00} Z(P_1 - P_2) d(X(P_1 + P_2 + Q)) d(Z(P_1 + P_2 + Q)),\\
Z(P_1 + P_2) &= \kappa_{11} X(P_1 - P_2) d(X(P_1 + P_2 + Q)) d(Z(P_1 + P_2 + Q)),\\
X(P_1 + P_2 + Q) &= n(X(P_1 + P_2 + Q) d(Z(P_1 + P_2 + Q)) X(P_1 - P_2) Z(P_1 - P_2),\\
Z(P_1 + P_2 + Q) &= n(Z(P_1 + P_2 + Q) d(X(P_1 + P_2 + Q)) X(P_1 - P_2) Z(P_1 - P_2).
\end{aligned}$$

We could use a similar strategy to compute a biextension double $2\widetilde{P}, 2\widetilde{P+Q}$, but it is faster to use a cubical double and differential addition for this case, as explained in Section 4.7.

These algorithms combined give us the $32M + 4S + 2m_0$ count for an addition. There is probably room for improvement for these formulas by rearranging the arithmetic operations. Also it could potentially be faster to do a compatible addition to compute $P_1 + P_2 + Q$ projectively rather than $P_1 - P_2$, and then using cubical arithmetic to get $P_1\widetilde{+}P_2$ from a choice of $P_1 + \widetilde{P_2} + Q$.

We refer to the implementation in [Rob23b] for a variant which combines a double and addition into a DoubleAndAdd, which costs $17M + 8S + 3m_0$ (so strangely, is faster than just an addition, but this is probably an artefact of the fact that the addition is not at all optimised).

5.2.3. *Results.* Giacomo Pope kindly implemented our algorithm in Rust. Section 5.2.3 give some timings on a Intel Core i5-1335U, with turboboost disabled, on a supersingular curve E over $\mathbb{F}_{p^2}$ with $p = 2^{604} \cdot 3^{363} - 1$ and $p = 2^{74} \cdot 3^{41} - 1$ respectively.

	2^{604}	3^{363}	$2^{604} \cdot 3^{363}$	2^{74}	3^{41}	$2^{74} \cdot 3^{41}$
Tate pairing	26.6 ms	34.7 ms	57.9 ms	80.6 µs	100.8 µs	166.4 µs
Weil pairing	35.2 ms	50.9 ms	103.84 ms	112.8 µs	149.7 µs	300.9 µs

One nice advantage of Theorem 1.1 is that the cubical ladder, since it is derived from the Montgomery ladder, is naturally constant time. Constant time constraints are different for pairings than for standard elliptic curve cryptography: in elliptic curve cryptography the scalar is secret, while the base point may or may not be public. For pairings, the scalar is usually public, but the paired points may be secret. The pairing arithmetic such need to not

depend on the paired point. By Remark 5.3, this is naturally the case for the cubical ladder in the Montgomery model, except for a few exceptions (which do not happen when ℓ is odd).

Compared to a previous constant time implementation using Miller's algorithm, our new algorithm is between 2.5× to 3× faster over $\mathbb{F}_{p^2}$ for $p = 2^{74} \cdot 3^{41} - 1$.

A last advantage of Theorem 1.1 is that in isogeny based cryptography, we often represent a basis (P_1, P_2) of N-torsion by $x(P_1), x(P_2), x(P_1 + P_2)$ (or via their $(X : Z)$ projective coordinates). Indeed, it usually does not matter if we compute an isogeny ϕ or its opposite $-\phi$, so it is harmless to replace the basis (P_1, P_2) by $(-P_1, -P_2)$, hence represent it by $x(P_1), x(P_2), x(P_1 + P_2)$. Since Theorem 1.1 naturally take such a data as input, it allows to compute the pairing $e(P_1, P_2)$ without the need of taking a square root first.

5.3. **The theta model.** Explicit formulas in the level 2 theta model (in any dimension) are given in [LR16]. Although it was not stated in this form there, the formulas compute the correct cubical arithmetic.

In the theta model, a cubical differential addition (with the base point normalised) costs $2S + 3M$, and a doubling costs $3S + 1M + 2m_0$. Since the cubical biextension ladder uses one doubling and two differential additions by bits, the total cost is $7S + 7M + 2m_0$.

There is a variant which costs $2S + 3M + 1m_0$ for a differential addition, and $4S + 2m_0$ for a doubling; using this variant the cubical biextension ladder than costs $8S + 6M + 4m_0$ by bits.

5.4. **Short Weierstrass curves.** On a short Weierstrass curve $y^2 = x^3 + ax + b$,, x-only doublings and differential additions formulas are given by:

$$x(P+Q)x(P-Q) = \frac{-4b(x(P)+x(Q)) + (x(P)x(Q)-a)^2}{(x(P)-x(Q))^2} = \frac{-4bZ(P)Z(Q)(X(P)Z(Q)+X(Q)Z(P)) + (X(P)X(Q$$

$$x(2P) = \frac{(x(P)^2-a)^2 - 8bx(P)}{4(x(P)^3+ax(P)+b} = \frac{(X(P)^2-aZ(P))^2 - 8bX(P)Z(P)^3}{4Z(P)(X(P)^3+aX(P)Z(P)^2+bZ(P)^3)}.$$

Exactly as for Montgomery curves the denominators are precisely the ones coming from the cubical arithmetic:

$$Z(\widetilde{P+Q})Z(\widetilde{P-Q}) = \left(X(\widetilde{Q})Z(\widetilde{P}) - X(\widetilde{P})Z(\widetilde{Q})\right)^2,$$

$$Z(2\widetilde{P}) = 4(X(\widetilde{P})^3Z(\widetilde{P}) + aX(\widetilde{P})Z(\widetilde{P})^3 + bZ(\widetilde{P})^4).$$

This means that the standard ladder [BJ02] on short Weierstrass curves already compute the correct cubical arithmetic (modulo the fact that, in the ladder, we need to divide the X coordinate by $x(P_0)$ where $P_0 = (x(P_0), 1)$ is the base point, rather than multiplying the Z coordinate by $x(P_0)$, in order to keep track of the correct cubical factor).

Thus a cubical doubling costs $3M+4S+2m_0$ and a cubical differential addition costs $6M+2S+2m_0$, so a cubical biextension ladder costs a quite expensive $15M+8S+6m_0$ by bits. This allows to compute pairings on short Weierstrass curves knowing only $x(P), x(Q), x(P+Q)$.

We leave for future work improving these formulas. Maybe keeping track of the Y coordinate could help. An alternative approach would be to use a Montgomery ladder in co-Z coordinates [Mel07; GJMRV11; Ham20], since these gives faster formula. But we would need to extend this co-Z approach to a 3 point ladder (i.e., a ladder computing ℓP, $\ell P + Q$), and we would still need to track the correct cubical factor for each point somehow (scaling to the same co-Z coordinates on mP, $(m+1)P$, $(m+1)P+Q$ kills the cubical information if we do not keep it somewhere else).

6. Applications

We discuss various applications of the cubical arithmetic. In Section 6.1 we look at pairing based cryptography and how the cubical arithmetic on Montgomery models from Section 5 compares with the usual Miller's algorithm. We also derive new (to my knowledge), more or less interesting, formulas for the standard Miller algorithm by using the cubical point of view.

In Section 6.2, we briefly mention some applications of cubical arithmetic beyond pairings, notably on isogenies and radical isogenies.

In Section 6.3, we reinterpret, using cubical arithmetic, Doliskani's supersingularity test (see [Dol18; BGS22]) as a self Tate pairing test.

Finally, our most important application is probably Section 6.4 where we use cubical arithmetic to obtain a new powerful side channel attack against the Montgomery ladder for Montgomery curves. Namely, one projective coordinate leak in the Montgomery ladder allows to solve the elliptic curve dlp by reduction to the base field dlp.

Each of these applications would deserve an article on its own[2]. We won't detail these applications much in this paper, because it is quite long already.

6.1. **Pairing based cryptography.**

6.1.1. *Comparison with Miller's algorithm for pairing based cryptography.* For pairing based cryptography on elliptic curves, with embedding degree $d > 1$, it is convenient to use the Tate pairing with $P \in \mathbb{G}_1 \subset E(\mathbb{F}_q)$, $Q \in \mathbb{G}_2 \subset E(\mathbb{F}_{q^d})$, and d even to allow for denominator elimination. We recall that $Q \in \mathbb{G}_2$ if and only if $\pi_q(Q) = qQ$.

Counting only operations involving the big field $\mathbb{F}_{q^d}$, Miller's algorithm, with denominator elimination, costs $1M + 1S + 1m$ by doubling, and $1M + 1m$ by addition. Here $1m$ denotes a "mixed multiplication", meaning a multiplication between a coefficient in $\mathbb{F}_q$ and a coefficient in $\mathbb{F}_{q^d}$.

When denominator elimination is not possible (because d is odd or Q is not in $\mathbb{G}_2$), the cost becomes $2M + 2S + 1m$ by doubling, and $2M + 1m$ by addition. (In practice, working with projective coordinates for P, the number of mixed multiplications m is a bit higher, see Section 6.1.2.) On the other hand, when d is odd, there are variants of Miller's algorithm in the literature (notably [BELL10]) which we will discuss in Section 6.1.2, which achieve $2S + 1M + 3m$ for a doubling and $1M + 2.5m$ for an addition.

Using our arithmetic of biextension on Kummer lines in the Montgomery model, only counting the operations on the big field, we have $2S + 1M + 2m$ by bit for the cubical biextension ladder. This is better than Miller's algorithm (even the improved variant), except when denominator elimination is available.

Our main difficulty is that it is not clear how to do an efficient denominator elimination in cubical arithmetic when the embedding degree is even. If d is even and $Q \in \mathbb{G}_2$, then $x(Q)$ lies in a strict subfield of $\mathbb{F}_{q^d}$. This allows to do denominator elimination, i.e. not compute the denominator of the Miller functions $\mu_{uP,vP}$ evaluated at Q, since this evaluated denominator is $x(Q) - x((u+v)P)$ is in a strict subfield, hence killed by the final exponentiation.

In Theorem 1.1, for $P \in \mathbb{G}_1, Q \in \mathbb{G}_2$ and d even, then while $x(P), x(Q)$ lies in a subfield, we also need to use $x(P+Q)$ which does not. This means that for the cubical biextension

[2]Which I cannot promise I will ever write someday; I already made such promises in previous papers and so far I have a bad track record on keeping these…The only exception will hopefully be the monodromy leak attack, because several people have encouraged me to publish it in its own paper, rather than to hide it at the end of a paper about biextensions.

ladder, the cost is $2S + 1M + 2m$ when the current bit is 1 and the difference point is $P + Q$, and $2S + 1M' + 2m$ when the current bit is 0 and the difference point is Q, where M' means that we multiply by a coordinate in $\mathbb{F}_{p^{d/2}}$. So even embedding degree d still helps for the cubical ladder.

Another way it helps, which is well known, is that an inverse in $\mathbb{F}_{q^d}$ can be written $1/x = \overline{x}/(x\overline{x})$ where $\overline{x}$ is the quadratic Galois conjugate of x and in particular $x\overline{x} \in \mathbb{F}_{q^{d/2}}$. In situations where $x\overline{x}$ will be killed by the final exponentiation, this allows to use $\overline{x}$ instead of $1/x$, hence replace a division by (one or several) multiplications.

It could be interesting to work out formulas where we represent $m\widetilde{P+Q}$ using only $Z(m\widetilde{P+Q})$ rather than via $(X(m\widetilde{P+Q}), Z(m\widetilde{P+Q}))$.

On the other hand, for the Ate or optimal Ate pairing, since the computation is done with $P \in \mathbb{G}_2$ and $Q \in \mathbb{G}_1$, it is plausible that the cubical arithmetic could be faster than the usual Miller's algorithm, since the overall number of operations for a generic pairing is much smaller.

6.1.2. *Cubical arithmetic and new formulas for Miller's algorithm.* In this section, rather than comparing Miller's algorithm with the cubical arithmetic on Kummer lines, we will compare Miller's algorithm with the cubical arithmetic on elliptic curves, where we represent $\widetilde{P}$ by using the level 3 projective Weierstrass coordinates $P = (X(P) : Y(P) : Z(P))$ along with the level 1 affine cubical coordinate $Z_1(\widetilde{P})$.

To simplify, we will consider the case of the Tate pairing $P \in \mathbb{G}_1, Q \in \mathbb{G}_2$. In Miller's algorithm, we compute $f_{\ell,P}((Q) - (0))$. In the cubical arithmetic, we compute $\ell\widetilde{P}$, $\ell\widetilde{P+Q}$. The relationship between the two approaches is given by Porism 4.25. In particular, assuming we fix $\widetilde{P}, \widetilde{Q}, \widetilde{P+Q}$ so that $Z_1(\widetilde{P}) = Z_1(\widetilde{Q}) = Z_1(\widetilde{P+Q}) = 1$, under our assumptions on P, Q, we have that $Z_1(\ell\widetilde{P+Q})$ is equal to $f_{\ell,P}((Q) - (0))$ up to a factor lying in the small field $\mathbb{F}_q$, which will be killed by the final exponentiation.

Now we already saw in Remark 3.16 that the biextension arithmetic provided new insight on how to compute $f_{\ell,P}$ via the functions $\mu_{uP,vP}$. We will now derive new formulas using the cubical arithmetic.

Let's start by formulas to compute $Z_1(\ell\widetilde{P+Q})$, using Section 4.9.3. We have the doubling formula

$$Z_1(\widetilde{2mP}) = 2y(mP)^2 Z_1^4(\widetilde{mP})$$

and the differential addition formulas

$$Z_1((\widetilde{2m+1})P)Z_1(\widetilde{P}) = Z_1((\widetilde{m+1})P)^2 Z_1(\widetilde{mP})^2(x(mP) - x((m+1)P)),$$
$$Z_1(\widetilde{2mP+Q})Z_1(\widetilde{Q}) = Z_1(\widetilde{mP+Q})^2 Z_1(\widetilde{mP})^2(x(mP) - x(mP+Q)),$$
$$Z_1((\widetilde{2m+1)P+Q})Z_1(\widetilde{P+Q}) = Z_1((\widetilde{m+1)P+Q})^2 Z_1(\widetilde{mP})^2(x(mP) - x((m+1)P+Q)).$$

This allows to compute $\ell\widetilde{P+Q}$ via a cubical ladder using the points $\widetilde{mP}$, $(\widetilde{m+1})P$, $(\widetilde{m+1)P} + Q$. (Note the similarity with Remark 3.16).

We could also use a double and add algorithm, using

$$Z_1(\widetilde{2mP+Q})Z_1(\widetilde{Q}) = Z_1(\widetilde{mP+Q})^2 Z_1(\widetilde{mP})^2(x(mP) - x(mP+Q))$$

for doublings, and the three way addition

$$Z_1(P + \widetilde{mP} + Q) = \frac{Z_1(\widetilde{P+mP})Z_1(\widetilde{mP+Q})Z_1(\widetilde{P+Q})}{Z_1(\widetilde{P})Z_1(\widetilde{mP})Z_1(\widetilde{Q})} \frac{l_{mP,P}(Q)}{(x(Q) - x(mP))(x(Q) - x(P))},$$

for an addition. We could also compute the doubling using the following three way addition:

$$Z_1(mP + \widetilde{mP} + Q) = \frac{Z_1(mP\widetilde{+mP})Z_1(mP\widetilde{+}Q)^2}{Z_1(\widetilde{mP})^2 Z_1(\widetilde{Q})} \frac{l_{mP,mP}(Q)}{(x(Q) - x(mP))^2)}.$$

I implemented both methods and the pairing is computed correctly, but these formulas seem a priori no faster than Miller's standard algorithm.

However, the following approach seems the most promising: use a double and add algorithm, using

$$Z_1(2m\widetilde{P+}Q)Z_1(\widetilde{Q}) = Z_1(m\widetilde{P+}Q)^2 Z_1(\widetilde{mP})^2(x(mP) - x(mP+Q))$$

as before for doublings, and

$$Z_1((2m+\widetilde{1)P} + Q)Z_1(P\widetilde{+}Q) = Z_1((m+\widetilde{1)P} + Q)^2 Z_1(\widetilde{mP})^2(x(mP) - x((m+1)P+Q))$$

for a doubling and addition.

Now we remark that the function $Q \mapsto x(mP) - x(mP+Q)$ has for divisor $(0_E) + (-2mP) - 2(-mP)$ so is equal to a biextension function $g_{mP,mP}(Q)$. In particular, $(x(mP) - x((m+1)P+Q))$ is given by $g_{mP,mP}(P+Q)$.

This gives the following alternative strategy to compute the (normalised) Miller function $f_{\ell,P}(Q)$. Usually we use $f_{u+v,P}(Q) = f_{u,P}(Q)f_{v,P}(Q)\mu_{uP,vP}(Q)$ with $\mu_{uP,vP}$ the normalised function with divisor $D_{uP+vP} - D_{uP} + D_{vP}$. In particular, the above formula gives the same Miller doubling: $f_{2u,P}(Q) = f_{u,P}(Q)^2\mu_{uP,uP}(Q)$. But we also have the following Miller DoubleAndAdd: $f_{2u+1,P}(Q) = Cf_{u+1,P}(Q)^2\mu_{uP,uP}(P+Q)$. Here the constant C refers to the fact that the function $Q \mapsto \mu_{uP,uP}(P+Q)$ is no longer normalised; however under our assumptions that $P \in \mathbb{G}_1$, C is in a strict subfield so will be killed by the final exponentiation. Of course, this formula can be proved directly without going through the cubical arithmetic: the left hand side has divisor $D_{(2u+1)P} - (2u+1)D_P$ while the right hand side has divisor $2D_{(u+1)P} - 2(u+1)D_P + t_P^*(D_{2uP} - 2D_{uP}) = 2D_{(u+1)P} - 2(u+1)D_P + D_{(2u+1)P} - 2D_{(u+1)P} + D_P = D_{(2u+1)P} - (2u+1)D_P$.

Using the standard sign conventions, the latter formulas reads:

$$\mathbf{f}_{2u+1,P}(Q) = C\mathbf{f}_{u+1,P}(Q)^2\boldsymbol{\mu}_{uP,uP}(Q-P).$$

This gives the following double and DoubleAndAdd algorithm: at each step we compute mP or $(m+1)P$ along with $\mathbf{f}_{m+1,P}(Q)$ up to some factor in a subfield.

We use either: $\mathbf{f}_{2m+1,P}(Q) = C_1\mathbf{f}_{m+1,P}(Q)^2\boldsymbol{\mu}_{mP,mP}(Q-P)$ (in which case we need to compute mP if we had $(m+1)P$, and we also compute $2mP$) or $\mathbf{f}_{2m+2,P}(Q) = C_2\mathbf{f}_{m+1,P}(Q)^2\boldsymbol{\mu}_{(m+1)P,(m+1)P}(Q)$ (in which case we need to compute $(m+1)P$ if we had mP, and we also compute $2(m+1)P$). This makes a DoubleAndAdd very similar to a simple doubling.

We compute in this way $\mathbf{f}_{\ell,P}(Q)$ (or $\mathbf{f}_{\ell+1,P}(Q)$ if we prefer). As far as I know, this DoubleAndAdd method is new. We refer to [Rob23b] for the implementation.

Let us compare this strategy with known optimisation of Miller's algorithm. In [BELL10], they use the formula $\mathbf{f}_{u+v,P}(Q) = \frac{1}{\mathbf{f}_{-u,P}(Q)\mathbf{f}_{-v,P}(Q)l_{-uP,-vP}(Q)}$ Compared to the usual formula which involves $\mu_{uP,vP}(Q) = l_{uP,vP}(Q)/v_{uP+vP}(Q)$, where $v_{uP+vP} = l_{uP+vP,-uP-vP}$, this saves the denominator $v_{uP+vP}(Q)$ of $\mu_{uP,vP}$.

In [DZZZ22] the authors introduce functions with divisors $\ell(P) + (-\ell P) - (\ell+1)(0)$ which give a streamlined double and add formula. We remark that such a function is given by $\mathbf{f}_{-\ell,P}$, so [DZZZ22] is subsumed by the formula above from [BELL10].

Under the assumption $P \in \mathbb{G}_1, Q \in \mathbb{G}_2$ and counting only operations in the big field, [BELL10] have a complexity (for d odd) of $2S + 1M + 3m$ for a doubling (compared to $2S + 2M + 5m$ for the classical Miller formula), and $1M + 2.5m$ for an addition (compared to $2M + 5m$ for the classical Miller formula); so a DoubleAndAdd costs $2M + 2S + 5.5m$ (compared to $4M + 2S + 10m$). When d is even, they have the same complexity (in the big field) as the usual Miller's formula of $1M + 1S + 1m$ for a doubling and $1M + 1m$ for an addition; so a DoubleAndAdd costs $2M + 1S + 2m$. (We assume here that multiplying an element in $\mathbb{F}_p$ with an element of $\mathbb{F}_{p^{d/2}}$ is half the cost of a multiplication of an element in $\mathbb{F}_p$ with an element in $\mathbb{F}_{p^d}$.)

Our new DoubleAndAdd method cost $2S + 2M + 5m$ both for a doubling and a DoubleAndAdd. When d is even, we have the same difficulty as in Section 6.1.1 that $x(P + Q)$ does not lies in a subfield. If $\mathbb{F}_{p^d} = \mathbb{F}_{p^{d/2}}[i]$ with $i^2 = c$, we can use that $1/(x + iy) = (x - iy)/(x^2 - dy^2)$ where $x, y \in \mathbb{F}_{p^{d/2}}$ and since $x^2 - dy^2$ is in a strict subfield, this allows to replace divisions by multiplications. The cost becomes $2M+1S+3m$ for a DoubleAndAdd which involves $x(P + Q)$ and $1M + 1S + 1m$ for a doubling which involves $x(Q)$ which is in a subfield. So for d odd, the DoubleAndAdd method is better than the usual Miller's formula but not better than [BELL10], and for d even roughly similar to [BELL10].

However we can combine the DoubleAndAdd variant with the following line folding trick (see [LL11, Lemma 3.2] and the references): $\frac{l_{T,T}(Q)}{v_t^2(Q)v_{2T}(Q)} = \frac{1}{l_{-T,-T}(Q)}$. This allows to delay a vertical line evaluation and fold it in a standard line evaluation. With the DoubleAndAdd algorithm, since we change the evaluated point according to whether the current bit is 0 or 1, we can only apply this line folding trick for consecutive 0s or 1s. Using this line folding trick, [LL11] obtain a complexity, for d odd and forgetting mixed additions, of $2S + 1M$ for a doubling and $2S + 2M$ for a double and add. (They give $2S + 1M$ or $2S + 2M$ according to which branch of the algorithm is evaluated, but it seems to me that in the branch where a parabola is computed one should account that evaluating this parabola costs $2M$). By comparison, our DoubleAndAdd algorithm (and also not counting mixed additions), we have a cost of $2S + 1M$ for consecutive doublings or DoubleAndAdds, and, as we have seen above, a cost of $2S + 2M$ when we switch from a bit equal to 0 to a bit equal to 1 or conversely. In the case when d is even, using the line folding trick, consecutive DoubleAndAdds cost $1M+1S$, the same as doublings (whether they are consecutive or not). So this DoubleAndAdd algorithm could be interesting when ℓ has low Hamming weight.

6.2. **Algorithms for abelian varieties.** The cubical arithmetic we develop has important applications for algorithms on abelian varieties beside pairings.

Notably, we will see that explicit formulas for the cubical arithmetic gives explicit formulas for the theta group action (at level ℓ). Using the generic isogeny framework from [Rob21a, § 2.9, § 4.2], this explicit action gives formulas for isogenies (or even allows to compute a basis of algebraic theta functions or to change levels). We have also just seen in this paper how the cubical arithmetic allows to compute pairing.

Now by the construction of the algebraic Riemann relations from Proposition 4.1, the cubical arithmetic can be derived from explicit formulas for the theorem of the square: $t_{P+Q}^*\mathcal{L} \otimes \mathcal{L} \simeq t_P^*\mathcal{L} \otimes t_Q^*\mathcal{L}$, which also (implicitly) encodes addition formulas. In summary: explicit formulas for the theorem of the square, which induces explicit formulas for the cubical arithmetic, is key for all standard algorithms on abelian varieties: additions, pairings, isogenies.

6.2.1. *Theta group arithmetic of higher level through cubical arithmetic.* First, we show that cubical arithmetic gives a convenient way to work out the arithmetic of theta group of level ℓn while using coordinates of level n evaluated on cubical points of ℓ-torsion. As explained in [LR22a], having this explicit theta group action can then be used to compute isogenies by taking suitable traces; so the cubical arithmetic can be used to find isogeny formulas in any models (by contrast to [LR12; CR15; LR22a] which use the theta model).

The basic idea is as follows: let $(A, \mathcal{L})$ be a polarised abelian variety. To compute ℓ-isogenies, we need to work out the arithmetic of the theta group $G(\mathcal{L}^\ell)$. We recall from Section 2 that an element $(P, g_{\mathcal{L}^\ell,P}) \in G(\mathcal{L}^\ell)$ is such that $P \in \operatorname{Ker} \Phi_{\mathcal{L}^\ell}$ where $\Phi_{\mathcal{L}^\ell}(P) = t_P^*\mathcal{L}^\ell \otimes \mathcal{L}^{-\ell}$ is the polarisation associated to $\mathcal{L}^\ell$ and $g_{\mathcal{L}^\ell,P}$ is a section of $t_P^*\mathcal{L}^\ell \otimes \mathcal{L}^{-\ell}$.

In particular, a cubical point $\widetilde{P}$ for $\mathcal{L}^\ell$ is enough to determine $g_{\mathcal{L}^\ell,P}$. Now a cubical point $\widetilde{P}$ for $\mathcal{L}$ does determine a cubical point for $\mathcal{L}^\ell$, simply by taking the tensor product to the ℓ of the rigidification at P; we will denote this by $\widetilde{P}^{\otimes\ell}$. (In particular, $\zeta\widetilde{P}$ induce the same cubical point for $\mathcal{L}^\ell$ whenever $\zeta^\ell = 1$.) We remark also that if $[\widetilde{P}_1, \widetilde{P}_2, \widetilde{P}_3, \widetilde{P}_4; \widetilde{Q}_1, \widetilde{Q}_2, \widetilde{Q}_3, \widetilde{Q}_4]$ are in Riemann position, then certainly their ℓ-fold tensor product are in Riemann position; so $\widetilde{P} \mapsto \widetilde{P}^{\otimes\ell}$ is compatible with the cubical arithmetic with respect to $\mathcal{L}$ and $\mathcal{L}^\ell$ respectively. Furthermore, if γ is the function associated to the Riemann relation $[P_1, P_2, P_3, P_4; Q_1, Q_2, Q_3, Q_4]$ for $\mathcal{L}$, then γ^ℓ is the corresponding function for $\mathcal{L}^\ell$. Finally, if $X_1, \dots, X_\ell$ are sections of $\Gamma(\mathcal{L})$, then $X = \prod_{i=1}^\ell X_i$ is a section of $\Gamma(\mathcal{L}^\ell)$, and $X(\widetilde{P}^{\otimes\ell}) = \prod_{i=1}^\ell X_i(\widetilde{P})$.

This allows to work on the theta group $G(\mathcal{L}^\ell)$ (i.e. at level ℓ) while using cubical points for $\mathcal{L}$ (i.e. at the base level). (We remark that the method of Remark 2.11, to compute the ℓm Tate pairing for D_1 when working with $D = mD_1$ goes in the other direction: we work with cubical points for D, but we try to normalise things such that the cubical points come from the m-fold tensor product of cubical points on D_1; that's how I found out this trick!)

We illustrate this approach in the following situation. Let $K \subset A[\ell]$ be an isotropic subgroup. By [Rob21a, Chapter 2], to compute isogeny formulas, we first need to compute a lift $\widetilde{K}$ of K to the theta group $G(\mathcal{L}^\ell)$, and then compute the action of $\widetilde{K}$ on sections of $\Gamma(\mathcal{L}^\ell)$.

If $P \in A[\ell]$, and $\widetilde{P}$ is a cubical point with respect to $\mathcal{L}$ then we have seen above that $\widetilde{P}^{\otimes\ell}$ is a cubical point with respect to $\mathcal{L}^\ell$, hence encodes a theta group element $g_P \in G(\mathcal{L}^\ell)$. This element g_P acts on the sections of $\Gamma(\mathcal{L}^\ell)$. Here is how we can recover this action using the cubical point $\widetilde{P}$, in the particular case when the section $u \in \Gamma(\mathcal{L}^\ell)$ can be written as $u = \prod_{i=1}^\ell u_i$, where $u_i \in \Gamma(\mathcal{L})$.

First, we need to take $\widetilde{P}$ such that g_P is an element of ℓ-torsion. If $\mathcal{L}$ is symmetric, it if often convenient to add the further constraint that g_P should be symmetric: when $\mathcal{L}$ is symmetric there is a canonical involution δ_{-1} on $G(\mathcal{L}^\ell)$ defined in [Mum66], and an element is symmetric if $\delta_{-1}g_P = g_P^{-1}$. We will see below the definition of a canonical lift of ℓ-torsion $\widetilde{P}$, and that such a $\widetilde{P}$ gives an element $\widetilde{P}^{\otimes\ell}$ such that the associated theta group element $g_P \in G(\mathcal{L}^\ell)$ is symmetric of ℓ-torsion.

Next, for a $R \in A$, we take an arbitrary cubical point $\widetilde{R}$ for $\mathcal{L}$, and we take $\widetilde{R+P}$ such that $\widetilde{R+\ell P} = \widetilde{R}$. By homogeneity, this equation constrains $\widetilde{R+P}$ from $\widetilde{R}, \widetilde{P}$ completely, up to the action by $\mu_\ell \subset \mathbb{G}_m$.

Now the theta group action of g_P on $u = \prod u_i$ is induced by the value $\prod u_i(\widetilde{P+R})$; since we take an ℓ-fold product the action of μ_ℓ above is killed, hence this product is well defined. More precisely, if X_0 has for divisor D, then $g_P \in G(\ell D)$ can be described by the formula $g_P(R) = X_0(\widetilde{R+P})^m / X_0(\widetilde{R})^m$.

6.2.2. *Canonical lifts of ℓ-torsion.* As a warm up, we first look at the theta group arithmetic on $G(D)$ encoded by cubical points $\widetilde{P}$ for $P \in A[D]$. Then we will extend this to the theta group arithmetic for $G(\ell D)$ encoded by $\widetilde{P}^{\otimes \ell}$ for $P \in A[\ell]$.

In this section, we freely use the notations $\delta_n, e_\star, \dots$ from [Mum66]. Let's assume that we are on level n for simplicity, so P is a point of n-torsion, and we assume further that D is symmetric, and if n is even that (the line bundle associated to) D is totally symmetric. In particular, $e_{D,\star}(T) = 1$ for all $T \in A[2]$, so the symmetric elements $g_T \in G(D)$ above T are of order 2. Furthermore, over $\overline{k}$, the theta group $G(D)$ is isomorphic to the Heisenberg group $\overline{k}^* \times (\mathbb{Z}/n\mathbb{Z})^g \times \mu_n^g$. On the Heisenberg group, we have that $\delta_n(\alpha, x, \zeta) = (\alpha^{n^2}, nx, \zeta^n)$, and that $(\pm 1, x, \zeta)$ are the two symmetric elements above (x, ζ).

One need to be careful that there are two different laws for the cubical arithmetic when $P \in A[D]$: the first one comes the action by $G(D)$. Iterating this action m times we get the cubical point $\widetilde{P} \cdot \widetilde{P} \cdot \dots \widetilde{P}$. If $\widetilde{P}$ corresponds to $g_P \in G(D)$, this laws corresponds, by definition of the cubical action, to the theta group law g_P^m. The second law comes from the cubical exponentiation: $m\widetilde{P}$. One can check (the easiest way is to use the analytic interpretation of the cubical law through the canonical factor of automorphy, see [LR22a]) that $m\widetilde{P}$ corresponds to $\delta_m g_P$. (A way to reformulate this is as follows: in the cubical ladder, if $\widetilde{P}$ corresponds to $g_P \cdot \tilde{0}$, then $-\widetilde{P}$ corresponds to $\delta_{-1} g_P \cdot \tilde{0}$. That's why the ladder will give $n\widetilde{P} = \delta_n g_P \cdot \tilde{0}$. But the cubical arithmetic is also compatible with the group action by Lemma 4.12. So in the ladder if we would use $\widetilde{P}^{-1} := g_P^{-1} \cdot \tilde{0}$ instead of $-\widetilde{P}$, we would obtain $\widetilde{P}^n = g_P^n \cdot \tilde{0}$.)

In particular, $n\widetilde{P} = \tilde{0}$ corresponds to $\delta_n g_P = 1$ which is equivalent to g_P being of n^2-torsion. This is why the equation $n\widetilde{P} = \tilde{0}$ has n^2-solutions over $\overline{k}$.

If $n = 2n' + 1$ is odd, we can check that g_P is of n-torsion is equivalent to $\delta_{n'+1} g_P = \delta_{-n'} g_P$. This corresponds to the equation $(n'+1)\widetilde{P} = -n'\widetilde{P}$, which indeed has n solutions over $\overline{k}$. If $n = 2n' + 2$ is even, we have that g_P is of $2n$-torsion is equivalent to $\delta_{n'+2} g_P = \delta_{-n'} g_P$, which corresponds to the equation $(n'+2)\widetilde{P} = n'\widetilde{P}$ and has $2n$ solutions over $\overline{k}$.

Now let's consider a point $P \in A[\ell]$, and take a cubical point $\widetilde{P}$. By Section 4.2.5, we have if $\ell\widetilde{P} = \lambda_0 \tilde{0}$, $(\ell+1)\widetilde{P} = \lambda_0 \lambda_1' \widetilde{P}$, with $\lambda_0^2 = {\lambda_1'}^\ell$.

Let us first assume that $\ell = 2\ell' + 1$ is odd. Then there is a unique β such that $\beta^\ell = \lambda_0$, $\beta^2 = \lambda_1'$, β is rational since λ_0, λ_1' are rational, and we have $(u\ell + v)\widetilde{P} = \beta^{\ell u^2 + 2v} v \widetilde{P}$. Furthermore, it is easy to check that replacing $\widetilde{P}$ by $\gamma\widetilde{P}$ changes β by $\beta\gamma^\ell$, hence λ_0 by $\lambda_0 \gamma^{\ell^2}$ and λ_1' by $\lambda_1' \gamma^{2\ell}$. Also, by the monodromy interpretation of the Tate pairing, $\lambda_1' = \beta^2$ is (a representative of the class of) the non reduced Tate pairing $e_{T,\ell}(P, P)$.

From this, we see that the equation $\ell\widetilde{P} = \tilde{0}$ (which has ℓ^2 solutions over $\overline{k}$) imposes $\beta^\ell = 1$, which is not enough to guarantee that $(\ell+1)\widetilde{P} = \widetilde{P}$. However, both equations $\ell\widetilde{P} = \tilde{0}$, $(\ell+1)\widetilde{P} = \widetilde{P}$ are enough to impose $\lambda_0 = \lambda_1' = 1$, which imposes $\beta = 1$ in the odd case, and $\beta^2 = 1$ in the even case. If these equations are satisfied, we say that $\widetilde{P}$ is a canonical cubical point of ℓ-torsion; we will also say it is a canonical lift of ℓ-torsion (beware that it is not unique: if $\widetilde{P}$ is a cubical point of ℓ-torsion, so is $\zeta\widetilde{P}$ for $\zeta \in \mu_\ell$. Since $\lambda_0 = \lambda_1'$ in that case, $m\widetilde{P}$ is also a canonical cubical point of ℓ-torsion). Using the formula for $(u\ell + v)\widetilde{P}$ above, we see that that these two equations can be rewritten as a single equation $(\ell'+1)\widetilde{P} = -\ell'\widetilde{P}$. Finding a canonical point of ℓ-torsion thus corresponds to solving the equation $x^\ell = \beta$, and since $\beta^2 = e_{T,\ell}(P, P)$ and ℓ is odd, we see that we have an obstruction to finding a rational solution given by the self Tate pairing. Note however that even if a canonical point $\widetilde{P}$ is given over an extension, $\widetilde{P}^{\otimes \ell}$ is rational since β is rational; and one can check (using the analytic

formulas of [LR22a]) that this point corresponds exactly to the unique element $g_P \in G(\ell D)$ which is symmetric of order ℓ (this is coherent with Example 2.7). By the discussion above, the other elements g_P of order ℓ are given by the $\widetilde{P}^{\otimes \ell}$ where $\widetilde{P}$ satisfies the relaxed equation $(\ell'+1)\widetilde{P} = \zeta - \ell'\widetilde{P}$ for some $\zeta \in \mu_\ell$.

Now assume that $\ell = 2m$ is even. The situation is more tricky. First, consider $\ell = 2$ and $P \in A[2]$; then $\lambda_0^2 = {\lambda'_1}^2$, and we have two cases: $\lambda_0 = \lambda'_1$ or $\lambda_0 = -\lambda'_1$. One can check that the sign is given by $e_{D,\star}(P) = (-1)^{m_D(P)-m_D(0)}$ where $m_D(P)$ is the multiplicity of D at P (so for an elliptic curve and $D = (0_E)$, $e_{D,\star}(P) = 1$ for all $P \in E[2]$). In the later case, when $e_{D,\star}(P) = -1$, there are no canonical cubical points of 2-torsion above P. But when $e_{D,\star}(P) = 1$, the equation $2\widetilde{P} = \tilde{0}$ has four solutions (over $\bar{k}$), and is enough for $\widetilde{P}$ to be a canonical point of 2-torsion.

Now for the general even case $\ell = 2m = 2\ell' + 2$, $P \in A[\ell]$. Let $P_0 = mP$. We have $\lambda_0^2 = {\lambda'_1}^\ell$, so ${\lambda'_1}^m = \pm\lambda_0$. But $\lambda_0(P_0) = \lambda_0(P)$, $\lambda'_1(P_0) = \lambda'_1(P)^m$, so ${\lambda'_1}^m = e_{D,\star}(P_0)\lambda_0$. If $e_{D,\star}(P_0) = -1$, there are no canonical points of ℓ-torsion. If $e_{D,\star}(P_0) = 1$, there are 2ℓ canonical points of ℓ-torsion, which can be found by solving the two equations $\ell\widetilde{P} = \tilde{0}$, $(\ell+1)\widetilde{P} = \widetilde{P}$, or the single equation $(\ell'+2)\widetilde{P} = -\ell'\widetilde{P}$. The solutions are parametrized by $\gamma^{2\ell} = \lambda'_1$. These 2ℓ solutions for $\mathcal{L}$ induces 2 distinct elements $\widetilde{P}^{\otimes\ell}$, which correspond to the two symmetric elements above P (necessarily of order ℓ) in $G(\ell D)$. These two elements are rational if and only if $\lambda'_1 = e_{T,\ell}(P,P)$ is a square; this is coherent with Example 2.7.

6.2.3. *Radical isogenies.* We remark that the cubical arithmetic also extends to isogenies. Let $f : A \to B$ be an isogeny of abelian varieties, $\mathcal{M}$ a line bundle on B and $\mathcal{L}$ a line bundle on A such that f is a ℓ-isogeny, i.e, $f^*\mathcal{M} \simeq \mathcal{L}^\ell$. We fix such an isomorphism once and for all. We remark that a rigidification $\widetilde{P}$ of $f^*\mathcal{M}$ (hence via the isomorphism above, a rigidification of $\mathcal{L}^\ell$) at a point $P \in A$ corresponds to a rigidification of $\mathcal{M}$ at the point $f(P)$, which we denote by $f(\widetilde{P})$ (or rather $\tilde{f}(\widetilde{P})$ where $\tilde{f}$ depends on the choice of isomorphism $f^*\mathcal{M} \simeq \mathcal{L}^\ell$). It is convenient to normalise the neutral points so that $\tilde{f}(\tilde{0}_A) = \tilde{0}_B$.

Cubical points thus also give the correct conceptual framework to build algorithms for radical isogenies. It is known (see [CDV20; CDHV22] for dimension 1 and [CD21; LR22b] for the general case) that radical isogenies can be described by choices of ℓ-th root of unity on Tate pairings.

Let us focus to the case of radical elliptic curve isogenies for simplicity. We start with $P \in E$ a point of ℓ-torsion, and the goal is to find formulas not only for the isogeny $\varphi : E \to E' = E/\langle P\rangle$ but also to find a new point $P' \in E'$ of ℓ-torsion such that the associated isogeny is not backtracking. It is not hard to see that such choices of P' are in bijection with $\widehat{\varphi}^{-1}(P)$, and the theory of radical isogenies state that this fiber is in bijection with choices of ℓ-th root of the non reduced Tate self pairing $e_{T,\ell}(P,P)$. We refer to [LR22b] for a geometric description of this isomorphism. Since cubical point arithmetic can be used to compute the Tate pairing, it is not surprising that they can explain radical isogenies, or more generally fibers of isogenies.

In brief, let $f : E_1 \to E_2$ be a cyclic isogeny of degree ℓ, and $\tilde{f}$ the contragredient isogeny. Fixing a point $P \in \operatorname{Ker}\tilde{f}$ gives through the Weil-Cartier pairing e_f an isomorphism $\operatorname{Ker} f \simeq \mu_\ell$, and if $Q \in E_2(Q)$, the (non reduced) Tate pairing $e_{T,f}(P,Q) = e_{T,\ell}(P,Q)$, seen as a μ_ℓ-étale torsor $\zeta^\ell = e_{T,\ell}(P,Q)$, is isomorphic to the fiber $f^{-1}(Q)$ which is a $\operatorname{Ker} f$-torsor. (In other words, the Galoisian structure of $f^{-1}(Q)$ can be derived from the Galoisian structure of $\zeta^\ell = e_{T,\ell}(P,Q)$.) For an ℓ-isogeny $f : A_1 \to A_2$ of principally polarised abelian varieties of dimension g, fixing a basis $P_1, \dots, P_g$ of $\operatorname{Ker}\tilde{f}$ splits $\operatorname{Ker} f$ as a product μ_ℓ^g, hence splits the

torsor $f^{-1}(Q)$ as a product of μ_ℓ-torsors, whose isomorphism classes are given by the torsors $\zeta^\ell = e_{T,\ell}(P_i, Q)$.

Let us go back to an ℓ-isogeny $f : E_1 \to E_2$ of elliptic curves for simplicity, and assume ℓ-odd. The symmetric divisor $\ell(0_{E_1})$ on E_1 descends (up to linear equivalence) to the symmetric divisor 0_{E_2} on E_2, so by Mumford's theory corresponds to a lift $\widetilde{K}$ of $\mathrm{Ker}\, f$ to the theta group $G(\ell(0_{E_1})$ (in practice: this is the unique lift given by symmetric elements of order ℓ). Now let $P \in \mathrm{Ker}\, \tilde{f}$ and $P' \in E_1[\ell]$ be a preimage of P by f. By the discussion above, a choice of cubical point $\widetilde{P}$ (of level 1) for P gives a choice of cubical point $\widetilde{P'}$ (of level ℓ) for P', but since P' is in $E_1[\ell]$, the local rigidification of the line bundle associated to $\ell(0_E)$ at P' gives a global rigidification, hence an element $g_{P'} \in G(\ell(0_{E_1}))$, and conversely. So in the other direction, since we have the canonical level subgroup $\widetilde{K} \subset G(\ell(0_{E_1}))$ above $K = \mathrm{Ker}\, f$, we have a canonical cubical point of level ℓ, for P', hence a canonical cubical point $\widetilde{P}$ of level 1 for P. One can check that this $\widetilde{P}$ is symmetric and of order ℓ for the cubical arithmetic (in the terminology above it is a canonical cubical point above $\widetilde{P}$), and that replacing P' by $P' + T$ with $T \in \mathrm{Ker}\, f$ replace $\widetilde{P}$ by $\zeta\widetilde{P}$ for $\zeta = e_f(T, P)$. In particular, the ℓ canonical cubical points $\widetilde{P}$ above P all come from a choice of $P' \in f^{-1}(P)$.

On the other hand, to compute a canonical cubical point $\widetilde{P}$ above P, one can start with an arbitrary cubical point $\widetilde{P}$, write $\ell = 2\ell' + 1$, and compute the monodromy λ_P between $(\ell' + 1)\widetilde{P}$ and $-\ell'\widetilde{P}$. By Remark 3.13 and Section 4.2.5, this monodromy λ_P is exactly the square root of the non reduced Tate pairing $e_{T,\ell}(P, P)$. Canonical cubical points are then of the form $\zeta\widetilde{P}$, where $\zeta^{-\ell} = \lambda_P$. Since we have seen that we also had a bijection with the preimages of P, this gives an explicit bijection between $\zeta^{-\ell} = \lambda_P = e_{T,\ell}(P, P)^{1/2}$ and $f^{-1}(P)$.

To study the general fiber $f^{-1}(Q)$ for some point $Q \in E_2(k)$, we can proceed similarly. Fix a Q' in the preimage, and an arbitrary cubical point of level 1 for Q (hence of level ℓ for Q'). We have seen that each preimage of P' of P gives a level ℓ theta group element $g_{P'}$, hence an action $g_{P'} \cdot \widetilde{Q'}$, hence by descent a cubical point $\widetilde{P + Q} = g_{P'} \cdot \widetilde{Q}$ above $P + Q$. One can check that $\widetilde{P + Q}$ satisfy the equation $\ell\widetilde{P + Q} = \widetilde{Q}$, and that changing P' by $P' + T$ changes $\widetilde{P + Q}$ to $e_f(T, P)\widetilde{P + Q}$. Hence we get all ℓ-cubical points $\widetilde{P + Q}$ satisfying this equation.

On the other hand, we can also compute these cubical points on E_2 directly: take an arbitrary $\widetilde{P + Q}$, an arbitrary canonical cubical point $\widetilde{P}$ so that $\ell\widetilde{P} = \tilde{0}$, and compute the monodromy $\ell\widetilde{P + Q} = \lambda_P\widetilde{Q}$. By the monodromy interpretation of the Tate pairing, we have $\lambda_P = e_{T,\ell}(P, Q)$. The points $\widetilde{P + Q}$ we search for are the ones given by $\zeta\widetilde{P + Q}$ where $\zeta^{-\ell} = \lambda_P$. Hence in this case, we have an explicit bijection between couples $(\widetilde{Q}, \widetilde{Q + P})$ satisfying $\widetilde{Q + \ell P} = \widetilde{Q}$; roots of the equation $\zeta^{-\ell} = \lambda_P = e_{T,\ell}(P, Q)$; and couples $(Q', Q' + P' + T)$ where $f(Q') = Q, f(P') = P$ are fixed and T goes through $\mathrm{Ker}\, f$.

The same strategy works for higher dimension. In practice, this allows to obtain explicit formulas[3] for fibers of isogenies between abelian varieties in the theta model, and also to obtain explicit multiradical isogeny formulas in the theta model. The formulas can be derived from the study of fibers of modular correspondances of [FLR11; Rob21a, § 5.2.2], then using [LR10; LR15] to reinterpret the constants appearing in these formulas as suitable Tate pairings, and then (a not yet published) adaptation of the descent level formula from [LR22a] to go down in level without multiplying by ℓ.

[3]These formulas, resulting from joint work with David Lubicz, were already promised more than one year ago in [Rob23c, Remark 5.19], but see footnote [2]…

6.3. **Supersingularity testing as a self pairing test.** Let $E/\mathbb{F}_q$ be an elliptic curve (with j-invariant different from $0, 1728$), $q = p^2$. We would like to test if it is supersingular. In [Dol18], Doliskani proposes the following probabilistic supersingularity test.

Recall that on an elliptic curve with a Weierstrass equation, there are canonical polynomials ϕ_n, ω_n, ψ_n (depending on E) such that if $P = (x_P, y_P)$, $nP = (\frac{\phi_n(x_P)}{\psi_n^2(x_P)}, \frac{\omega_n(x_P,y_P)}{\psi_n^3(x_P,y_P)})$. The polynomial ψ_n is the n-th division polynomial: $\psi_n(x_P, y_P) = 0$ if and only if P is a non trivial point of n-torsion. (If n is odd, ψ_n depends only on x).

Furthermore, there is a linear recurrence relation for these polynomials, which allows to compute $\psi_n(x_P, y_P)$ efficiently.

If the characteristic is p, then $\psi_p(x) = \widetilde{\psi}_p(x)^p$, where $\widetilde{\psi}_p(x)$ is a polynomial of degree $(p-1)/2$ if E is ordinary, or ± 1 if E is supersingular. In particular, E is supersingular if and only if $\psi_p^2 = 1$ [Dol18, Lemma 4; BGS22, Lemma 3].

Doliskani's supersingularity test is then to sample a random $x \in \mathbb{F}_q$, and test if $\psi_p(x)^2 = 1$. If it is not the case, we know that E is ordinary. Otherwise, by the Schwartz-Zippel lemma [Dol18, Lemma 2], we know that E is supersingular with high probability [Dol18, § 3]. In [Dol18, § 4] this test is further refined.

We now reinterpret this test as a pairing test. First, the division polynomials ψ_n give precisely the elliptic nets of rank 1 (i.e., elliptic divisibility sequences) [Sta08, Theorem 1.2.1], and elliptic nets give pairings by [Sta08, § 17, § 18]. Here we will rather use the cubical point of view (this is essentially the same thing by Section 4.9.5).

Analytically we have $\Psi_n(z) = \frac{\sigma(nz)}{\sigma(z)^n}$. It follows from Sections 4.8 and 4.9 that if $\widetilde{P} = (x(P), y(P), 1)$ is a level 3 cubical point given by level 3 affine Weierstrass coordinates, then $n\widetilde{P} = ((\phi_n(x_P)\psi_n(x_P, y_P), \omega_n(x_P, y_P), \psi_n^3(x_P, y_P))$. Likewise, if $\widetilde{P}$ is a level 1 cubical point normalised by $Z_1(\widetilde{P}) = 1$, then $Z_1(n\widetilde{P}) = \psi_n(x_P, y_P)$. (As an aside, by Remark 4.24, using division polynomial for cubical arithmetic means that $\tilde{0}$ is normalised with respect to $-x/y$ rather than with respect to x/y).

So Doliskani's supersingularity test can be reinterpreted as follows: given $\widetilde{P}$ normalised as above, is $Z_2(p\widetilde{P}) := Z_1^2(p\widetilde{P}) = 1$?

Now if E is supersingular over $\mathbb{F}_q$ (with maximal endomorphism ring), then $E(\mathbb{F}_q) = (\mathbb{Z}/(p \pm 1)\mathbb{Z})^2$. We first sample a random point $x(P)$ on the Kummer line; P lies either in E or its quadratic twist, so P is of order $p+1$ or $p-1$. In both cases, $x(pP) = x(P)$, this gives us a first test for a point order.

We can refine this test as follows: take $\widetilde{P}$ a cubical point of level 1 normalised by $Z_1(\widetilde{P}) = 1$, and of order $p \pm 1$, which we have checked via the equation $x(pP) = x(P)$. We want to compute the self Tate pairing $e_{p\pm1}(P, P)$. By Theorem 4.19 and Remark 3.13, if P is of order $p-1$, we can compute $e_p(P, P)$ by comparing $Z_1(p\widetilde{P})$ and $Z_1(\widetilde{P}) = 1$. In particular, $Z_1(p\widetilde{P}) = 1$ (i.e. $\psi_p(P) = 1$) if and only if $e_{p-1}(P, P) = 1$. And if P is of order $p+1$, we can compute $e_p(P, P)$ by comparing $Z_1(p\widetilde{P})$ and $Z_1(-\widetilde{P}) = -1$. In particular, $Z_1(p\widetilde{P}) = -1$ (i.e. $\psi_p(P) = -1$) if and only if $e_{p+1}(P, P) = 1$. (We remark that if $\lambda \in \mathbb{F}_q = \mathbb{F}_{p^2}$, $p(\lambda \cdot \widetilde{P}) = \lambda^{p^2} \cdot p\widetilde{P} = p\widetilde{P}$ so the Tate pairing is already reduced here.) So we can reframe Doliskani's supersingularity test as a self pairing test: is $e_{p\pm1}(P, P) = 1$ for a randomly sampled P?

Lemma 6.1. *Let $P \in E(\mathbb{F}_q)$ be of order $p \pm 1$. Then the reduced Tate self pairing $e_{T,p\pm1}(P, P) = 1$ if and only if the elliptic curve quotient $E' = E/\langle P \rangle$ has its full $p \pm 1$-torsion rational over $\mathbb{F}_q$.*

Proof. This follows from the geometric interpretation of the Tate pairing, see [Rob23c, Example 5.14]. □

In particular, if E is supersingular, then since $E/\langle P\rangle$ has the same Galois structure than E we always have $e_{T,p\pm1}(P) = 1$: self pairings are trivial on a supersingular curve. (This can also be seen more directly as follows: if P is of order ℓ, $e_{T,\ell}(P) = e_{W,\ell}(P, \pi P' - P')$ where $\ell P' = P$, but π is a scalar for a supersingular curve and the Weil pairing is alternate, hence the self pairing is trivial.) Lemma 6.1 thus gives an alternative proof that $\psi_p(x(P)) = \pm 1$ for a supersingular curve.

We can also use Lemma 6.1 and our reinterpreration of Doliskani's supersingularity test as a self pairing test to give a precise description of the points $P \in E(\mathbb{F}_q)$ such that Doliskani's test fails for an ordinary elliptic curve E: P has to be of $p \pm 1$-torsion and the isogeneous curve $E/\langle P\rangle$ has to have fully rational $p \pm 1$-torsion. Using the group structure of isogeny volcanoes of ordinary curves, this allows to refine the probability of failure (depending on where E is in the volcano).

In practice, in isogeny based cryptography we work with the Montgomery model. We can thus use the fast cubical ladder formulas from Section 5.2 to do our self pairing test: sample a random point $x(P)$, start with $\widetilde{P}$ such that $X_2(\widetilde{P}) = x(P)$, $Z_2(\widetilde{P}) = 1$ in level 2 affine cubical coordinates, and compute whether $p\widetilde{P} = \widetilde{P}$. This test both that P is of $p \pm 1$ torsion and that the self pairing is trivial (or rather its square, since we are using level 2 coordinates).

We recover precisely the fast supersingular test from [BGS22]. In that paper, the authors apply the above strategy, but using the usual projective Montgomery ladder to compute $p\widetilde{P}$. As we have seen in Section 5.2, the standard projective ladder almost correctly computes the cubical arithmetic, and it is easy to keep track of a correcting factor to apply afterwards to obtain the correct cubical point $p\widetilde{P}$. Now comparing Algorithms 5.4 and 5.5 with [BGS22, Algorithm 1], we can check that this correcting factor is precisely the factor from [BGS22, Proposition 2]. Since the authors of [BGS22] use this correcting factor in their fast supersingularity test [BGS22, Algorithm 6], they are really computing the true cubical exponentiation [BGS22]! (We remark that computing $1/x(P)$ and then directly using the formulas Algorithms 5.4 and 5.5 for the cubical ladder would save $1M$ by bit.)

6.4. **Monodromy leak and the DLP.** Our last non pairing application, and perhaps the most important one, is a new devastating side-channel attack against the Montgomery ladder on Montgomery curves, which we call the monodromy leak, a type of projective coordinate leak.

Let $P \in E(\mathbb{F}_q)$ be a point of ℓ-torsion, with ℓ prime for simplicity. It is known since [NSS04] that a projective coordinate leak, that is leaking the individual projective coordinates $X(mP), Y(mP), Z(mP)$ which are computed during the scalar multiplication, where $mP = (X(mP) : Y(mP) : Z(mP))$, would yield information on the secret m. The attack of [NSS04] can only recover a few bits of m, so the attack was only used for attacking the signature scheme ECDSA. For ECDSA, obtaining a few bit of leakage for each signature, combined with lattices methods, allows to recover the full secret key. This attack was revisited in [AGB20] where the authors found that many implementation were still vulnerable to projective coordinate leaks, and extended it to the Montgomery ladder.

By contrast, our monodromy leak attack on the Montgomery ladder is much more devastating, since it allows to recover the full key via only one leak, by reduction to some DLPs in $\mathbb{F}_q^*$.

The main idea is as follows: first we can assume that $\mu_\ell \not\subset \mathbb{F}_q$, otherwise pairings already give an efficient reduction from DLPs to $E(\mathbb{F}_q)$ to DLPs in $\mathbb{F}_q^*$. We can also assume ℓ odd for simplicity. Under this hypothesis, for a point $P \in E[\ell](\mathbb{F}_q)$ there is only one rational cubical point $\widetilde{P}$ which is still of ℓ-torsion. We will call $\widetilde{P}$ the canonical lift of P, and denote it also by $\hat{P}$. We can efficiently compute $\hat{P}$ by taking an arbitrary rational cubical lift $\widetilde{P}$ and then computing $\hat{P} = u\widetilde{P}$ where $u \equiv 1 \mod \ell$ and $u \equiv 0 \mod q-1$; such a u exists because ℓ is prime to $q-1$ by assumption.

Lemma 6.2. *If $\mu_\ell(\mathbb{F}_q) = 1$ (i.e. ℓ is coprime to $q-1$), there is a unique rational canonical cubical point of ℓ-torsion $\hat{P}$. It suffices to check that $\ell\hat{P} = \tilde{0}$, and for any rational cubical point $\widetilde{P}$ above P, $\hat{P} = u\widetilde{P}$, where $u \equiv 1 \mod \ell$ and $u \equiv 0 \mod q-1$.*

Furthermore, for any m, we have $m\hat{P} = \widehat{mP}$

Proof. We defined canonical cubical points of ℓ-torsion in Section 6.2; they satisfy $\ell\widetilde{P} = \tilde{0}$, $(\ell+1)\widetilde{P} = \widetilde{P}$. Using the notations of Section 6.2.2, if $\ell\widetilde{P} = \tilde{0}$ then $\lambda_0 = \beta^\ell = 1$, hence $\beta = 1$ since β is rational and $\mu_\ell(\mathbb{F}_q) = 1$, so $\lambda_1' = 1$, and in this case the equation $\ell\widetilde{P}$ is sufficient to define a canonical point of ℓ-torsion.

For a random cubical point $\widetilde{P}$, if $\ell\widetilde{P} = \lambda_0\tilde{0}$, then $\gamma\widetilde{P}$ is a canonical point of ℓ-torsion for $\gamma^\ell = \lambda_0^{-1}$. Since $\mu_\ell(\mathbb{F}_q) = 1$, $x \mapsto x^\ell$ is bijective in $\mathbb{F}_q^*$, so this equation has a unique solution in $\mathbb{F}_q^*$.

We have $u\hat{P} = \hat{P}$ because $u \equiv 1 \mod \ell$ and $\hat{P}$ is a canonical cubical point of ℓ-torsion. On the other hand, starting with an arbitrary rational cubical point $\widetilde{P} = \lambda\hat{P}, \lambda \in \mathbb{F}_q$, we have $u\widetilde{P} = \lambda^{u^2}u\hat{P}$, and since $u \equiv 0 \mod q-1$, $\lambda^{u^2} = 1$. So $u\widetilde{P} = \hat{P}$.

If $Q = mP$, taking an arbitrary rational $\widetilde{P}$ and letting $\widetilde{Q} = m\widetilde{P}$, we obtain $\widehat{Q} = u\widetilde{Q} = um\widetilde{P} = mu\widetilde{P} = m\hat{P}$. □

(We remark that [Stao8, Chapter 19; LSo8] also compute an elliptic net representation of the level 1 canonical lift $\hat{P}$ for elliptic curves. Their formula to compute $\hat{P}$ is slightly different than the one above, but do give the same canonical lift.)

The relation $m\hat{P} = \widehat{mP}$ can also be proved by unicity and the compatibility relations of the cubical arithmetic: in general, if $\widetilde{P}$ is a canonical lift of ℓ-torsion, so is $m\widetilde{P}$ for any m. In our situation, the canonical lift is unique (if we want it to be rational), so by Corollary 4.33, we even have that if $[P_1, P_2, P_3, P_4; Q_1, Q_2, Q_3, Q_4]$ are points of ℓ-torsion in Riemann position, so are their canonical lift of ℓ-torsion.

Now the cubical arithmetic is a mix of elliptic curve arithmetic and $\mathbb{F}_q^*$ arithmetic, and we can use $\hat{P}$ to try to reduce to DLPs in $\mathbb{F}_q^*$. Namely, for the Montgomery ladder, one start with a normalised point $P = (x(P) : 1)$ in order to reduce the number of multiplications. Let us define $\widetilde{P} = (x(P), 1)$, and assume we obtain a projective coordinate leak of $m.P = (X(mP), Z(mP))$. Now assume furthermore that $m.P$ was computed using the cubical formulas, in other words $X(mP), Z(mP)$ gives $m\widetilde{P}$ exactly. Then we have $\widetilde{P} = \lambda\hat{P}$ with $\lambda = 1/Z(\hat{P})$, and $m\widetilde{P} = \lambda^{m^2}m\hat{P}$ by homogeneity.

Now we know $mP = (X(mP) : Z(mP))$ because it is public, we know $\widetilde{P} = (x(P), 1)$ because P is normalised, we also know $m\widetilde{P} = (X(mP), Z(mP))$ because we have assumed we had a projective coordinate leak. We know the canonical lift $\hat{P}$, and we can also compute $m\hat{P}$ even if we don't know m, because the canonical lift $\widehat{mP}$ of mP is precisely equal to $m\hat{P}$.

So we obtain an equation of the form $\lambda^{m^2} = Z(\widetilde{mP})/Z(\widehat{mP})$, where the only unknown is m. Solving a DLP in $\mathbb{F}_q^*$ followed by a square root allows to recover the possible values of m modulo $q-1$, or more precisely modulo the order of λ.

In practice, we know that in the Montgomery ladder implementation, m is as small as possible, so in particular is smaller than $\#E(\mathbb{F}_q) \leq q+1+2\sqrt{q}$. We first solve the DLP in $\mathbb{F}_q^*$ to recover m^2 modulo the order N of λ (so that $N \mid q-1$). We then test all possible square roots, there are at most 2^t such square roots where t is the number of distinct prime factors of $q-1$. Unless $q-1$ is very smooth, computing all square roots is not a bottleneck compared to the DLP in $\mathbb{F}_q^*$.

Now, for each possible square root m modulo N, we need to test all possibilities $m+aN$ up to the upper bound $m \leq \#E(\mathbb{F}_q)$. The number of possibilities is in $O(q/N)$, the smaller the order of λ the more tries we need, but on the other hand the lower the probability to stumble on such a λ. By Merten's theorem, the average number of tries for a random λ is in $O(\log q)$.

Note that the above method applies even if $\widetilde{P}$ is not normalised, as long as we know both $\widetilde{P}$ (say because the implementation is public) and $m\widetilde{P}$ (because of a projective coordinate leak).

However, this only applies because we know that $m \leq \ell$. If we don't know the projective coordinate leak $m\widetilde{P}$, we can still take an arbitrary cubical point $\widetilde{mP}$ (well not quite arbitrary, we want to choose $Z(\widetilde{mP})$ such that $Z(\widetilde{mP})/Z(\widehat{mP})$ is in the group generated by λ). Then we have $\widetilde{mP} = M\widetilde{P}$ for some $M \equiv m \mod \ell$, but this time we only have the bound $M \leq \ell(q-1)$. The cubical arithmetic allows to recover the value of M modulo $q-1$ (or rather modulo N the order of λ), which gives zero information on the value of M modulo ℓ since ℓ is prime to $q-1$. In other words, the monodromy attack only applies if we know the number (or we can pin this number in a small interval) of loop around ℓ we did when computing $M.\widetilde{P}$.

Remark 6.3 (Biextension monodromy leak)**.** We can also do a monodromy attack using biextension arithmetic, this requires more information but bypass the square root step. Namely, still with our hypothesis that ℓ is prime to $q-1$, given another rational point Q, there is a unique rational biextension element $\hat{g}_{P,Q}$ above P of order ℓ for $\star_1$. We have $\hat{g}_{P,P}^{\star_1,m} = \hat{g}_{mP,P}$.

It follows that if through some projective coordinate leak we are able to recover both some biextension element $g_{P,P} = \lambda\hat{g}_{P,P}$ and $g_{P,P}^{\star_1,m} = \lambda^m\hat{g}_{mP,P}$, we can recover m modulo the order $N \mid q-1$ of λ by solving an equation $\lambda^m = C$.

In practice, this could happen if we have a projective coordinate leak of both $m\widetilde{P}$, $(m+1)\widetilde{P}$ (this is not unreasonable since the Montgomery ladder computes $mP, (m+1)P$).

Namely, we know that on input the ladder start with the normalised point $\widetilde{P} = (x(P), 1)$; we compute $2\widetilde{P}$ by a cubical doubling, and represent $g_{P,P}$ by the cubical points $[\widetilde{P}, \widetilde{P}; \tilde{0}, 2\widetilde{P}]$. Then $g_{P,P}^{\star_1,m}$ is represented by $[\widetilde{P}, m\widetilde{P}; \tilde{0}, (m+1)\widetilde{P}]$, so a projective leak of $m\widetilde{P}$, $(m+1)\widetilde{P}$ indeed gives us $g_{P,P}^{\star_1,m}$. (As an aside: $\hat{g}_{mP,P} = [\hat{P}, \hat{mP}; \tilde{0}, \hat{(m+1)}P]$.)

Now, all the discussion above assumes that the Montgomery ladder is implemented using the cubical arithmetic; which is definitively *not* the case. But we saw in Section 1.2 that it is *very close* to the cubical ladder.

More precisely, the projective doubling formula is exactly the same as in Algorithm 5.4. However, the projective differential addition formula uses the equation

$$\begin{aligned} U &= (X(P) - Z(P))(X(Q) + Z(Q)) \\ V &= (X(P) + Z(P))(X(Q) - Z(Q)) \\ X(P+Q) &= Z(P-Q)(U+V)^2 \\ Z(P+Q) &= X(P-Q)(U-V)^2 \end{aligned}$$

which differs from the cubical differential addition Algorithm 5.5 by the factor $4X(P-Q)Z(P-Q)$.

Taking this into account, we need to solve a slightly different degree two equations to recover m. Namely, let $[m\widetilde{P}]$ be the cubical point computed by the usual Montgomery ladder. Then by [BGS22, Proposition 2], $[m\widetilde{P}] = (4x(P))^{m2^{l(m)}-m}m\widetilde{P}$ where $l(m)$ is the binary length of m (see Section 6.3 for the link between the division polynomials in the statement of [BGS22, Proposition 2], and cubical arithmetic).

Using $\widetilde{P} = \lambda_1\hat{P}$, $Q = mP$, $m\widetilde{P} = \lambda_1^{m^2}\hat{mP}$, $[m\widetilde{P}] = (4x(P))^{m2^{l(m)}-m}m\widetilde{P}$, and since we know P, Q, hence $\hat{P}, \hat{Q}$, and also $\widetilde{P}$ and $[m\widetilde{P}]$ by assumption that we have a projective coordinate leak on the standard projective ladder, we can recover λ_2 such that $[m\widetilde{P}] = \lambda_2\hat{mP}$ and we have the following equation where the only unknown is m:

$$(4x(P))^{m2^{l(m)}-m}\lambda_1^{m^2} = \lambda_2 \tag{22}$$

In practice, we also know the length $l(m)$ of m. So we fix ζ a primitive root of $\mathbb{F}_q^*$, and compute the dlps with respect to ζ: $\mathrm{dlp}_\zeta(4x(P))$, $\mathrm{dlp}_\zeta(\lambda_1)$, $\mathrm{dlp}_\zeta(\lambda_2)$. Then by Equation (22), m is a solution of the degree two equation:

$$X^2(\mathrm{dlp}_\zeta(\lambda_1) - \mathrm{dlp}_\zeta(4x(P))) + 2^{l(m)}\,\mathrm{dlp}_\zeta(4x(P))X - \mathrm{dlp}_\zeta(\lambda_2) = 0.$$

We then proceed as before. The above discussion can thus be summarised in:

Theorem 6.4. *Let $P = (X(P), Z(P))$ be a known public point of order ℓ on a Montgomery Kummer line associated to a Montgomery curve $E/\mathbb{F}_q$ (here we assume that we know not only P, but $X(P), Z(P)$, in practice P is normalised via $Z(P) = 1$). Assume that ℓ is prime to $q-1$.*

Let $m \leq \ell$, and let $mP = (X(mP), Z(mP))$ as computed by the standard projective Montgomery ladder. Assume that we obtain a projective coordinate leak of mP, i.e., we not only know $x(mP) = X(mP)/Z(mP)$, but also $X(mP), Z(mP)$.

Let u be the number of distinct prime factors of $q-1$. Let $\hat{P} = (X(\hat{P}), Z(\hat{P}))$ be the unique canonical cubical rational point above P, $N \mid q-1$ be the multiplicative order of $Z(P)/Z(\hat{P})$ and $v = (q-1)/N$.

Then one can recover m by solving three discrete algorithms in $\mathbb{F}_q^*$ (two of which can be seen as a precomputation depending only on P, not m), followed by an algorithm polynomial in $\log q$, 2^u and v.

Any constant time division algorithm to compute $x(mP) = X(mP)/Z(mP)$, such as the one employed by NaCL, prevents this attacks. Compared to [NSS04], we can recover m fully from one leak, but we need to assume that we know $X(P), Z(P)$, not only P. So another protection is to mask P by multiplying $X(P), Z(P)$ by some random scalar factor before doing the exponentiation.

For more details on the monodromy leak (and some wild speculations), we refer to [Rob23a, § 7.2] and to [Rob23b] for a toy implementation.

7. Perspectives

We have seen that biextensions provide a convenient framework to study pairings as monodromy information, and that the efficient representation of biextension elements provided by cubical points gave fast formulas to compute pairings on elliptic curves.

In this paper, we have mainly looked in Section 5 at cubical points on Montgomery Kummer lines represented by level 2 affine coordinates. It would be interesting to look at other models and also explore further a mix of projective and affine coordinates (notably in the Edwards model), as suggested in Remark 4.34. There is also potentially room for optimisation of the adding formulas in the double and add biextension algorithm for the Montgomery model. We have also seen in Section 6.1 that the cubical arithmetic could potentially be interesting for pairing based cryptography, not only for generic pairing computations. In fact, we saw that biextensions and cubical arithmetic shed new light (and gave new formulas) on Miller's algorithm.

In the other direction, one could ask if the refinements of Miller's algorithm from [BELL10; LL11; DZZZ22] (see Section 6.1.2) could also be applied in some sense to the cubical ladder. Notably so far in our explicit cubical arithmetic, we have only used doublings, differential additions and three way additions, but not the more general algebraic Riemann relations from Section 4.1. Although the general Riemann relations are induced by the squared torsor structure, and explicit formulas can be given via the differential addition formulas (see Section 4.9.3 for an example), expressing the general Riemann relations through differential addition formulas involve some divisions by 2, hence potentially points in field extensions, while using the formulas from Section 4.1 we can stay in the base field. Notably, the work of [BELL10] suggest looking at the following Riemann relations to compute the cubical exponentiation: $[(m_1+m_2)P+Q, -m_1P+Q, -m_2P+Q, -Q; -(m_1+m_2)P, m_1P, m_2P, 2Q]$. Maybe some other types of Riemann relations could also be useful.

We have focused in this article on computing pairings through monodromy (except in the setting of Remark 2.11. Furthermore, we work with biextension elements represented via cubical points, and we represent cubical points fully (or almost fully in level 2) through their cubical coordinates. This helps making the pairing formula complete, see Remarks 4.26 and 5.3: we can switch coordinates on the fly according to which point monodromy we want to compute. On the other hand, this means that for pairings, we compute redundant information: by Lemma 4.15 the cubical representation of a biextension element is not unique, and by Sections 4.5.2 and 4.6 cubical points are overkill to compute pairings, we just need cubical functions (a cubical point, represented by its cubical coordinates, encode a bunch of cubical functions at once). We already exploited the non unicity of the biextension representation in the double and add algorithm from Algorithm 4.1: the cubical points computed via this algorithm are not compatible with the cubical arithmetic, but they still give the correct biextension arithmetic. All in all this means that we have various ways to represent biextension elements (the evaluation representation, the cubical representation) and various ways to represent cubical points or cubical functions themselves, which makes our framework quite flexible, and there could still be potential for optimisations.

An interesting direction would also be to look at the arithmetic of a general biextension X_f associated to an isogeny rather than just a polarisation, and see if it can help to compute the Weil-Cartier pairing e_f and the generalised Tate pairing. It also could be helpful in deriving isogeny formulas; see Section 6.2 for why the cubical arithmetic can help to understand isogenies and their fibers.

One can also wonder whether this point of view can help about pairing inversion (we saw in Section 6.4 that it gave new insight on the dlp). From the point of view of biextensions, the reduced Tate pairing inversion is just the computation of a $q-1$-th root in some theta group. But taking root of elements is not faster than a dlp for a generic group.

Going beyond pairings, we hope that Section 6 has convinced the reader that the notion of cubical points and cubical arithmetic is worthwhile to study, beyond just as a convenient way to work out the biextension arithmetic and computing the monodromy pairings. Indeed, some of our applications really need the full power of the cubical arithmetic. A cubical point is an amalgamation of an abelian variety point with some extra data coming from an action of $\mathbb{G}_m$. Exploiting this action carefully through monodromy computations gave us back the usual pairings. One can wonder whether there could be some cryptographic applications which uses the cubical arithmetic directly, not just through pairing computations.

References

[AGB20] A. C. Aldaya, C. P. García, and B. B. Brumley. "From A to Z: Projective coordinates leakage in the wild". In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* (2020), pp. 428–453 (cit. on p. 76).

[BGS22] G. Banegas, V. Gilchrist, and B. Smith. "Efficient supersingularity testing over GF (p) and CSIDH key validation". In: *Mathematical Cryptology* 2.1 (2022), pp. 21–35 (cit. on pp. 67, 75, 76, 79).

[BBLP13] D. Bernstein, P. Birkner, T. Lange, and C. Peters. "ECM using Edwards curves". In: *Mathematics of Computation* 82.282 (2013), pp. 1139–1179 (cit. on p. 53).

[BDLS20] D. Bernstein, L. De Feo, A. Leroux, and B. Smith. "Faster computation of isogenies of large prime degree". In: *Algorithmic Number Theory Symposium (ANTS XIV)*. Vol. 4. 1. Mathematical Sciences Publishers, 2020, pp. 39–55. arXiv: 2003.10118. URL: https://msp.org/obs/2020/4/p04.xhtml (cit. on p. 64).

[BBM79] P. Berthelot, L. Breen, and W. Messing. *Théorie de Dieudonné cristalline II*. Springer, 1979 (cit. on pp. 17, 18).

[BELL10] J. Boxall, N. El Mrabet, F. Laguillaumie, and D.-P. Le. "A variant of miller's formula and algorithm". In: *Pairing-Based Cryptography-Pairing 2010: 4th International Conference, Yamanaka Hot Spring, Japan, December 2010. Proceedings 4*. Springer. 2010, pp. 417–434 (cit. on pp. 1, 2, 9, 24, 67, 69, 70, 80).

[Bre83] L. Breen. *Fonctions thêta et théoreme du cube*. Vol. 980. Springer, 1983 (cit. on pp. 6, 9, 18, 19, 27, 35, 51, 54).

[BJ02] E. Brier and M. Joye. "Weierstraß elliptic curves and side-channel attacks". In: *International workshop on public key cryptography*. Springer. 2002, pp. 335–345 (cit. on p. 66).

[CLZ24] S. Cai, K. Lin, and C.-A. Zhao. "Pairing Optimizations for Isogeny-based Cryptosystems". In: *Cryptology ePrint Archive* (2024) (cit. on p. 9).

[CD21] W. Castryck and T. Decru. "Multiradical isogenies". In: *Cryptology ePrint Archive* (2021) (cit. on p. 73).

[CDHV22] W. Castryck, T. Decru, M. Houben, and F. Vercauteren. "Horizontal racewalking using radical isogenies". In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2022, pp. 67–96 (cit. on p. 73).

[CDV20] W. Castryck, T. Decru, and F. Vercauteren. "Radical isogenies". In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2020, pp. 493–519 (cit. on p. 73).

[CGF08] W. Castryck, S. Galbraith, and R. R. Farashahi. "Efficient arithmetic on elliptic curves using a mixed Edwards-Montgomery representation". In: *Cryptology ePrint Archive* (2008) (cit. on p. 3).

[CHM+23] W. Castryck, M. Houben, S.-P. Merz, M. Mula, S. v. Buuren, and F. Vercauteren. "Weak instances of class group action based cryptography via self-pairings". In: *Annual International Cryptology Conference*. Springer. 2023, pp. 762–792 (cit. on pp. 8, 9).

[Con04] B. Conrad. "Polarizations". 2004. URL: http://math.stanford.edu/~conrad/vigregroup/vigre04/polarization.pdf (cit. on pp. 8, 11).

[CR15] R. Cosset and D. Robert. "An algorithm for computing (ℓ, ℓ)-isogenies in polynomial time on Jacobians of hyperelliptic curves of genus 2". In: *Mathematics of Computation* 84.294 (Nov. 2015), pp. 1953–1975. DOI: 10.1090/S0025-5718-2014-02899-8. URL: http://www.normalesup.org/~robert/pro/publications/articles/niveau.pdf. HAL: hal-00578991, eprint: 2011/143. (Cit. on p. 71).

[CLN10] C. Costello, T. Lange, and M. Naehrig. "Faster pairing computations on curves with high-degree twists". In: *Public Key Cryptography–PKC 2010* (2010), pp. 224–242 (cit. on pp. 26, 27).

[DZZ23] Y. Dai, F. Zhang, and C.-a. Zhao. "Don't Forget Pairing-Friendly Curves with Odd Prime Embedding Degrees". In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2023.4 (2023), pp. 393–419 (cit. on p. 9).

[DZZZ22] Y. Dai, Z. Zhou, F. Zhang, and C.-A. Zhao. "Software implementation of optimal pairings on elliptic curves with odd prime embedding degrees". In: *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* 105.5 (2022), pp. 858–870 (cit. on pp. 9, 24, 69, 80).

[Dol18] J. Doliskani. "On division polynomial PIT and supersingularity". In: *Applicable Algebra in Engineering, Communication and Computing* 29.5 (2018), pp. 393–407 (cit. on pp. 67, 75).

[FLR11] J.-C. Faugère, D. Lubicz, and D. Robert. "Computing modular correspondences for abelian varieties". In: *Journal of Algebra* 343.1 (Oct. 2011), pp. 248–277. DOI: 10.1016/j.jalgebra.2011.06.031. arXiv: 0910.4668 [cs.SC]. URL: http://www.normalesup.org/~robert/pro/publications/articles/modular.pdf. HAL: hal-00426338. (Cit. on p. 74).

[FK22] E. V. Flynn and K. Khuri-Makdisi. "An analog of the Edwards model for Jacobians of genus 2 curves". In: *arXiv preprint arXiv:2211.01450* (2022) (cit. on p. 53).

[GL08] S. Galbraith and X. Lin. "Computing Pairings Using x-Coordinates Only". In: *Designs, Codes and Cryptography* (2008). to appear (cit. on pp. 3, 50).

[GJMRV11] R. R. Goundar, M. Joye, A. Miyaji, M. Rivain, and A. Venelli. "Scalar multiplication on Weierstraß elliptic curves from Co-Z arithmetic". In: *Journal of cryptographic engineering* 1 (2011), pp. 161–176 (cit. on p. 66).

[Gro72] A. Grothendieck. *Groupes de Monodromie en Géométrie Algébrique: SGA 7*. Springer-Verlag, 1972 (cit. on pp. 4, 17–20).

[Ham20] M. Hamburg. "Faster Montgomery and double-add ladders for short Weierstrass curves". In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* (2020), pp. 189–208 (cit. on p. 66).

[Koh11a] D. Kohel. "Addition law structure of elliptic curves". In: *Journal of Number Theory* 131 (2011), pp. 894–919 (cit. on p. 53).

[Koh11b] D. Kohel. "Arithmetic of split Kummer surfaces: Montgomery endomorphism of Edwards products". In: *Coding and Cryptology*. Springer, 2011, pp. 238–245 (cit. on p. 53).

[LS08] K. E. Lauter and K. E. Stange. "The elliptic curve discrete logarithm problem and equivalent hard problems for elliptic divisibility sequences". In: *International Workshop on Selected Areas in Cryptography*. Springer. 2008, pp. 309–327 (cit. on pp. 8, 77).

[LL11] D.-P. Le and C.-L. Liu. "Refinements of Miller's algorithm over Weierstrass curves revisited". In: *The Computer Journal* 54.10 (2011), pp. 1582–1591 (cit. on pp. 70, 80).

[LWXZ23] K. Lin, W. Wang, Z. Xu, and C.-A. Zhao. "A faster software implementation of sqisign". In: *Cryptology ePrint Archive* (2023) (cit. on p. 9).

[LR10] D. Lubicz and D. Robert. "Efficient pairing computation with theta functions". In: ed. by G. Hanrot, F. Morain, and E. Thomé. Vol. 6197. Lecture Notes in Comput. Sci. 9th International Symposium, Nancy, France, ANTS-IX, July 19-23, 2010, Proceedings. Springer–Verlag, July 2010. DOI: 10.1007/978-3-642-14518-6_21. URL: http://www.normalesup.org/~robert/pro/publications/articles/pairings.pdf. Slides: 2010-07-ANTS-Nancy.pdf (30min, International Algorithmic Number Theory Symposium (ANTS-IX), July 2010, Nancy), HAL: hal-00528944. (Cit. on pp. 6, 9, 51, 74).

[LR12] D. Lubicz and D. Robert. "Computing isogenies between abelian varieties". In: *Compositio Mathematica* 148.5 (Sept. 2012), pp. 1483–1515. DOI: 10.1112/S0010437X12000243. arXiv: 1001.2016 [math.AG]. URL: http://www.normalesup.org/~robert/pro/publications/articles/isogenies.pdf. HAL: hal-00446062. (Cit. on p. 71).

[LR15] D. Lubicz and D. Robert. "A generalisation of Miller's algorithm and applications to pairing computations on abelian varieties". In: *Journal of Symbolic Computation* 67 (Mar. 2015), pp. 68–92. DOI: 10.1016/j.jsc.2014.08.001. URL: http://www.normalesup.org/~robert/pro/publications/articles/optimal.pdf. HAL: hal-00806923, eprint: 2013/192. (Cit. on pp. 6, 9, 24, 50, 51, 74).

[LR16] D. Lubicz and D. Robert. "Arithmetic on Abelian and Kummer Varieties". In: *Finite Fields and Their Applications* 39 (May 2016), pp. 130–158. DOI: 10.1016/j.ffa.2016.01.009. URL: http://www.normalesup.org/~robert/pro/publications/articles/arithmetic.pdf. HAL: hal-01057467, eprint: 2014/493. (Cit. on pp. 3, 49, 51, 53, 66).

[LR22a] D. Lubicz and D. Robert. "Fast change of level and applications to isogenies". In: *Research in Number Theory (ANTS XV Conference)* 9.1 (Dec. 2022). DOI: 10.1007/s40993-022-00407-9. URL: http://www.normalesup.org/~robert/pro/publications/articles/change_level.pdf. HAL: hal-03738315. (Cit. on pp. 33, 71–74).

[LR22b] D. Lubicz and D. Robert. "Multiradical isogenies in the theta model". Sept. 2022. In preparation. (Cit. on p. 73).

[Melo7] N. Meloni. "New point addition formulae for ECC applications". In: *Arithmetic of Finite Fields: First International Workshop, WAIFI 2007, Madrid, Spain, June 21-22, 2007. Proceedings 1*. Springer. 2007, pp. 189–201 (cit. on p. 66).

[Mor85] L. Moret-Bailly. *Pinceaux de variétés abéliennes*. Société mathématique de France, 1985 (cit. on pp. 6, 9, 19, 27, 29, 31, 35, 52).

[Mum66] D. Mumford. "On the equations defining abelian varieties. I". In: *Invent. Math.* 1 (1966), pp. 287–354 (cit. on pp. 10, 12, 13, 29, 50, 52, 71, 72).

[Mum69] D. Mumford. "Bi-extensions of formal groups". In: *Algebraic geometry* 307-322 (1969) (cit. on pp. 4, 17).

[Mum70] D. Mumford. *Abelian varieties*. Tata Institute of Fundamental Research Studies in Mathematics, No. 5. Published for the Tata Institute of Fundamental Research, Bombay, 1970, pp. viii+242 (cit. on pp. 10, 11).

[Mum83] D. Mumford. *Tata lectures on theta I*. Vol. 28. Progress in Mathematics. With the assistance of C. Musili, M. Nori, E. Previato and M. Stillman. Boston, MA: Birkhäuser Boston Inc., 1983, pp. xiii+235. ISBN: 3-7643-3109-7 (cit. on p. 51).

[Mum84] D. Mumford. *Tata lectures on theta II*. Vol. 43. Progress in Mathematics. Jacobian theta functions and differential equations, With the collaboration of C. Musili, M. Nori, E. Previato, M. Stillman and H. Umemura. Boston, MA: Birkhäuser Boston Inc., 1984, pp. xiv+272. ISBN: 0-8176-3110-0 (cit. on p. 51).

[NSS04] D. Naccache, N. P. Smart, and J. Stern. "Projective coordinates leak". In: *Advances in Cryptology-EUROCRYPT 2004: International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004. Proceedings 23*. Springer. 2004, pp. 257–267 (cit. on pp. 76, 79).

[OKUO11] N. Ogura, N. Kanayama, S. Uchiyama, and E. Okamoto. "Cryptographic pairings based on elliptic nets". In: *Advances in Information and Computer Security: 6th International Workshop, IWSEC 2011, Tokyo, Japan, November 8-10, 2011. Proceedings 6*. Springer. 2011, pp. 65–78 (cit. on pp. 54, 56).

[Rei23] K. Reijnders. "Effective Pairings in Isogeny-Based Cryptography". In: *International Conference on Cryptology and Information Security in Latin America*. Springer. 2023, pp. 109–128 (cit. on pp. 1, 9).

[Rob21a] D. Robert. "Efficient algorithms for abelian varieties and their moduli spaces". HDR thesis. Université Bordeaux, June 2021. URL: http://www.normalesup.org/~robert/pro/publications/academic/hdr.pdf. Slides: 2021-06-HDR-Bordeaux.pdf (1h, Bordeaux). (Cit. on pp. 3, 14, 23, 50, 70, 71, 74).

[Rob21b] D. Robert. *General theory of abelian varieties and their moduli spaces*. Mar. 2021. URL: http://www.normalesup.org/~robert/pro/publications/books/avtheory.pdf. Draft version. (Cit. on pp. 14, 50).

[Rob23a] D. Robert. "Improving the arithmetic of Kummer lines". Aug. 2023. URL: http://www.normalesup.org/~robert/pro/publications/notes/2023-11-kummer_lines.pdf (cit. on p. 79).

[Rob23b] D. Robert. "Kummer Line". Toolbox for computing on Kummer lines. Oct. 2023. URL: https://gitlab.inria.fr/roberdam/kummer-line (cit. on pp. 2, 9, 65, 69, 79).

[Rob23c] D. Robert. "The geometric interpretation of the Tate pairing and its applications". Feb. 2023. URL: http://www.normalesup.org/~robert/pro/

publications/articles/geometric_tate_pairing.pdf. eprint: 2023/177, HAL: hal-04295743v1. (Cit. on pp. 13, 14, 74, 76).

[RS24] D. Robert and N. Sarkis. "Computing 2-isogenies between Kummer lines". In: *IACR Communications in Cryptology* 1 (1 Jan. 2024). URL: http://www.normalesup.org/~robert/pro/publications/articles/kummer_isogenies.pdf. eprint: 2024/037. (Cit. on p. 13).

[Sil86] J. H. Silverman. *The arithmetic of elliptic curves*. Vol. 106. Graduate Texts in Mathematics. Corrected reprint of the 1986 original. New York: Springer-Verlag, 1986, pp. xii+400. ISBN: 0-387-96203-4 (cit. on p. 54).

[Sta08] K. Stange. "Elliptic nets and elliptic curves". PhD thesis. Brown University, 2008. URL: https://repository.library.brown.edu/studio/item/bdr:309/PDF/ (cit. on pp. 4–6, 18, 20, 32, 46, 48, 54, 56, 75, 77).

[Sta11] K. Stange. "Elliptic nets and elliptic curves". In: *Algebra & Number Theory* 5.2 (2011), pp. 197–229 (cit. on pp. 4, 6, 9).

[Tra14] C. Tran. "Formules d'addition sur les jacobiennes de courbes hyperelliptiques: application à la cryptographie". PhD thesis. Rennes 1, 2014. URL: https://www.theses.fr/185903150 (cit. on pp. 6, 56).

[Ver10] F. Vercauteren. "Optimal pairings". In: *IEEE Transactions on Information Theory* 56.1 (2010), pp. 455–461 (cit. on p. 25).

INRIA Bordeaux–Sud-Ouest, 200 avenue de la Vieille Tour, 33405 Talence Cedex FRANCE
Email address: damien.robert@inria.fr
URL: http://www.normalesup.org/~robert/

Institut de Mathématiques de Bordeaux, 351 cours de la liberation, 33405 Talence cedex FRANCE