LMS J. Comput. Math. 19 (Special issue A) (2016) 267–282
doi:10.1112/S146115701600036X

Explicit isogenies in quadratic time in any characteristic

Luca De Feo, Cyril Hugounenq, Jérôme Plût and Éric Schost

Abstract

Consider two ordinary elliptic curves E, E' defined over a finite field $\mathbb{F}_q$, and suppose that there exists an isogeny ψ between E and E'. We propose an algorithm that determines ψ from the knowledge of E, E' and of its degree r, by using the structure of the ℓ-torsion of the curves (where ℓ is a prime different from the characteristic p of the base field). Our approach is inspired by a previous algorithm due to Couveignes, which involved computations using the p-torsion on the curves. The most refined version of that algorithm, due to De Feo, has a complexity of $\tilde{O}(r^2)p^{O(1)}$ base field operations. On the other hand, the cost of our algorithm is $\tilde{O}(r^2)\log(q)^{O(1)}$, for a large class of inputs; this makes it an interesting alternative for the medium- and large-characteristic cases.

1. *Introduction*

Isogenies are non-zero morphisms of elliptic curves, that is, non-constant rational maps preserving the identity element. They are also algebraic group morphisms. Isogeny computations play a central role in the algorithmic theory of elliptic curves. They are notably used to speed up Schoof's point-counting algorithm [**1**, **15**, **29**, **30**]. They are also widely applied in cryptography, where they are used to speed up point multiplication [**18**, **25**], to perform cryptanalysis [**26**], and to construct new cryptosystems [**6**, **12**, **20**, **35**, **39**].

The *degree* of an isogeny is its degree as a rational map. If an isogeny has degree r, we call it an r-isogeny, and we say that two elliptic curves are r-isogenous if there exists an r-isogeny relating them. Accordingly, we say that two field elements j and j' are r-isogenous if there exist r-isogenous elliptic curves E and E' such that $j(E) = j$ and $j(E') = j'$. The *explicit isogeny* problem has many incarnations. In this paper, we are interested in the variant defined below.

Explicit isogeny problem. Given two j-invariants j and j', and a positive integer r, determine if they are r-isogenous. In that case, compute curves E, E' with $j(E) = j$ and $j(E') = j'$, and the rational functions defining an r-isogeny $\psi : E \to E'$.

A good measure of the computational difficulty of the problem is given by the isogeny degree r. Indeed the output is represented by $O(r)$ base field elements, hence an asymptotically optimal algorithm would solve the problem using $O(r)$ field operations. Even though the input size is logarithmic in r, by a slight abuse we say that an algorithm solves the isogeny problem in polynomial time if it does so in the size of the output. Thanks to Vélu's formulas [**40**], in particular the version appearing in [**22**, § 2.4], we can compute ψ from knowledge of the polynomial h vanishing on the abscissas of the points in $\ker\psi$, at the cost of a constant number of multiplications of polynomials of degree $O(r)$. Given that all known algorithms to compute h

Received 22 February 2016.

2010 Mathematics Subject Classification 11Y40 (primary), 11G20, 14H52 (secondary).

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

This work was partially supported by the DIGITEO grant 2013-0531D (ARGC) and NSERC.

require more than a few polynomial multiplications, we often say that we have computed ψ whenever we have computed h, and conversely.

This paper focuses on the explicit isogeny problem for *ordinary* elliptic curves over finite fields. A famous theorem by Tate [**38**] states that two curves are isogenous over a finite field if and only if they have the same cardinality over that field. The explicit isogeny problem stated here appears naturally in the Schoof–Elkies–Atkin (SEA) point-counting algorithm. There, E is a curve over $\mathbb{F}_q$ whose rational points we wish to count, and E' is an r-isogenous curve with r a prime of size approximately $\log(q)$. For this reason, the explicit isogeny problem is customarily solved without prior knowledge of the cardinality of $E(\mathbb{F}_q)$. We abide by this convention here.

Many algorithms have been suggested over the years to solve the explicit isogeny problem. Early algorithms were due to Atkin [**2**] and Charlap, Coley and Robbins [**5**]. Elkies' [**3**, **15**] was the first algorithm targeted to finite fields (of large enough characteristic). Assuming r is prime, its complexity is dominated by the computation of the modular polynomial Φ_r, which is an object of bit size $O(r^3 \log(r))$. Later Bröker, Lauter and Sutherland [**4**] optimized the modular polynomial computation in the context of the SEA algorithm [**37**]. Finally, Lercier and Sirvent [**23**, **24**] generalized Elkies' algorithm to work in any characteristic. Despite these advances, the overall cost of Elkies' algorithm and its variants is still at least cubic in r.

Another line of work to solve the explicit isogeny problem for ordinary curves was initiated by Couveignes [**7**–**9**], and later improved by De Feo and Schost [**10**, **13**]. These algorithms use an interpolation approach combined with *ad hoc* constructions for towers of finite fields of characteristic p. Their complexity is quasi-quadratic in r, but exponential in $\log(p)$, hence they are only practical for very small characteristic.

In this paper we present a variant of Couveignes' algorithm with complexity polynomial in $\log(p)$ and quasi-quadratic in r. Like the original algorithm, it is limited to isogenies of ordinary curves. Together with the Lercier–Sirvent algorithm, they are the only polynomial-time isogeny computation algorithms working in any characteristic, hence they are especially relevant for counting points in *medium* characteristic (that is, counting points over $\mathbb{F}_{p^n}$, when $n \gg p/\log(p)$).

Note that although Couveignes-type algorithms do not make use of the modular polynomial Φ_r, its computation is still necessary in the context of the SEA algorithm. Thus our new algorithm does not improve the overall complexity of point counting, though it may provide a speed-up in some cases. It gives, however, an effective algorithm for solving the explicit isogeny problem, with potential applications in other contexts, for example, cryptography.

1.1. *Notation*

Throughout this paper r is a positive integer, p an odd prime, q a power of p, and $\mathbb{F}_q$ the finite field with q elements. E is an ordinary elliptic curve over $\mathbb{F}_q$, its group of n-torsion points is denoted by $E[n]$, its q-Frobenius endomorphism by π. The endomorphism ring of E is denoted by $\mathcal{O}$, with $K = \mathcal{O} \otimes \mathbb{Q}$ the corresponding number field, $\mathcal{O}_K$ its maximal order, and d_K the discriminant of $\mathcal{O}_K$. For a prime ℓ different from p and not dividing r, we denote by $E[\ell^k]$ the group of ℓ^k-torsion points of E, $E[\ell^\infty] = \varinjlim E[\ell^k]$ the union of all $E[\ell^k]$, and $T_\ell(E) = \varprojlim E[\ell^k]$ the ℓ-adic Tate module [**34**, III.7], which is free of rank 2 over $\mathbb{Z}_\ell$. The factorization of the characteristic polynomial of π over $\mathbb{Z}_\ell$ is determined by the Kronecker symbol (d_K/ℓ). If $(d_K/\ell) = +1$ then we also define λ, μ as the eigenvalues of π in $\mathbb{Z}_\ell$ and write $h = v_\ell(\lambda - \mu)$, where v_ℓ is the ℓ-adic valuation.

We measure all computational complexities in terms of operations in $\mathbb{F}_q$; the boolean costs associated to the algorithms presented next are negligible compared to the algebraic costs, and will be ignored. We use the Landau notation $O(\,)$ to express asymptotic complexities, and the notation $\tilde{O}(\,)$ to neglect (poly)logarithmic factors. We let $\mathsf{M}(n)$ be a function such that polynomials in $\mathbb{F}_q[x]$ of degree less than n can be multiplied using $\mathsf{M}(n)$ operations in $\mathbb{F}_q$, under the assumptions of [**41**, Chapter 8.3]. Using FFT multiplication, one can take $\mathsf{M}(n) \in O(n \log(n) \log\log(n))$.

1.2. *Couveignes' algorithm and our contribution*

Couveignes' isogeny algorithm takes as input two *ordinary* j-invariants $j, j' \in \mathbb{F}_q$, and a positive integer r not divisible by p, and returns, if it exists, an r-isogeny $\psi : E \to E'$, with $j(E) = j$ and $j(E') = j'$. It is based on the observation that the isogeny ψ must put $E[p^k]$ in bijection with $E'[p^k]$, in a way that is compatible with their structure as cyclic groups. It proceeds in three steps.

(1) Compute generators P, P' of $E[p^k]$ and $E'[p^k]$ respectively, for k large enough.
(2) Compute the interpolation polynomial L sending $x(P)$ to $x(P')$, and the abscissas of their scalar multiples accordingly.
(3) Deduce a rational fraction $g(x)/h(x)$ that coincides with L at all points of $E[p^k]$, and verify that it defines the x-component of an isogeny of degree r. If it does, return it; otherwise, replace P' with a scalar multiple of itself and go back to step (2).

For this algorithm to succeed, enough interpolation points are required. Given that the x-component of ψ is defined by $O(r)$ coefficients, we have $p^k \in \Theta(r)$. However, most of the time, those points will not be defined in the base field $\mathbb{F}_q$, so we must use efficient algorithms to construct and compute in towers of extensions of finite fields. Indeed, Couveignes and his successors go to great lengths to study the arithmetic of *Artin–Schreier towers* [**9**, **13**], and the adaptation of the fast interpolation algorithm to that setting [**10**]. Using these highly specialized constructions, steps (1) and (2) are both executed in time $\tilde{O}(p^{k+O(1)}) = \tilde{O}(rp^{O(1)})$. However, the last step only succeeds for one pair of torsion points P, P', in general, thus $O(r)$ trials are expected on average. Hence, the overall complexity of Couveignes' algorithm is $\tilde{O}(r^2 p^{O(1)})$, that is, quadratic in r, but exponential in $\log(p)$. Although the exponent of p is relatively small, Couveignes algorithm quickly becomes impractical as p grows.

In this paper we introduce a variant of Couveignes' algorithm with the same quadratic complexity in r, and *no exponential dependency in* $\log(p)$.

The bottom line of our algorithm is elementary: replace $E[p^k]$ in the algorithm with $E[\ell^k]$, for some small prime ℓ. However, a naive application of this idea fails to yield a quadratic-time algorithm. Indeed, in the worst case one has $\ell^{2k} \in \Theta(r)$, with $E[\ell^k] \simeq (\mathbb{Z}/\ell^k\mathbb{Z})^2$. Hence, two generators P, Q of $E[\ell^k]$ must be mapped onto two generators of $E'[\ell^k]$. This can be done in $O(\ell^{4k})$ possible ways, with a best possible cost of $O(\ell^{2k})$ per trial, thus yielding an algorithm of complexity $O(\ell^{6k}) = O(r^3)$ at best.

To avoid this pitfall, we carefully study in § 2 the structure of $E[\ell^k]$, and its relationship with the Frobenius endomorphism π. With that knowledge, we can put some restrictions on the generators P, Q, as explained in § 3, thus limiting the number of trials to $O(\ell^{2k})$. In § 4 we present an interpolation algorithm adapted to the setting of ℓ-adic towers, and in § 5 we put all steps together and analyze the full algorithm. Finally, in § 6 we discuss our implementation and the performance of the algorithm.

1.3. *Towers of finite fields*

The algorithms presented next operate on elements defined in finite extensions of $\mathbb{F}_q$. Specifically, we will work in a *tower* of finite fields $\mathbb{F}_q = F_0 \subset F_1 \subset \ldots \subset F_n$, with ℓ dividing $\#F_1 - 1$, $d_1 = [F_1 : F_0]$ dividing $\ell - 1$, and $[F_{i+1} : F_i] = \ell$ for any $i > 0$. For $\ell = 2$, we build upon the work of Doliskani and Schost [**14**], whereas for general ℓ we use towers of Kummer extensions in a way similar to [**11**, § 2]. Both constructions represent elements of F_i as univariate polynomials with coefficients in $\mathbb{F}_q$, thus basic arithmetic operations can be performed using modular polynomial arithmetic over $\mathbb{F}_q$. While constructing the tower, we also enforce special relations between the generators of each level, so that moving elements up and down the tower, and testing membership, can be done at negligible cost.

We briefly sketch the construction for odd ℓ. We first look for a primitive polynomial $P_1 \in \mathbb{F}_q[x]$ of degree equal to $[F_1 : F_0]$. There are many probabilistic algorithms to compute P_1 in expected time polynomial in ℓ and $\log(q)$; since their cost does not depend on the height n of the tower, we neglect it (in all that follows, by *expected cost* of an algorithm, we refer to a Las Vegas algorithm, whose runtime is given in expectation). Then, the image x_1 of x in $F_1 = \mathbb{F}_q[x]/P_1(x)$ is an element of multiplicative order $\#F_1 - 1$, and in particular it is not an ℓth power. Hence for any $i > 1$ we define F_i as $\mathbb{F}_q[x]/P_1(x^{\ell^{i-1}})$, the computation of the polynomials $P_1(x^{\ell^{i-1}})$ incurring no algebraic cost. Using this representation, elements of F_i can be expressed as elements of a higher level F_{i+j}, and reciprocally, by a simple rearrangement of the coefficients. Another fundamental operation can be done much more efficiently than in generic finite fields, as the following generalization of [**14**, § 2.3] shows.

LEMMA 1.1. *Let $F_0 \subset \ldots \subset F_n$ be a Kummer tower as defined above, and let $a \in F_i$ for some $0 \leqslant i \leqslant n$. For any integer j, we can compute the $(\#F_j)$th power of a using $O(\ell^{i-1}\mathsf{M}(\ell))$ operations in $\mathbb{F}_q$, after a precomputation independent of a of cost $O(\ell\mathsf{M}(\ell)\log(q))$.*

Proof. Without loss of generality, we can assume that $j < i$; otherwise, the output is simply a itself. Let $s = \#F_j$, and let $d = [F_i : F_1] = \ell^{i-1}$. Let x_i be the image of x in $F_i = \mathbb{F}_q[x]/P_i(x)$, so that $x_i^d = x_1$.

The first step, independently of a, is to compute $y = x_i^s$. Writing $s = ud + r$, with $r < d$, we see that y is given by $x_1^{u \bmod \#F_1} x_i^r$. We compute $x_1^{u \bmod \#F_1}$ using $O(\ell\mathsf{M}(\ell)\log(q))$ operations in $\mathbb{F}_q$, and we keep this element as a monomial of $F_1[x_i]$. By assumption, a is represented as a polynomial in x_i of degree less than $[F_i : F_0]$. We rewrite it as $a = a_0 + a_1x_i + \ldots + a_{d-1}x_i^{d-1}$, with $a_i \in F_1$. This is done by a simple rearrangement of the coefficients of a.

Finally, we compute $a(y)$ by a Horner scheme. All the powers y^k we need are themselves monomials in $F_1[x_i]$, each computed from the previous one using $O(\mathsf{M}(\ell))$ operations in $\mathbb{F}_q$, for a total of $O(\ell^{i-1}\mathsf{M}(\ell))$. Finally, the monomials $a_k y^k$ are combined together to form a polynomial in (x_1, x_i) of degree less than (d_1, d), and then expressed in a canonical form in F_i via another rearrangement of coefficients. □

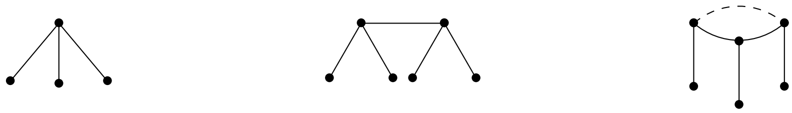
Summarizing, the following computations can be performed in a Kummer tower at the indicated asymptotic costs, all expressed in terms of operations in $\mathbb{F}_q$:

- basic arithmetic operations (addition, multiplication) in F_i, using $O(\mathsf{M}(\ell^i))$ operations;
- inversion in F_i using $O(\mathsf{M}(\ell^i)\log(\ell^i))$ operations (when $\ell = 2$, a factor of i can be saved here [**14**], but we will disregard this optimization for simplicity);
- mapping elements from F_{i-1} to F_i and vice versa at no arithmetic cost;
- multiplication and Euclidean division of polynomials of degree at most d in $F_i[x]$ using $O(\mathsf{M}(d\ell^i))$ operations, via Kronecker's substitution, as already done in, for example, [**42**];
- computing a $(\#F_j)$th power in F_i using $O(\ell^{i-1}\mathsf{M}(\ell))$ operations, after a precomputation that uses $O(\ell\mathsf{M}(\ell)\log(q))$ operations.

For one fundamental operation, we only have an efficient algorithm in the case $\ell = 2$, hence we introduce the following notation:

- $\mathsf{R}(i)$ is a bound on the expected cost of finding a root of a polynomial of degree ℓ in $F_i[x]$.

Note that we allow Las Vegas algorithms here, as no deterministic polynomial-time algorithm is known. For $\ell = 2$, Doliskani and Schost show that $\mathsf{R}(i) = O(\mathsf{M}(\ell^i)\log(\ell^i q))$. For general ℓ, we have $\mathsf{R}(i) = O(\ell^i\mathsf{M}(\ell^{i+1})\log(\ell)\log(\ell q))$ using the variant of the Cantor–Zassenhaus algorithm described in [**41**, Chapter 14.5], or $\mathsf{R}(i) = O((\ell^{i(\omega+1)/2} + \mathsf{M}(\ell^{i+1}\log(q)))i\log(\ell))$ using [**21**].



'Stromboli': $(d_K/\ell) = -1$ 'Vesuvius': $(d_K/\ell) = 0$ 'Etna': $(d_K/\ell) = +1$

FIGURE 1. *The three shapes of volcanoes of* 2*-isogenies.*

Here, ω is such that matrix multiplication in size m over any ring can be done in $O(m^\omega)$ base ring operations (so we can take $\omega = 2.38$ using the Coppersmith–Winograd algorithm). In any case, $\mathsf{R}(i)$ is between linear and quadratic in the degree ℓ^i.

2. *The Frobenius and the volcano*

In this section we explore some fundamental properties of ordinary elliptic curves over finite fields: the structure of their isogeny classes, its relationship with the rational ℓ^∞-torsion points, and with the Frobenius endomorphism π.

2.1. *Isogeny volcanoes*

For an extensive introduction to isogeny volcanoes we refer the reader to [**36**]. We recall here, without their proof, two results about ℓ-isogenies between ordinary elliptic curves.

PROPOSITION 2.1 [**22**, Proposition 21]. *Let $\phi : E \to E'$ be an ℓ-isogeny between ordinary elliptic curves and $\mathcal{O}, \mathcal{O}'$ be their endomorphism rings. Then one of the three following cases is true:*
(i) *$[\mathcal{O}' : \mathcal{O}] = \ell$, in which case we call ϕ ascending;*
(ii) *$[\mathcal{O} : \mathcal{O}'] = \ell$, in which case we call ϕ descending;*
(iii) *$\mathcal{O}' = \mathcal{O}$, in which case we call ϕ horizontal.*

PROPOSITION 2.2 [**22**, Proposition 23]; [**36**, Lemma 6]. *Let E be an ordinary elliptic curve with endomorphism ring $\mathcal{O}$.*
(i) *If $\mathcal{O}$ is ℓ-maximal then there are $(d_K/\ell) + 1$ horizontal ℓ-isogenies from E (and no ascending ℓ-isogenies).*
(ii) *If $\mathcal{O}$ is not ℓ-maximal then there are no horizontal ℓ-isogenies from E, and one ascending ℓ-isogeny.*

A *volcano of ℓ-isogenies* is a connected component of the graph of rational ℓ-isogenies between curves defined on $\mathbb{F}_q$. The *crater* is the subgraph corresponding to curves having an ℓ-maximal endomorphism ring. The shape of the crater is given by the Kronecker symbol (d_K/ℓ), as per Proposition 2.2. For any $k \geqslant 0$, an ℓ^k-isogeny is *horizontal* if it is the composite of k horizontal ℓ-isogenies. The *depth* of a curve is its distance from the crater. It is also the ℓ-adic valuation of the conductor of $\mathcal{O} = \mathrm{End}(E)$. We illustrate the three possible shapes for a volcano in Figure 1.

2.2. *The ℓ-adic Frobenius*

In the rest of this paper we consider only a volcano with a cyclic crater (that is, we assume $(d_K/\ell) = +1$), so that ℓ is an Elkies prime for these curves. This implies that the Frobenius automorphism on $T_\ell(E)$, which we write $\pi|T_\ell(E)$, has two distinct eigenvalues $\lambda \neq \mu$. The depth of the volcano of $\mathbb{F}_q$-rational ℓ-isogenies is $h = v_\ell(\lambda - \mu)$ [**36**, Theorem 7(iv)].

Proposition 2.3. *Let E be an ordinary elliptic curve with Frobenius endomorphism π. Assume that the characteristic polynomial of π has two distinct roots λ, μ in $\mathbb{Z}_\ell$, so that the ℓ-isogeny volcano has a cyclic crater. Then there exists a unique $e \in [\![0, h]\!]$ such that $\pi | T_\ell(E)$ is conjugate, over $\mathbb{Z}_\ell$, to the matrix $\left(\begin{smallmatrix}\lambda & \ell^e\\ 0 & \mu\end{smallmatrix}\right)$. Moreover, $e = h$ if E lies on the crater, otherwise $h - e$ is the depth of E in the volcano.*

We note here that the matrix $\left(\begin{smallmatrix}\lambda & \ell^h\\ 0 & \mu\end{smallmatrix}\right)$ is conjugate over $\mathbb{Z}_\ell$ to $\left(\begin{smallmatrix}\lambda & 0\\ 0 & \mu\end{smallmatrix}\right)$.

Proof. Since the characteristic polynomial of π splits over $\mathbb{Z}_\ell$, the matrix of $\pi|T_\ell(E)$ is trigonalizable. Conjugating the matrix $\left(\begin{smallmatrix}\lambda & a\\ 0 & \mu\end{smallmatrix}\right)$ by $\left(\begin{smallmatrix}1 & b\\ 0 & 1\end{smallmatrix}\right)$ replaces a by $a - b(\lambda - \mu)$, and conjugating by $\left(\begin{smallmatrix}c & 0\\ 0 & 1\end{smallmatrix}\right)$ replaces a by $c \cdot a$, so that the valuation $e = v_\ell(a)$ is an invariant under matrix conjugation. This proves the first part. For the second part, by Tate's theorem [**34**, Isogeny Theorem 7.7 (a)], $\mathcal{O} \otimes \mathbb{Z}_\ell$ is isomorphic to the order in $\mathbb{Q}_\ell[\pi_\ell]$ of matrices with integer coefficients, which is generated by the identity and $\ell^{-\min(h, v_\ell(a))}(\pi_\ell - \lambda)$. □

We now study the action of ℓ-isogenies on the ℓ-adic Frobenius by showing the link between two related notions of diagonalization.

Definition 2.4 (Horizontal and diagonal bases). Let E be a curve lying on the crater. We call a point of $E[\ell^k]$ *horizontal* if it generates the kernel of a horizontal ℓ^k-isogeny. We call a basis of $E[\ell^k]$ *diagonal* if π is diagonal in it, *horizontal* if both basis points are horizontal.

Proposition 2.5. *Let E be a curve lying on the crater and P be a point of $E[\ell^k]$ such that $\ell^h P$ is an eigenvector of π. Then $\ell^h P$ is horizontal if, and only if, P is an eigenvector for π. If $\pi(P) = \lambda P$ then we say that $\ell^h P$ has direction λ.*

This proposition being trivially true for $h \geqslant k$, we assume that $k \geqslant h$ in what follows.

Let R be a point of E of order ℓ^k, let ϕ be the isogeny with kernel $\langle R\rangle$, and let E' be its image. The subgroup $\langle R\rangle$ defines a point in the projective space of $E[\ell^k]$, which is a projective line over $\mathbb{Z}/\ell^k\mathbb{Z}$. There exists a canonical bijection [**32**, II.1.1] between this projective line and the set of lattices of index ℓ^k in the $\mathbb{Z}_\ell$-module $T_\ell(E)$: it maps a line $\langle R\rangle$ to the lattice $\Lambda_R = \langle R\rangle + \ell^k T_\ell(E)$. This lattice is also the preimage by ϕ of the lattice $\ell^k T_\ell(E')$.

Fix a basis (P, Q) of $E[\ell^k]$, let Π be the matrix of π in this basis, and let $R = xP + yQ$. The lattice Λ_R is generated by the columns of the matrix $L_R = \left(\begin{smallmatrix}\ell^k & 0 & x\\ 0 & \ell^k & y\end{smallmatrix}\right)$. The Hermite normal form of L_R is $M_R = \left(\begin{smallmatrix}\ell^{k-m} & x/y'\\ 0 & \ell^m\end{smallmatrix}\right)$, where we write $y = \ell^m y'$ with $\ell \nmid y'$, and the columns of M_R also generate the lattice Λ_R. We check that M_R has determinant ℓ^k. Since $\Lambda_R = \phi_R^{-1}(\ell^k T_\ell(E'))$, there exists a basis of $T_\ell(E')$ in which ϕ_R has matrix $\ell^k M_R^{-1}$. Therefore, in that basis of $T_\ell(E')$, the matrix of $\pi|T_\ell(E')$ is $M_R^{-1} \cdot \Pi \cdot M_R$.

Proof of Proposition 2.5. Fix a basis (R, S) of $E[\ell^k]$ that diagonalizes π. We can write $P = xR + yS$; without loss of generality we may assume $y = 1$. Let ϕ be the isogeny determined by $\ell^h P$, and let E' be its image. Since $\ell^h P$ is an eigenvector of π, ϕ is a rational isogeny. According to the previous discussion, $\pi|T_\ell(E')$ has matrix $\left(\begin{smallmatrix}\lambda & \ell^{h-k}x(\lambda-\mu)\\ 0 & \mu\end{smallmatrix}\right)$. This matrix is diagonalizable only if $v_\ell(x) \geqslant k - h$. On the other hand, we can compute $(\pi - \mu)P = x(\lambda - \mu)R$, so that P is an eigenvector on the same condition $v_\ell(x) \geqslant k - h$. □

While horizontal bases are our main interest, diagonal bases are easier to compute in practice. Algorithms computing bases of both kinds are given in § 3. The main tool for this is the next proposition: given a horizontal point of order ℓ^k, it allows us to compute a horizontal point of order ℓ^{k+1}.

PROPOSITION 2.6. *Let $\psi : E \to E'$ be a horizontal ℓ-isogeny with direction λ. For any point $Q \in E[\ell^\infty]$, if ℓQ is horizontal with direction μ, then $\psi(Q)$ is horizontal with direction μ.*

Proof. Let $Q' = \psi(Q)$ and $\widehat{\psi}$ be the isogeny dual to ψ. Since both $\widehat{\psi}$ and $\widehat{\psi}(Q') = \ell Q$ are horizontal with direction μ, Q' is also horizontal. □

PROPOSITION 2.7. *Let $\psi : E \to E'$ be an isogeny of degree r prime to ℓ.*

(i) *The curves E and E' have the same depth in their ℓ-isogeny volcanoes.*
(ii) *For any point $P \in E[\ell^k]$, the isogenies with kernel $\langle P\rangle$ and $\langle \psi(P)\rangle$ are of the same type (ascending, descending, or horizontal with the same direction).*
(iii) *If $P \in E[\ell]$ and $P' \in E'[\ell]$ are both ascending, or both horizontal with the same direction, then E/P and E'/P' are again r-isogenous.*

Proof. Points (i) and (ii) are consequences of Proposition 2.3 and of the fact that ψ, being rational and of degree prime to ℓ, induces an isomorphism between the Tate modules of E and E', commuting to the Frobenius endomorphisms. For point (iii), we just note that since there exists a unique subgroup of order ℓ which is either ascending or horizontal with a given direction, we must have $\langle P'\rangle = \langle \psi(P)\rangle$. □

2.3. *Galois classes in the ℓ-torsion*

Assume that E has a ℓ-maximal endomorphism ring. The following proposition summarizes the properties of $E[\ell^k]$ that we will need for our main interpolation algorithm. If ℓ is odd, let $\alpha = v_\ell(\lambda^{\ell-1} - 1)$ and $\beta = v_\ell(\mu^{\ell-1} - 1)$; if $\ell = 2$, let $\alpha = v_2(\lambda^2 - 1) - 1$ and $\beta = v_2(\mu^2 - 1) - 1$, and assume without loss of generality that $\alpha \geqslant \beta$. Since $\lambda \not\equiv \mu \pmod{\ell^{h+1}}$, it is impossible that $\lambda \equiv \mu \equiv 1 \pmod{\ell^h}$, so that at least one of the two valuations α, β is $\leqslant h$, and therefore $\beta \leqslant h$.

PROPOSITION 2.8. *For any k, let d_k be the degree of the smallest field extension $F/\mathbb{F}_q$ such that $E[\ell^k] \subset E(F)$. Then:*

(i) *the order of q in $(\mathbb{Z}/\ell\mathbb{Z})^\times$ divides d_1, and d_1 divides $(\ell - 1)$;*
(ii) *if ℓ is odd then for all $k \geqslant 1$, $d_k = \ell^{\min(v_\ell(d_1),k-\beta)}$;*
(iii) *if $\ell = 2$ then $d_2 \in \{1, 2\}$ and, for all $k \geqslant 2$, $d_k = \ell^{\min(v_\ell(d_2),k-\beta)}$;*
(iv) *letting $[F : \mathbb{F}_q] = d_1\ell^n$, the group $E[\ell^\infty](F)$ is isomorphic to $(\mathbb{Z}/\ell^{n+\alpha}\mathbb{Z}) \times (\mathbb{Z}/\ell^{n+\beta}\mathbb{Z})$;*
(v) *the group $E[\ell^k]$ contains at most $k \cdot \ell^{k+\beta}$ Galois conjugacy classes over $F_1 = \mathbb{F}_{q^{d_1}}$.*

Proof. The degree d_k is exactly the order of the matrix $\pi|E[\ell^k]$. It is therefore the least common multiple of the multiplicative orders of λ, μ modulo ℓ^k. This proves (i) using the fact that $\lambda \cdot \mu = q$. For points (ii)–(v) we may assume that $d_1 = 1$. Then, for any N, $v_\ell(\lambda^{2N} - 1) = \alpha + v_\ell(2N)$. Let (P, Q) be a diagonal basis of $E[\ell^k]$. The point $(\pi^N - 1)(xP + yQ) = (\lambda^N - 1)xP + (\mu^N - 1)yQ$ is zero if and only if $v_\ell(x) + \alpha + v_\ell(N) \geqslant k$ and $v_\ell(y) + \beta + v_\ell(N) \geqslant k$. This shows (iv). The largest Galois classes are those for which $v_\ell(y) = 0$ and their size is $\ell^{k-\beta}$, proving (ii) and (iii). Moreover, for any $i \leqslant k - \beta$ the points in an orbit of size less than or equal to ℓ^i are those for which $v_\ell(x) \geqslant k - \alpha - i$ and $v_\ell(y) \geqslant k - \beta - i$; there are at most $\ell^{\min(\alpha+i,k)+\min(\beta+i,k)}$ such points, and therefore $\ell^{\min(\alpha+i,k)+\min(\beta,k-i)} \leqslant \ell^{k-i+\beta}$ corresponding classes. Summing this over all i proves (v). □

3. *Computing the action of the Frobenius endomorphism*

We continue here our study of the action of the Frobenius π on $E[\ell^k]$. Given an ordinary elliptic curve E with ℓ-maximal endomorphism ring, we explicitly compute diagonal and horizontal bases of $E[\ell^k]$ as defined in the previous section. We will use the latter basis of $E[\ell^k]$ in § 5.2, to put restrictions on the interpolation problem of our algorithm.

We suppose that $k \geqslant h$. By Proposition 2.8, there exists a Kummer tower $F_0 \subset \ldots \subset F_{k-\beta}$ such that all the points of $E[\ell^k]$ are rational over $F_{k-\beta}$. The algorithms presented next assume that the tower has already been computed.

3.1. *Computation of a diagonal basis*

In Algorithm 1 we describe how to compute eigenvalues of the Frobenius mod ℓ^k and corresponding eigenvectors in the ℓ^k-torsion subgroup. We write $Q \leftarrow \mathsf{divide}(\ell, P)$ for the computation of a preimage of P by multiplication by ℓ.

Algorithm 1 Computing a diagonal basis of $E[\ell^k]$

Input: E, an ordinary, ℓ-maximal elliptic curve; k, a positive integer;
Output: (P_k, Q_k), a basis of $E[\ell^k]$; $\lambda, \mu \in \mathbb{Z}/\ell^k\mathbb{Z}$ such that $\pi(P_k) = \lambda P_k$, $\pi(Q_k) = \mu Q_k$.
1: $\lambda \leftarrow 0$; $\mu \leftarrow 0$; $P_0, Q_0 \leftarrow$ neutral element of $E[\ell]$.
2: **for** $i = 0$ to $k-1$ **do**
3: $\quad P' \leftarrow \mathsf{divide}(\ell, P_i)$; $Q' \leftarrow \mathsf{divide}(\ell, Q_i)$.
4: $\quad$ compute $\pi|(P', Q') = \begin{pmatrix} \lambda + a\ell^i & b\ell^i \\ c\ell^i & \mu + d\ell^i \end{pmatrix} \pmod{\ell^{i+1}}$.
5: $\quad$ **if** $b = 0$ **then** $x \leftarrow 0$; solve equation $c\ell^i + ((d-a)\ell^i + \mu - \lambda)y = 0$;
6: $\quad$ **else** solve equation $c\ell^i x^2 + ((d-a)\ell^i + \mu - \lambda)x - b\ell^i = 0$; $y \leftarrow -cx/b$; **end if**.
7: $\quad P_{i+1} \leftarrow P' + yQ'$; $Q_{i+1} \leftarrow xP' + Q'$.
8: $\quad \lambda \leftarrow \lambda + \ell^i(a + bx)$; $\mu \leftarrow \mu + \ell^i(d + cy)$.
9: **end for**
10: **return** (P_k, Q_k, λ, μ).

Proposition 3.1. *Algorithm 1 computes a diagonal basis of $E[\ell^k]$ using an expected $O(\mathsf{R}(k-\beta) + \ell^2\mathsf{M}(\ell^{k-\beta}) + \ell\mathsf{M}(\ell^2)\log(\ell)\log(\ell q))$ operations in $\mathbb{F}_q$.*

Proof. The equation at line 5 or 6 is first divided out by the largest power of ℓ possible, which is $\ell^{\min(h,i)}$, then solved modulo ℓ. For $i \leqslant h-1$, since $a = d$ and $b = c = 0$, the solutions are $x = y = 0$, and steps 5–7 do nothing. A straightforward calculation shows that after each loop the basis (P_{i+1}, Q_{i+1}) is diagonal.

For $i = 0$, the basis of $E[\ell](F_1)$ at step 3 is computed by factoring the ℓ-division polynomial at an expected cost of $O(\ell\mathsf{M}(\ell^2)\log(\ell)\log(\ell q))$ operations using the Cantor–Zassenhaus algorithm. Once $E[\ell]$ has been computed, we can factor the multiplication-by-ℓ map as a product of two ℓ-isogenies. Then, for any P defined in $E(F_{i-\beta})$, the computation of $\mathsf{divide}(\ell, P)$ at step 3 costs $O(\mathsf{R}(i-\beta+1))$ operations. Evaluating $\pi(P')$ in step 4 has a cost of $O(\ell^{i-\beta}\mathsf{M}(\ell))$. Writing $\pi(P')$ as a linear combination $\alpha P' + \beta Q'$ needs at most ℓ^2 point additions, with a cost of $\ell^2\mathsf{M}(\ell^{i-\beta+1})$. The cost of solving the equations at steps 5 and 6 by exhaustive search is negligible, as are the remaining operations. Since the cost of each loop grows geometrically, the last loop dominates all others, and gives the stated complexity. □

3.2. *Computation of a horizontal basis*

Using Algorithm 1, we can compute a diagonal basis of $E[\ell^{h+1}]$. By Proposition 2.5, this gives us a horizontal basis of $E[\ell]$. Thanks to Proposition 2.6, we can use this information to improve horizontal points of $E[\ell^i]$ into horizontal points of $E[\ell^{i+1}]$, as illustrated in Algorithm 2.

Proposition 3.2. *Algorithm 2 is correct and computes its output using an expected $O(\mathsf{R}(k-\beta) + k\mathsf{R}(h-\beta+1) + k\ell^2\mathsf{M}(\ell^{h-\beta+1}))$ operations in $\mathbb{F}_q$.*

Algorithm 2 Computing a horizontal point of order ℓ^k

Input: (P_0, Q_0), a diagonal basis of $E[\ell^{h+1}]$; k, an integer, $k \geqslant h+1$.
Output: R, a horizontal point of $E[\ell^k]$ with direction λ.
1: **for** $i = 1$ to $k-1$ **do**
2: $\quad \phi_i \leftarrow$ isogeny with kernel $\langle \ell^h P_{i-1} \rangle$
3: $\quad Q_i \leftarrow \phi_i(Q_{i-1})$
4: $\quad P' \leftarrow \mathsf{divide}(\ell, \phi_i(P_{i-1}))$.
5: $\quad$ Write $\pi(P') = \lambda P' + bQ_i$ for $b \in \mathbb{Z}/\ell\mathbb{Z}$ and let $P_i \leftarrow P' - (b/\mu)Q_i$.
6: **end for**
7: **return** $R = \widehat{\phi}_1 \circ \ldots \circ \widehat{\phi}_{k-1}(\mathsf{divide}(\ell^{k-(h+1)}, P_{k-1}))$.

Proof. Let E_i be the image curve of ϕ_i. We check that at step i of the loop, the points (P_i, Q_i) form a diagonal basis of $E_i[\ell^{h+1}]$, and ϕ_i has direction λ. The fact that R is horizontal is then a consequence of Proposition 2.6. The two most expensive operations in the loop are steps 4 and 5, costing respectively $O(\mathsf{R}(h-\beta+1))$ and $O(\ell^2\mathsf{M}(\ell^{h-\beta+1}))$, as discussed in the proof of Proposition 3.1. They are repeated k times. Finally, step 7 is dominated by the last divide operation, which costs $O(\mathsf{R}(k-\beta))$. □

One application of Algorithm 1 (with input $k \leftarrow h+1$) and two applications of Algorithm 2 allow us to compute a horizontal basis of $E[\ell^k]$. This could be done directly with Algorithm 1 instead, but that would require computing in an extension $F_{k+h-\beta}$.

4. *Interpolation step*

After constructing bases (P, Q) of $E[\ell^k]$ and (P', Q') of $E'[\ell^k]$ using the algorithms of the previous section, our algorithm computes the polynomial with coefficients in $\mathbb{F}_q$ mapping $x(P)$ to $x(P')$, $x(Q)$ to $x(Q')$, and the other abscissas accordingly. In this section we give an efficient algorithm for this specific interpolation problem. The algorithm appeared in [**10**] in the context of the Artin–Schreier extensions used in Couveignes' isogeny algorithm; it uses original ideas from [**16**]. We recall this algorithm here, and adapt the complexity analysis to our setting of Kummer extensions.

We start by tackling a simpler problem. We suppose we have constructed a tower of Kummer extensions $\mathbb{F}_q = F_0 \subset F_1 \subset \ldots \subset F_n$, with $[F_1 : F_0] \mid (\ell-1)$, and $[F_{i+1} : F_i] = \ell$ for any $i > 0$. Given two elements $v, w \in F_n \setminus F_{n-1}$, we want to compute polynomials T and L such that:

– $T \in \mathbb{F}_q[x]$ is the minimal polynomial of v, of degree $d = \deg T < \ell^n$;
– L is in $\mathbb{F}_q[x]$, of degree less than d, and $L(v) = w$.

Observe that, since $v, w \notin F_{n-1}$, we necessarily have $v_\ell(d) = n-1$, so that $\ell^{n-1} \leqslant d < \ell^n$. Using a fast interpolation algorithm [**41**, Chapter 10.2], the polynomials T and L could be computed in $O(n\mathsf{M}(\ell^{2n})\log(\ell))$ operations in $\mathbb{F}_q$. We can do much better by exploiting the form of the Kummer tower, and the Frobenius algorithm given in Lemma 1.1.

Following [**10**], we first compute T, starting from $T^{(0)} = x - v$. We let σ_i be the map that takes all the coefficients of a polynomial in $F_{n-i}[x]$ to the power $\#F_{n-i-1}$. For $i = 0, \ldots, n-1$, suppose we know a polynomial $T^{(i)}$ of degree ℓ^i in $F_{n-i}[x]$. Then compute the polynomials $T^{(i,j)}$ given by $T^{(i,j)} = \sigma_i^j(T^{(i)})$ for $0 \leqslant j \leqslant \ell-1$, and define

$$T^{(i+1)} = \prod_{j=0}^{b} T^{(i,j)} \quad \text{with } b = \begin{cases} \ell - 1 & \text{if } i < n-1, \\ d/\ell^{n-1} & \text{otherwise.} \end{cases} \tag{4.1}$$

One can easily see that $T^{(i+1)}$ is the minimal polynomial of v over F_{n-i+1}.

LEMMA 4.1. *The cost of computing T is $O(n\mathsf{M}(\ell^{n+1})\log(\ell))$ operations in $\mathbb{F}_q$.*

Proof. At each step i, from the knowledge of $T^{(i)}$ we compute all $T^{(i,j)}$ using Lemma 1.1. The cost for a single polynomial $T^{(i,j)}$ is $O(\ell^i\ell^{n-i-1}\mathsf{M}(\ell))$ operations, that is, $O(\ell^n\mathsf{M}(\ell))$ for all $O(\ell)$ of them. From the $T^{(i,j)}$ we compute $T^{(i+1)}$ using a subproduct tree, as in [**41**, Lemma 10.4]. The result has degree $O(\ell^{i+1})$ and coefficients in F_{n-i}, thus the overall cost is $O(\mathsf{M}(\ell^{n+1})\log(\ell))$. After $T^{(i+1)}$ is computed this way, we can convert its coefficients to F_{n-i-1} at no algebraic cost. Summing over all i, we obtain the stated complexity. □

We can finally proceed with the interpolation itself. First, compute $w' = w/T'(v)$ and let $L^{(0)} = w'$. Next, for $i = 0, \dots, n-2$, suppose we know a polynomial $L^{(i)}$ in $F_{n-i}[x]$ of degree less than ℓ^i. We compute the polynomials $L^{(i,j)}$ given by $L^{(i,j)} = \sigma_i^j(L^{(i)})$ and

$$L^{(i+1)} = \sum_{j=0}^{b} L^{(i,j)}\frac{T^{(i+1)}}{T^{(i,j)}}, \quad \text{with } b \text{ defined as in equation (4.1).}$$

As shown in [**10**], $L^{(n)}$ is the polynomial L we are looking for.

PROPOSITION 4.2. *Given $v, w \in F_n \setminus F_{n-1}$, the cost of computing the minimal polynomial $T \in \mathbb{F}_q[x]$ of v and the interpolating polynomial $L \in \mathbb{F}_q[x]$ such that $L(v) = w$ is $O(n\mathsf{M}(\ell^{n+1})\log(\ell))$ operations in $\mathbb{F}_q$.*

Proof. After the polynomials $T^{(i)}$ have been computed, we need to compute $T'(v)$. This is done by means of successive Euclidean remainders, since $T'(v) = (((T' \bmod T^{(1)}) \bmod T^{(2)}) \dots \bmod T^{(n)})$. At stage i, we have to compute the Euclidean division of a polynomial of degree $O(\ell^{n-i+1})$ by one of degree $O(\ell^{n-i})$ in $F_i[x]$. Using the complexities from § 1.3, we deduce that each division can be done in time $O(\mathsf{M}(\ell^{n+1}))$, for a total of $O(n\mathsf{M}(\ell^{n+1}))$ operations. Then computing $w' = w/T'(v)$ takes $O(\mathsf{M}(\ell^n)\log(\ell^n))$ operations.

Finally, at each step i, the polynomials $L^{(i,j)}$ are computed at a cost of $O(\ell^n\mathsf{M}(\ell))$, as in the proof of Lemma 4.1. The computation of $L^{(i+1)}$ uses the same subproduct tree as for the computation of $T^{(i)}$, requiring $O(\log\ell)$ additions, multiplications and divisions of polynomials of degree $O(\ell^{i+1})$ with coefficients in F_{n-i}, for a total of $O(\mathsf{M}(\ell^{n+1})\log(\ell))$. Summing over all i, the complexity statement readily follows. □

We end with the general problem of interpolating a polynomial in $\mathbb{F}_q[x]$ at points of F_n.

PROPOSITION 4.3. *Let $(v_1, w_1), \dots, (v_s, w_s)$ be pairs of elements of F_n, let t_i be the degree of the minimal polynomial of v_i, and let $t = \sum t_i$. The polynomials*

- *$T \in \mathbb{F}_q[x]$ of degree t such that $T(v_i) = 0$ for all i, and*
- *$L \in \mathbb{F}_q[x]$ of degree less than t such that $L(v_i) = w_i$ for all i,*

can be computed using $O(\mathsf{M}(t)\log(s) + n\mathsf{M}(\ell^2 t)\log(\ell))$ operations in $\mathbb{F}_q$.

Proof. The polynomial T is simply the product of all the minimal polynomials T_i. Let $n_i = v_\ell(t_i)$, so that $v_i, w_i \in F_{n_i+1} \setminus F_{n_i}$, and $\ell^{n_i} \leqslant t_i < \ell^{n_i+1}$. We convert (v_i, w_i) to a pair of elements of F_{n_i+1} at no algebraic cost, then we compute T_i as explained previously at a cost of $O(n\mathsf{M}(\ell^{n_i+2})\log(\ell))$ operations. Bounding ℓ^{n_i} by t_i, summing over all i, and using the superlinearity of M, we obtain a total cost of $O(n\mathsf{M}(\ell^2 t)\log(\ell))$ operations. Simultaneously, we compute all the polynomials L_i such that $L_i(v_i) = w_i$, at the same cost.

Then we arrange the T_i into a binary subproduct tree and multiply them together. A balanced binary tree, though not necessarily optimal, has a depth of $O(\log(s))$, and requires $O(\mathsf{M}(t))$ operations per level. Thus we can bound the cost of computing T by $O(\mathsf{M}(t)\log(s))$.

Finally, using the same subproduct tree structure, we apply the Chinese remainder algorithm of [41, Chapter 10] to compute the polynomial L at the same cost $O(\mathsf{M}(t)\log(s))$. $\square$

5. *The complete algorithm*

We finally come to the description of the full algorithm. Given two j-invariants, defining two elliptic curves E and E', and an integer r, we want to compute an isogeny $\psi : E \to E'$ of degree r. Since the algorithms of § 3 apply to curves on top of volcanoes with cyclic crater, we first need to determine a small Elkies prime ℓ for E and E', and then reduce to an explicit isogeny problem on the crater of the ℓ-volcanoes. These steps are discussed and analyzed next.

5.1. *Finding a suitable ℓ-volcano*

Our algorithm uses an Elkies prime ℓ. Since d_K is not yet assumed to be known, we need to be able to compute the height h of the volcano, the shape of its crater, as well as the shortest ℓ-isogeny chain from E to the crater.

The algorithms of Fouquet and Morain [17] compute the height h and find a curve $E_{\max}$ on the crater at the cost of $O(\ell h^2)$ factorizations of the ℓth modular polynomial Φ_ℓ. The polynomial Φ_ℓ is computed using $\tilde{O}(\ell^3\log(\ell))$ boolean operations, then each factorization costs an expected $O(\mathsf{M}(\ell)\log(\ell)\log(\ell q))$ operations using the Cantor–Zassenhaus algorithm (more efficient methods for special instances of volcanoes are presented in [27] and in [19], but we do not discuss them). Working on E and E', we compute the shortest path of ℓ-isogenies $\alpha : E \to E_{\max}$, $\alpha' : E' \to E'_{\max}$ linking the curves E, E' to the craters. We still have to determine the shape of these craters. Since the height h of the volcano is known, using Algorithm 1 we can compute a matrix of $\pi|E_{\max}[\ell^{h+1}]$. If this matrix has two distinct eigenvalues then the crater is cyclic, otherwise it is not.

By Proposition 2.7, the depth of E and E' below their respective craters is the same. By Proposition 2.7(iii), the curves $E_{\max}$ and $E'_{\max}$ are again r-isogenous; we can use our algorithm to compute such an isogeny $\psi_{\max}$. Then, since ℓ is coprime to r, $\psi = (\alpha')^{-1} \circ \psi_{\max} \circ \alpha$ is well defined and is the required r-isogeny. Its kernel can be computed in $O(h\mathsf{M}(\ell r)\log(\ell r))$ operations by evaluating the dual isogeny $\hat{\alpha}$ on the kernel of $\psi_{\max}$ via a sequence of resultants.

5.2. *Interpolating the isogeny*

We now assume that both curves E, E' have ℓ-maximal endomorphism rings. We fix bases of $E[\ell^k]$, $E'[\ell^k]$ and write π, π' for the matrices of the Frobenius. Since ψ is rational, its matrix satisfies the relation $\pi' \cdot \psi = \psi \cdot \pi$ in $\mathbb{Z}_\ell^{2\times 2}$ and hence in $(\mathbb{Z}/\ell^k\mathbb{Z})^{2\times 2}$.

If diagonal bases of $E[\ell^k], E'[\ell^k]$ are used, then, since π is a cyclic endomorphism of $\mathbb{Z}_\ell^2$, this condition seems to ensure that ψ is a diagonal matrix; however, $\mathbb{Z}/\ell^k\mathbb{Z}$ is not an integral domain and π is congruent, modulo ℓ^h, to the scalar matrix λ, so we can only say that $\psi \pmod{\ell^{k-h}}$ is diagonal. If on the other hand we choose *horizontal* bases of $E[\ell^k], E'[\ell^k]$ then, by Proposition 2.7(ii), we know that ψ is a diagonal matrix.

We then enumerate all the ℓ^{2k-2} invertible diagonal matrices; for each matrix M, we interpolate the action of M on $E[\ell^k]$ as a rational fraction, and verify that it is an r-isogeny. The successful interpolation will be our explicit isogeny ψ. Specifically, we interpolate using the abscissas of non-zero points of $E[\ell^k]$; there are $(\ell^{2k}-1)/2$ distinct such abscissas (or $2^{2k-1}+1$ when $\ell = 2$). The isogeny ψ acts on abscissas as a rational fraction of degrees $(r, r-1)$, which is thus defined by $2r$ coefficients; knowing this rational function allows us to find the kernel of ψ, and recover ψ itself using Vélu's formulas. For this method to work, we therefore select the smallest $k \geqslant h+1$ such that $\ell^{2k}-1 > 4r$.

Summarizing, our algorithm for two ℓ-maximal curves proceeds as follows.

(1) Use Algorithms 1 and 2 to compute horizontal bases $(P,Q),(P',Q')$ of $E[\ell^k],E'[\ell^k]$.

(2) Compute the polynomial T vanishing on the abscissas of $\langle P,Q\rangle$ as in § 4.

(3) For each invertible diagonal matrix $\left(\begin{smallmatrix} a & 0 \\ 0 & b\end{smallmatrix}\right)$ in $(\mathbb{Z}/\ell^k\mathbb{Z})^{2\times 2}$:

  (i) compute the interpolation polynomial $L_{a,b}$ such that $L_{a,b}(x(uP+vQ)) = x(a\,u\,P' + b\,v\,Q')$ for all $u,v\in\mathbb{Z}/\ell^k\mathbb{Z}$;

  (ii) use the *Cauchy interpolation algorithm* of [**41**, Chapter 5.8] to compute a rational fraction $F_{a,b}\equiv L_{a,b} \pmod T$ of degrees $(r,r-1)$;

  (iii) if $F_{a,b}$ defines an isogeny of degree r, return it and stop.

PROPOSITION 5.1. *Assuming that $\ell^h < \sqrt{r}$, the algorithm above computes an isogeny $\psi : E\to E'$ in expected time $O((r\ell^2\mathsf{M}(r\ell^4)+\mathsf{M}(r\ell^3)\log(\ell q))\log(r)\log(\ell))$.*

Proof. By definition of k, we know that $\ell^{2k}\in O(r\ell^2)$. By Proposition 2.8, there is a $\beta<h$ such that $E[\ell^k]$ is contained in $E(F_n)$ with $n=k-\beta$. We thus construct the Kummer tower $F_0\subset\ldots\subset F_n$, and we do the precomputations required by Lemma 1.1 at a cost of $O(\ell\mathsf{M}(\ell)\log(q))$.

Bounding h by $k-1$, step (1) costs on average $O(k\mathsf{R}(k-\beta)+k\ell^2\mathsf{M}(\ell\sqrt{r})+\ell\mathsf{M}(\ell^2)\log(\ell)\log(\ell q))$ according to Propositions 3.1 and 3.2. Using the most pessimistic estimates of § 1.3, we see that this cost is bounded by $O(\mathsf{M}(r\ell^3)\log(r)\log(\ell)\log(\ell q))$.

By Proposition 2.8(v), there are at most $O(k\cdot\ell^{k+\beta})$ Galois classes in $E[\ell^k]$. In order to apply the algorithms of § 4, we need to compute a representative for each class. Each representative is computed from the basis (P,Q) using point multiplication by two scalars less than or equal to ℓ^k in the field F_n, which costs $O(\mathsf{M}(\ell^n)\log(\ell^k))$ operations. We thus have a total cost of $O(k\mathsf{M}(\ell^{2k})\log(\ell^k))\subset O(\mathsf{M}(r\ell^2)\log(r)^2)$ to compute all such representatives.

Then, using Proposition 4.3, where the total degree is $t=(\ell^{2k}-1)/2\in O(r\ell^2)$, and the number of interpolation points is $s\in O(k\cdot\ell^{k+\beta})$, we can compute the polynomials T and $L_{a,b}$ at a cost of $O(\mathsf{M}(r\ell^4)\log(r)\log(\ell))$. The cost of computing $F_{a,b}$, and identifying the isogeny is dominated by that of computing $L_{a,b}$ [**10**, § 3.3]. Finally, in general approximately $\ell^{2k}=O(r\ell^2)$ candidate matrices must be tried before finding the isogeny. □

5.3. *Overall complexity*

By a result of Shparlinski and Sutherland [**33**, Theorem 1], for almost all primes q and curves $E/\mathbb{F}_q$, for $L\geqslant\log(q)^\varepsilon$ for any $\varepsilon>0$, asymptotically half of the primes $\ell\leqslant L$ are Elkies primes. Hence, we expect to have enough small Elkies primes to apply our algorithm. The following theorem states a worst-case bound depending on r and q alone.

THEOREM. *For almost all primes q and curves E,E' over $\mathbb{F}_q$, it is possible to solve the 'explicit isogeny problem' in expected time $O(r\mathsf{M}(r\log(q)^6)\log(r)\log\log(q))$.*

Proof. Given a curve E, we search for the smallest Elkies prime satisfying the conditions of Proposition 5.1. As a special case of [**33**, Theorem 1], we can take $L\in O(\log(q))$ such that the product of all Elkies primes $\ell\leqslant L$ exceeds $\Omega(\sqrt{q})$. On the other hand, we discard those primes $\ell\leqslant L$ for which the height h satisfies $\ell^h>\sqrt{r}$; since those discarded primes are divisors of $\sqrt{d_K}$, their product is at most $O(\sqrt{q})$. This shows that there remains enough 'good' Elkies primes in $[\![1,L]\!]$, so that in the worst case $\ell\in O(\log(q))$.

The most expensive steps in § 5.1 are the computation and the factorization of the modular polynomials for all primes up to ℓ. This is well within $O(\log(q)^6)$. The stated complexity then follows from substituting $\ell=O(\log(q))$ in Proposition 5.1. □

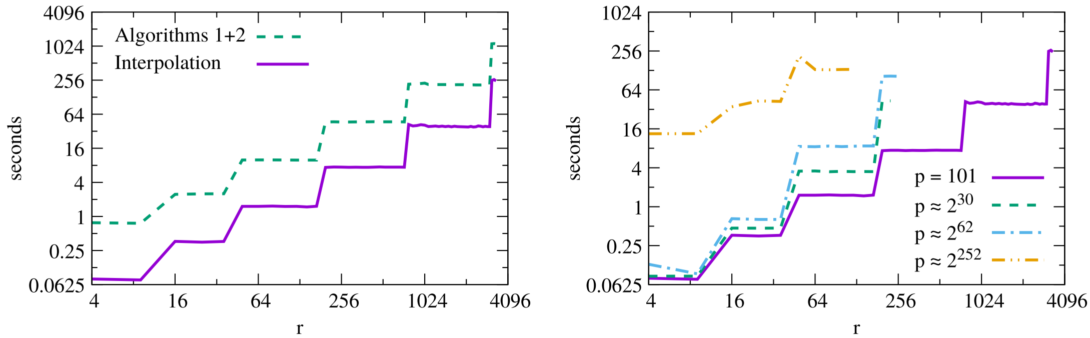


FIGURE 2. *Left: comparison of horizontal basis computation and interpolation phases, for a fixed curve defined over $\mathbb{F}_{101}$, and increasing r. Right: comparison of one interpolation phase for $\mathbb{F}_{101}$, $\mathbb{F}_{2^{30}+669}$, $\mathbb{F}_{2^{62}+189}$ and $\mathbb{F}_{2^{252}+421}$, and increasing r. Plots on logarithmic scale.*

6. *Conclusion and experimental results*

In the previous sections we have obtained a Las Vegas algorithm with an interesting asymptotic complexity. In particular, in the favorable case where $\ell = O(1)$, the running time of the algorithm is quasi-quadratic in the isogeny degree r and quasi-linear in $\log q$. Thus we expect it to be practical, and a substantial improvement over Couveignes' original algorithm, at least when small parameters ℓ and h can be found quickly. A large ℓ or h adversely affects performance in the following ways.

– All modular polynomials up to Φ_ℓ must be computed or retrieved from tables.
– All degrees $(\ell^{2k}-1)/4 \leqslant r < (\ell^{2k+1}-1)/4$ require essentially the same computational effort, thus resulting in a *staircase behavior* when r increases.
– Because we must have $k > h$, all degrees r smaller than $(\ell^{2h+2}-1)/4$ require the same computational effort.

For these reasons, it is wisest in practice to set small *a priori* bounds on ℓ and h, and only run our algorithm when parameters within these bounds can be found.

To validate our findings, we implemented a simplified version of our main algorithm using SageMath v7.1 [**28**]. In our current implementation, we only handle the case $\ell = 2$ and only work with curves on the crater of a 2-volcano. We implemented the construction of Kummer towers described in [**14**], in the favorable case where $p = 1 \bmod 4$. Source code and benchmark data are available in the GitHub project https://github.com/Hugounenq-Cyril/Two_curves_on_a_volcano/.

We ran benchmarks on an Intel Xeon E5530 CPU clocked at 2.4 GHz. We fixed a base field $\mathbb{F}_q$ and an elliptic curve E with height $h = 3$ and $\beta = 2$, then ran our algorithm to compute the multiplication-by-r isogeny $E \to E$, for r increasing. The torsion levels involved in the computations varied from 2^3 to 2^8. Figure 2 (left) shows the running times for the computation of the horizontal basis of $E[\ell^k]$, and for one execution of the interpolation step. Running times are close to linear in r, as expected. The staircase behavior of our algorithm is apparent from the plot. Since the interpolation steps must be repeated $\sim r$ times, we focus on this step to compare the running time for different base fields. In Figure 2 (right) we observe that the dependency in q, although much better than in Couveignes' original algorithm, is higher than the theoretical analysis would predict. This may due to low-level implementation details of SageMath, which, in the current implementation, are beyond our control.

In conclusion, our algorithm shows promise of being of practical interest within selected parameter ranges. Generalizing it to work with Atkin primes would considerably enlarge its applicability range; we hope to develop such a generalization in a future work. On the practical side, we plan to work on two improvements that seem within reach. First, the reduction from generic curves to ℓ-maximal curves seems superfluous and unduly expensive; it would be interesting to generalize the concept of horizontal bases to any curve. Second, a multi-modular approach interpolating on a torsion group of composite order is certainly possible, and could improve the running time of our algorithm by allowing it to work in smaller extension fields.

Appendix A. *Galois classes in* $E[\ell^k]$

We give here the full decomposition of $E[\ell^k]$ in Galois classes. This is a more precise form of Proposition 2.8(v).

Proposition A.1. *Let E be an elliptic curve with ℓ-maximal endomorphism ring. Assume $\ell \neq 2$, $\lambda \equiv \mu \equiv 1 \pmod{\ell}$ and let $\alpha = v_\ell(\lambda - 1), \beta = v_\ell(\mu - 1)$. Write $\nu(x, y) = \min(x + y, x+\beta-1, y+\alpha-1)$ and $\rho(x, y) = x+y-\nu(x, y) = \max(0, x-\alpha+1, y-\beta+1)$. The decomposition of the group $E[\ell^k]$ in Galois classes is as follows:*
- (i) *for $i, j = 1, \dots, k - 1$, $(\ell - 1)^2 \cdot \ell^{\nu(i,j)}$ classes of size $\ell^{\rho(i,j)}$;*
- (ii) *for $i = 1, \dots, k-1$, $(\ell-1)\cdot\ell^{\min(i,\alpha-1)}$ classes of size $\ell^{\max(0,i-\alpha+1)}$, and $(\ell-1)\cdot\ell^{\min(i,\beta-1)}$ classes of size $\ell^{\max(0,i-\beta+1)}$;*
- (iii) *the ℓ^2 singleton classes of $E[\ell]$.*

Proof. Fix a basis (P, Q) of $E[\ell^k]$ such that $\pi(P) = \lambda P$, $\pi(Q) = \mu Q$. Studying the Galois orbits of $E[\ell^k]$ means studying the map $\mathbb{Z}_\ell^2 \to \mathbb{Z}_\ell^2, (x, y) \mapsto (\lambda x, \mu y)$. In other words, the orbits correspond to elements of $\mathbb{Z}_\ell^2$ modulo the *multiplicative* subgroup generated by (λ, μ). An easy way to describe this is to consider a *multiplicative lattice* in $(\mathbb{Q}_\ell^\times)^2$.

Let ξ be a primitive $(\ell - 1)$th root of unity in $\mathbb{Z}_\ell$. Then by [**31**, Théorème II.3.2], the map $f(x, y, z) = \ell^x \cdot \xi^y \cdot \exp(\ell z)$ is a group isomorphism between $\mathbb{Z}\times(\mathbb{Z}/(\ell-1)\mathbb{Z})\times\mathbb{Z}_\ell$ and $\mathbb{Q}_\ell^\times$. For $i \in [\![0, k-1]\!]$ and $c \in \mathbb{Z}/(\ell-1)\mathbb{Z}$, let $V(i, c)$ be the image in $\mathbb{Z}/\ell^k\mathbb{Z}$ of the map $f(k-1-i, c, -)$; then the multiplicative structure of $V(i, c)$ is that of a principal homogeneous space under $\mathbb{Z}/\ell^i\mathbb{Z}$. We also define $W(i, j, c, d) = V(i, c) \cdot P + V(j, d) \cdot Q \subset E[\ell^k]$.

Since $\lambda \equiv 1 \pmod{\ell}$, we may write $\lambda = f(0, 0, u\,\ell^{\alpha-1})$ and $\mu = f(0, 0, v\,\ell^{\beta-1})$ for some u, $v \in \mathbb{Z}_\ell^\times$. This implies that the set $W(i, j, c, d)$ is stable under Galois. Moreover, the orbits of $W(i, j, c, d)$ correspond bijectively to points of a fundamental domain of the lattice $\Lambda_{i,j}$ generated by the columns of $\left(\begin{smallmatrix}\ell^i & 0 & u\ell^{\alpha-1}\\ 0 & \ell^j & v\ell^{\beta-1}\end{smallmatrix}\right)$, whereas the size of each orbit is $[(\mathbb{Z}/\ell^i\mathbb{Z})\times(\mathbb{Z}/\ell^j\mathbb{Z}) : \Lambda_{i,j}]$. By using elementary column manipulations, we find that the covolume of $\Lambda_{i,j}$ is $\ell^{\nu(i,j)}$, hence the point (i) of the proposition. (The case $i = j = 0$ yields singleton classes in $E[\ell]$.)

The union of all the sets $W(j, i, c, d)$ is exactly the set of all $xP+yQ$ for $x, y \neq 0$. We obtain the classes of (ii) by considering the sets $V(i, c) \cdot P$ and $V(j, d) \cdot Q$. □

We now state the equivalent proposition when $\ell = 2$. The proof is much the same as in the odd case.

Proposition A.2. *Let E be an elliptic curve with 2-maximal endomorphism ring. Assume $\lambda \equiv \mu \equiv 1 \pmod{4}$ and let $\alpha = v_2(\lambda - 1), \beta = v_2(\mu - 1)$. Write $\nu_2(x, y) = \min(x + y, x + \beta - 2, y + \alpha - 2)$ and $\rho_2(x, y) = x + y - \nu_2(x, y) = \max(0, x - \alpha + 2, y - \beta + 2)$. The decomposition of the group $E[2^k]$ in Galois classes is as follows:*
- (i) *for $i, j = 1, \dots, k - 2$, $4 \cdot 2^{\nu_2(i,j)}$ classes of size $2^{\rho_2(i,j)}$;*
- (ii) *for $i = 1, \dots, k - 2$, $4 \cdot 2^{\min(i,\alpha-2)}$ classes of size $2^{\max(0,i-\alpha+2)}$, and $4 \cdot 2^{\min(i,\beta-2)}$ classes of size $2^{\max(0,i-\beta+2)}$;*
- (iii) *the 16 singleton classes of $E[4]$.*

Note that if λ or $\mu \equiv -1 \pmod 4$ then by replacing the base field by a quadratic extension, we can always ensure that the condition $\lambda \equiv \mu \equiv 1 \pmod 4$ is satisfied.

Acknowledgement. We would like to thank the anonymous referees for their careful review and their insightful remarks.

References

1. A. O. L. Atkin, 'The number of points on an elliptic curve modulo a prime', mail to the nmbrthry mailing list, 1988.

2. A. O. L. Atkin, 'The number of points on an elliptic curve modulo a prime', mail to the nmbrthry mailing list, 1991.

3. A. Bostan, F. Morain, B. Salvy and É. Schost, 'Fast algorithms for computing isogenies between elliptic curves', *Math. Comp.* 77 (2008) no. 263, 1755–1778.

4. R. Bröker, K. Lauter and A. Sutherland, 'Modular polynomials via isogeny volcanoes', *Math. Comp.* 81 (2012) no. 278, 1201–1231.

5. L. S. Charlap, R. Coley and D. P. Robbins, 'Enumeration of rational points on elliptic curves over finite fields', Preprint, 1991.

6. D. X. Charles, K. E. Lauter and E. Z. Goren, 'Cryptographic hash functions from expander graphs', *J. Cryptology* 22 (2009) no. 1, 93–113.

7. J.-M. Couveignes, 'Quelques calculs en théorie des nombres', PhD Thesis, Université de Bordeaux, 1994.

8. J.-M. Couveignes, 'Computing l-Isogenies using the p-torsion', *ANTS-II: Proceedings of the Second International Symposium on Algorithmic Number Theory* (Springer, London, 1996) 59–65.

9. J.-M. Couveignes, 'Isomorphisms between Artin–Schreier towers', *Math. Comp.* 69 (2000) no. 232, 1625–1631.

10. L. De Feo, 'Fast algorithms for computing isogenies between ordinary elliptic curves in small characteristic', *J. Number Theory* 131 (2011) no. 5, 873–893.

11. L. De Feo, J. Doliskani and É. Schost, 'Fast algorithms for ℓ-adic towers over finite fields', *ISSAC'13: Proceedings of the 2013 International Symposium on Symbolic and Algebraic Computation* (ACM, New York, 2013) 165–172.

12. L. De Feo, D. Jao and J. Plût, 'Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies', *J. Math. Cryptol.* 8 (2014) no. 3, 209–247.

13. L. De Feo and É. Schost, 'Fast arithmetics in Artin–Schreier towers over finite fields', *J. Symbolic Comput.* 47 (2012) no. 7, 771–792.

14. J. Doliskani and É. Schost, 'Computing in degree 2^k-extensions of finite fields of odd characteristic', *Des. Codes Cryptogr.* 74 (2015) no. 3, 559–569.

15. N. D. Elkies, 'Elliptic and modular curves over finite fields and related computational issues', *Computational perspectives on number theory* (Chicago, IL, 1995), Studies in Advanced Mathematics 7 (AMS International Press, Providence, RI, 1998) 21–76.

16. A. Enge and F. Morain, 'Fast decomposition of polynomials with known Galois group', *AAECC'03: Proceedings of the 15th International Conference on Applied Algebra, Algebraic Algorithms and Error-correcting Codes* (Springer, Berlin, 2003) 254–264.

17. M. Fouquet and F. Morain, 'Isogeny volcanoes and the SEA algorithm', *Algorithmic number theory* (Sydney, 2002), Lecture Notes in Computer Science 2369 (Springer, Berlin, 2002).

18. R. P. Gallant, R. J. Lambert and S. A. Vanstone, 'Faster point multiplication on elliptic curves with efficient endomorphisms', *CRYPTO '01: Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology* (Springer, London, 2001) 190–200.

19. S. Ionica and A. Joux, 'Pairing the volcano', *Math. Comp.* 82 (2013) no. 281, 581–603.

20. D. Jao and V. Soukharev, 'Isogeny-based quantum-resistant undeniable signatures', *Post-Quantum Cryptography: 6th International Workshop, PQCrypto 2014* (Springer International Publishing, Waterloo, ON, 2014) 160–179.

21. E. Kaltofen and V. Shoup, 'Fast polynomial factorization over high algebraic extensions of finite fields', *ISSAC'97: Proceedings of the 1997 International Symposium on Symbolic and Algebraic Computation* (ACM, New York, 1997) 184–188.

22. D. Kohel, 'Endomorphism rings of elliptic curves over finite fields', PhD Thesis, University of California at Berkeley, 1996.

23. P. Lairez and T. Vaccon, 'On p-adic differential equations with separation of variables', *Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation, ISSAC'16* (ACM, New York, 2016) 319–323.

24. R. Lercier and T. Sirvent, 'On Elkies subgroups of ℓ-torsion points in elliptic curves defined over a finite field', *J. Théor. Nombres Bordeaux* 20 (2008) no. 3, 783–797.

25. P. Longa and F. Sica, 'Four-dimensional Gallant–Lambert–Vanstone scalar multiplication', *J. Cryptology* 27 (2014) no. 2, 248–283.
26. M. Maurer, A. Menezes and E. Teske, 'Analysis of the GHS Weil descent attack on the ECDLP over characteristic two finite fields of composite degree', *INDOCRYPT'01: Proceedings of the Second International Conference on Cryptology in India* (Springer, Berlin, 2001) 195–213.
27. J. M. Miret, R. Moreno, A. Rio and M. Valls, 'Determining the 2-sylow subgroup of an elliptic curve over a finite field', *Math. Comp.* 74 (2005) no. 249, 411–427.
28. The Sage Developers. 'Sage Mathematics Software (version 7.1)', 2016.
29. R. Schoof, 'Elliptic curves over finite fields and the computation of square roots mod p', *Math. Comp.* 44 (1985) no. 170, 483–494.
30. R. Schoof, 'Counting points on elliptic curves over finite fields', *J. Théor. Nombres Bordeaux* 7 (1995) no. 1, 219–254.
31. J.-P. Serre, *Cours d'arithmétique* (Presses Universitaires de France, 1970).
32. J.-P. Serre, *Arbres, amalgames,* SL_2, Astérisque 46 (Société Mathématique de France, Paris, 1977).
33. I. E. Shparlinski and A. V. Sutherland, 'On the distribution of Atkin and Elkies primes', *Found. Comput. Math.* 14 (2014) no. 2, 285–297.
34. J. H. Silverman, *The arithmetic of elliptic curves*, Graduate Texts in Mathematics 106 (Springer, New York, 1992).
35. A. Stolbunov, 'Constructing public-key cryptographic schemes based on class group action on a set of isogenous elliptic curves', *Adv. Math. Commun.* 4 (2010) no. 2, 215–235.
36. A. Sutherland, 'Isogeny volcanoes', *ANTS X: Proceedings of the Algorithmic Number Theory 10th International Symposium*, vol. 1 (Mathematical Sciences Publishers, Berkerley, CA, 2013) 507–530.
37. A. Sutherland, 'On the evaluation of modular polynomials', *ANTS X: Proceedings of the Algorithmic Number Theory 10th International Symposium*, vol. 1 (Mathematical Sciences Publishers, Berkerley, CA, 2013) 531–555.
38. J. Tate, 'Endomorphisms of abelian varieties over finite fields', *Invent. Math.* 2 (1966) no. 2, 134–144.
39. E. Teske, 'An elliptic curve trapdoor system', *J. Cryptology* 19 (2006) no. 1, 115–133.
40. J. Vélu, 'Isogénies entre courbes elliptiques', *C. R. Acad. Sci. Paris* 273 (1971) 238–241.
41. J. von zur Gathen and J. Gerhard, *Modern computer algebra* (Cambridge University Press, New York, 1999).
42. J. von zur Gathen and V. Shoup, 'Computing Frobenius maps and factoring polynomials', *STOC '92: Proceedings of the Twenty-Fourth Annual ACM Symposium on Theory of Computing* (ACM, New York, 1992) 97–105.

Luca De Feo
LMV – UVSQ
45 avenue des États-Unis
78035 Versailles
France

ORCID: 0000-0002-9321-0773

luca.de-feo@uvsq.fr

Cyril Hugounenq
LMV – UVSQ
45 avenue des États-Unis
78035 Versailles
France

hugounenq@msn.com

Jérôme Plût
ANSSI
51, boulevard de La Tour-Maubourg
75007 Paris
France

jerome.plut@ssi.gouv.fr

Éric Schost
Cheriton School of Computer Science
University of Waterloo
Waterloo, ON
Canada N2L 3G1

eschost@uwaterloo.ca