The AGM-$X_0(N)$ Heegner point lifting algorithm and elliptic curve point counting

David R. Kohel

School of Mathematics and Statistics
University of Sydney, NSW 2006, Australia
kohel@maths.usyd.edu.au

Abstract. We describe an algorithm, AGM-$X_0(N)$, for point counting on elliptic curves of small characteristic p using p-adic lifts of their invariants associated to modular curves $X_0(N)$. The algorithm generalizes the contruction of Satoh [10], SST [11], and Mestre [9]. We describe this method and give details of its implementation for characteristics 2, 3, 5, 7, and 13.

Key words: Elliptic curve cryptography, modular curves, point counting

1 Introduction

Elliptic curve cryptosystems can be designed using the reduction of precomputed CM curves or using randomly selected curves over a finite field. In the former case, the curve can be assumed to be drawn from a prespecified list of curves having many endomorphisms, on which an adversary can perform precomputations or exploit the existence of endomorphisms of small degree. On the general randomly selected curve, the only endomorphisms of small degree are scalar multiplication by a small integer. Such curves are believed to have higher security, but to implement an elliptic curve cryptosystem using randomly generated curves, it is imperative to have an efficient algorithm to determine the number of points on arbitrary elliptic curves.

The first theoretically polynomial time algorithm for point counting was due to Schoof [13]. Atkin and Elkies (see [3]) introduced the use of modular parametrizations of the torsion subgroups of elliptic curves to turn Schoof's algorithm into a practical one. Couveignes introduced an extension of this algorithm to curves over finite fields of small characteristic, and independently Lercier designed an efficient algorithm specific to characteristic 2.

In 1999, Satoh [10] introduced a novel idea of p-adically lifting the j-invariants of the cycle of curves which are related by the Frobenius isogeny $(x, y) \mapsto (x^p, y^p)$ over a finite field $\mathbb{F}_q = \mathbb{F}_{p^m}$ of small characteristic p. The j-invariants $j_0, j_1, \ldots, j_m = j_0$ can be lifted efficiently to a degree m extension of the p-adic field $\mathbb{Q}_p$ even though to lift the j-invariants to an extension of $\mathbb{Q}$ would in general require an extension of degree $O(\sqrt{q})$. The classical modular polynomial $\Phi_p(X, Y)$ provides the algebraic lifting condition. The unique p-adic lifts $\tilde{\jmath}_i$ are those for which the equations $\Phi_p(\tilde{\jmath}_i, \tilde{\jmath}_{i+1}) = 0$ continue to hold. This was followed by the exposition of extensions to characteristic 2 in [4] and [11]. Subsequently, in 2001, Mestre [9] introduced the use of the arithmetic–geometric mean, or AGM, to obtain elementary convergent recursion relations for the invariants of the p-adic lift of an elliptic curve.

In this work, we introduce a family of algorithms AGM-$X_0(N)$ given by convergent p-adic recursions for determining the p-adic lifts of *Heegner points* on modular curves $X_0(N)$. Heegner points are special points on modular curves which correspond to exceptional elliptic curves with CM, and are invariants from which we can "read off" the data for the trace of Frobenius, determining its number of points over $\mathbb{F}_q$. Specifically, we describe how the univariate version of Mestre's method as described in Gaudry [5] and Satoh [12] relates to the AGM-$X_0(8)$, and present essentially new generalizations AGM-$X_0(2)$, AGM-$X_0(4)$, and AGM-$X_0(16)$ which apply to point counting on elliptic curves in characteristic 2. In general this method is applicable to point counting on elliptic curves of any small characteristic p, with complete details described here for characteristics 2, 3, 5, 7, and 13.

The present work creates a general framework for point counting on elliptic curves over fields of small characteristic. While the AGM point counting method for even characteristic fields had outpaced comparable algorithms for curves over fields of other small characteristics, as well as the SEA for prime fields, the present AGM-$X_0(N)$ variants of the algorithm place all small characteristic base fields on an equal footing. Exploitation of the AGM for cryptographic constructions or any potential cryptanalytic attacks should therefore extend naturally to any small characteristic base field. The main elliptic curve standards admit only extensions of the binary field or large prime fields, but the omission of odd characteristic extension fields is not based on security considerations. Cryptographic standards for odd characteristic extension fields have been proposed [6], in part to permit efficient software implementations of curves over medium-sized characteristic [1]. A generic framework for odd characteristic extension fields also applies to fields of small characteristic, and makes it imperative to advance the theory of applicable algorithms and cryptographic characteristics of elliptic curves over arbitrary finite fields.

2 Modular curves and parametrizations

A modular curve $X_0(N)$ parametrizes elliptic curves together with some cyclic N-torsion subgroup. The simplest case is the modular curve $X_0(1)$ which classifies elliptic curves up to isomorphism via their j-invariants. Associated to any j other than 0 or 12^3, we can write down a curve

$$E : y^2 + xy = x^3 - \frac{36}{j - 12^3} x - \frac{1}{j - 12^3}$$

with associated invariant j. The curve $X_0(1)$ is identified with the line of j-values, each point corresponding to the class of curves with invariant j. The next simplest case is the curve $X_0(2)$, which is described by a function s_1, and which classifies an elliptic curve together with a 2-torsion subgroup.

$$E_1 : y^2 + xy = x^3 - 128s_1x^2 - \frac{36s_1}{64s_1+1}x + \frac{512s_1^2 - s_1}{64s_1+1}.$$

The j-invariant of this curve is $j = (256s_1+1)^3/s_1$ and the 2-torsion subgroup is specified by $P = (-1/4, 1/8)$. The quotient of the curve E_1 by this group gives a new curve

$$\begin{aligned} F_1 : y^2 + xy = x^3 - 128s_1x^2 - & \frac{327680s_1^2 + 3136s_1 + 5}{16(64s_1+1)}x \\ & + \frac{(512s_1+1)(262144s_1^2 + 1984s_1 + 3)}{64(64s_1+1)}, \end{aligned}$$

with j-invariant $(16s_1+1)^3/s_1^2$. If we try to put the curve F_1 into the form

$$E_2 : y^2 + xy = x^3 - 128s_2x^2 - \frac{36s_2}{64s_2+1}x + \frac{512s_2^2 - s_2}{64s_2+1}.$$

for some s_2, then we necessarily have an equality of their j-invariants

$$j(F_1) = (16s_1+1)^3/s_1^2 = (256s_2+1)^3/s_2 = j(E_2),$$

which gives rise to an relation $s_1^2 - 4096s_1s_2^2 - 48s_1s_2 - s_2 = 0$ between the s-invariants on E_1 and E_2, where we discard the trivial factor $4096s_1s_2 - 1$, determining the parametrized dual isogeny

$$\begin{tikzcd} E_1 \arrow[rr, "\phi"] & & E_1/\langle(0,0)\rangle = F_1 \arrow[d, "\cong"] \\ F_2 = E_2/\langle(0,0)\rangle \arrow[u, "\cong"] & & E_2 \arrow[ll, "\widehat{\phi}"'] . \end{tikzcd}$$

For the former equation, the resulting composition $\phi_1 : E_1 \to F_1 \cong E_2$, which may only exist over a quadratic extension of the field generated by s_1 and s_2, can be shown to induce the pullback $\phi_1^*\omega_2 = \pi(s_1,s_2)\omega_1$ where

$$\pi(s_1,s_2) = 2\left(\frac{(256s_1+1)(512s_2(64s_2+1) - 8s_1+1)}{(256s_2+1)(-256s_2(256s_2+1) + 16s_1+1)}\right)^{1/2},$$

and where ω_1 and ω_2 are the invariant differentials $dx/2y$ on the respective curves E_1 and E_2. Since the reduction of the relation between the s-invariants of the curves E_1 and E_2 gives $s_2 \equiv s_1^2 \bmod 2$, and the kernel is defined by to be those points (x,y) for which $4x - 1 \equiv 0$, we conclude that ϕ_1 defines a parametrized lift of the Frobenius isogeny.

The isogeny ϕ_1 can be extended similarly by an isogeny ϕ_2,

$$E_1 \longrightarrow F_1 \cong E_2 \longrightarrow F_2 \cong E_3 \longrightarrow \cdots$$

and the corresponding cycle of invariants $s_1, s_2, \ldots, s_m$, linked by a chain of isogeny relations, the product of the $\pi_i = \pi(s_i, s_{i+1})$ determines the action of Frobenius on the space of differentials of E_1 and we can read off its trace, which determines the number of points on the curve. This is the basis of the algorithm of Satoh [10] using the j-invariant and the algorithm of Mestre [9] using modular parametrizations of elliptic curves by the curve $X_0(8)$. The above example provides the equations necessary to use the curve $X_0(2)$ in an analogous manner.

2.1 Modular correspondences

The equation $\Phi(s_1, s_2) = s_1^2 - 4096 s_1 s_2^2 - 48 s_1 s_2 - s_2 = 0$, derived in the previous section, is an example of a modular correspondence. The function s on $X_0(2)$ generates the function field, and the relation between s_1 and s_2 determines the image of the modular curve $X_0(4)$ in the product $X_0(2) \times X_0(2)$.

At a high level we extend this construction as follows. A point on a modular curve $X_0(N)$ corresponds to the isomorphism class of a point (E, G), where E is an elliptic curve and G is a subgroup isomorphic to $\mathbb{Z}/N\mathbb{Z}$. For any such pair (E, G) we may associate the quotient curve $F = E/G$ together with the quotient isogeny $\phi : E \to F$. Conversely, to any isogeny $\phi : E \to F$ with cyclic kernel of order N, we can associate the pair $(E, \ker(\phi))$. We say that a map of curves $X_0(pN) \to X_0(N) \times X_0(N)$ is an *oriented modular correspondence* if the image of each point representing a pair (E, G) maps to $((E_1, G_1), (E_2, G_2))$ where $E_1 = E$ and G_1 is the unique subgroup of index p in G, and where $E_2 = E/H$ and $G_2 = G/H$, where H is the unique subgroup of order p. Since the composition

$$\phi : E = E_1 \to E_2 \to E_2/G_2 = E/G,$$

recovers the pair (E, G), one considers the point (E_2, G_2) as an extension of the degree N isogeny $\phi_1 : E_1 \to E_1/G_1$ determined by (E_1, G_1) to the isogeny of degree pN determined by (E, G). When the curve $X_0(N)$ has genus zero, there exists a single function x which generates its function field, and the correspondence can be expressed as a binary equation $\Phi(x, y) = 0$ in $X_0(N) \times X_0(N)$ cutting out $X_0(pN)$ inside of the product.

At a more basic level, the construction is determined as follows. Let $x = x(q)$ be a suitable modular function generating the function field of a genus zero curve $X_0(N)$, represented as a power series. Then $y = x(q^p)$ is a modular function on $X_0(pN)$, and an algebraic relation $\Phi(x, y) = 0$ determines an oriented modular correspondence as above. The application of modular correspondences to the lifting problem for elliptic curves is based on the following theorem.

Theorem 1. *Let p be a prime dividing N and let $\Phi(x, y) = 0$ be the equation defining an oriented modular correspondence $X_0(pN) \to X_0(N) \times X_0(N)$ on a modular curve $X_0(N)$ of genus zero such that $\Phi(x, y) \equiv y^p - x \bmod p$. Let $x_1, x_2, \ldots, x_m, x_{m+1} = x_1$ be a sequence of $m > 2$ distinct algebraic integers in some unramified extension of $\mathbb{Q}_p$ such that $\Phi(x_i, x_{i+1}) = 0$. Then the x_i form a Galois conjugacy class of invariants of CM curves.*

The above theorem describes the relation between cycles of points on modular curves and CM curves. A sequence of points satisfying the conditions of the theorem are examples of *Heegner points* on $X_0(N)$. After an initial precomputation to determine the equations as presented in this article, it is sufficient to dispense with the elliptic curves and compute only with their modular invariants. The defining functions and relations for the family determine the particular algorithm AGM-$X_0(N)$ to be used for p-adic lifting. Each is denoted according to the modular curve $X_0(N)$ on which we lift points. In each instance we have an initial condition of the form $x_1 \equiv 1/j \bmod p$ and a recursion for computing the function x_{i+1} in terms of x_i, which arises from the correspondence $X_0(2N) \to X_0(N) \times X_0(N)$ given by the equations $\Phi(x_i, x_{i+1}) = 0$ as below.

$$\begin{array}{ll} \underline{X_0(2)} : s_1^2 - 16(256s_2 + 3)s_1s_2 - s_2 = 0, & \underline{X_0(8)} : u_1^2(4u_2 + 1)^2 - u_2 = 0, \\ \underline{X_0(4)} : t_1^2 - 16(16t_1t_2 + t_1 + t_2)t_2 - t_2 = 0, & \underline{X_0(16)} : v_1^2(4v_2^2 + 1) - v_2 = 0. \end{array}$$

The relations between the above functions are given by the identities

$$\begin{array}{ll} j_1 = (256s_1 + 1)^3/s_1, & t_1 = u_1/(-4u_1^2 + 1), \\ s_1 = t_1(16t_1 + 1), & u_1 = v_1/(1 + 4v_1^2). \end{array}$$

Each function can be expressed in terms of the classical modular functions from which their relations were derived.

Families of p-adic liftings exist for odd characteristic, and in particular, when the genus of $X_0(N)$ is zero [1] we obtain a simple relation for the correspondence $X_0(pN) \to X_0(N) \times X_0(N)$. For instance, if $p = 3$ and N is 3 or 9 we give the correspondences defining algorithms AGM-$X_0(3)$ and AGM-$X_0(9)$ below.

$$\begin{array}{l} \underline{X_0(3)} : s_1^3 - 9\,(59049s_1s_2^2 + 2916s_1s_2 + 81s_2 + 30s_1 + 4)s_1s_2 - s_2 = 0 \\ \underline{X_0(9)} : t_1^3 - 9\,((27t_1^2 + 9t_1 + 1)(3t_2 + 1)t_2 + (3t_1 + 1)t_1)t_2 - t_2 = 0 \end{array}$$

The relations between these functions and the j-invariant is given by the equations:

$$\begin{array}{l} j_1 = (27s_1 + 1)(243s_1 + 1)^3/s_1, \\ s_1 = (27t_1^2 + 9t_1 + 1)t_1. \end{array}$$

2.2 Power series developments

Each of the selected functions are p-adically convergent away from the supersingular point $j_1 = 0 \bmod p$ when $p = 2$ or 3. The equations of the form $\Phi(x_i, x_{i+1}) = 0$ allow us to find a general solution for x_{i+1} as a power series

[1] The genus of $X_0(N)$ is zero if and only if N is one of the values 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 16, 18, or 25. In this case, there exists a single function which parametrizes $X_0(N)$. In the general case we would need multiple functions and the polynomial relations they satisfy. Here we will only be interested in the subset of these N which are powers of the characteristic p.

in x_i. We note that for all functions given above, j_1^{-1} is an initial approximation to the p-adic value of x_1.

$$
\begin{aligned}
&\underline{X_0(2):}\\
&s_{i+1} = s_i^2 - 48s_i^3 + 2304s_i^4 - 114688s_i^5 + 5898240s_i^6 + \cdots\\
&\quad = s_i^2(1-48s_i)(1+2304s_i^2)(1-4096s_i^3)(1+5701632s_i^4)\cdots\\
&\underline{X_0(4):}\\
&t_{i+1} = t_i^2 - 16t_i^3 + 240t_i^4 - 3584t_i^5 + 53760t_i^6 - 811008t_i^7 + \cdots\\
&\quad = t_i^2(1-16(t_i-15t_i^2))(1-3584(t_i^3+t_i^4))(1+13029376s_i^6)(1-8192s_i^5)\cdots\\
&\underline{X_0(8):}\\
&u_{i+1} = u_i^2 + 8u_i^4 + 80u_i^6 + 896u_i^8 + 10752u_i^{10} + 135168u_i^{12} + \cdots\\
&\quad = u_i^2(1+8u_i^2)(1+80u_i^4)(1+256u_i^6)(1+8704u_i^8)\cdots\\
&\underline{X_0(16):}\\
&v_{i+1} = v_i^2 + 4v_i^6 + 32v_i^{10} + 320v_i^{14} + 3584v_i^{18} + 43008t_i^{22} + \cdots\\
&\quad = v_i^2(1+4v_i^4)(1+32v_i^8)(1+192v_i^{12})(1+2816v_i^{16})(1+25600v_i^{20})\cdots
\end{aligned}
$$

Similarly the first few classes of algorithms on $X_0(3^n)$ give rise to the following p-adic analytic recursions.

$$
\begin{aligned}
&\underline{X_0(3):}\\
&s_{i+1} = s_i^3 - 36s_i^4 + 1026s_i^5 - 27216s_i^6 + 702027s_i^7 - 17898408s_i^8 + \cdots\\
&\quad = s_i^3(1-36s_i)(1+1026s_i^2)(1+9720s_i^3)\cdot\\
&\quad = s_i^3(1-36s_i)(1+1026s_i^2)(1+9720s_i^3)\cdot\\
&\qquad\qquad (1+1051947s_i^4)(1+9998964s_i^5+93927276s_i^6)\cdots\\
&\underline{X_0(9):}\\
&t_{i+1} = t_i^3 - 9t_i^4 + 54t_i^5 - 252t_i^6 + 891t_i^7 - 1701t_i^8 - 6426t_i^9 + \cdots\\
&\quad = t_i^3(1-9t_i-252t_i^3)(1+54t_i^2+649674t_i^6)(1+5265t_i^4)\cdot\\
&\qquad\qquad (1+486t_i^3+33048t_i^5+2925234t_i^7+98492517t_i^9)\cdots
\end{aligned}
$$

The above power series give explicit convergent series for the action of the Frobenius automorphism on ordinary CM points on the modular curves $X_0(p^n)$ for those particular values of p and n. The power product representations have the property that all but finitely many terms equal one to any fixed precision p^i. Note that the iteration $x_i \mapsto x_{i+1}$ is of the form $x_{i+1} = x_i^p f(x_i)$ for some power series $f(x_i)$ in x_i, and that the p-th powering gains relative precision. Thus in the initial phase we iterate the initial terms of the power product representation mod p^i to lift an approximation to the CM point as described in Table 1.

2.3 Action of Verschiebung

In order to apply the Heegner point constructions to the determination of the trace of Frobenius, we need to pullback of Frobenius between the differentials of parametrized curves specified by a modular correspondence. In Table 2 below we give the value of this scalar action of Verschiebung, the dual to Frobenius, in the left hand column. Using the identity $\mathrm{N}(x_1) = \mathrm{N}(x_2)$ for any Galois conjugates

Table 1. `HeegnerPointAnalyticLift`

Input: The modular polynomial $\Phi(x,y)$; the precomputed product decomposition for the analytic power series

$$y(x) = x^p f_1(x) f_2(x) \cdots \text{ such that } \Phi(x, y(x)) = 0$$

and $f_i(x) \equiv 1 \bmod p^i$; a finite field element x_0 such that $\Phi(x_0, x_0^p) = 0$; and a target precision w.
Output: An unramified p-adic lift x_1 of a Galois conjugate of x_0 such that (x_1, x_1^σ) is a zero of Φ to precision p^w.

Set x_1 to be any p-adic lift of x_0.
for $(1 \le k \le w-1)$ {
  $x_1 = x_1^p \prod_{i=1}^{k} f_i(x_1) \bmod p^{k+1}$
}
return x_1

x_1 and x_2, we are able to simplify the expressions by eliminating terms whose norm reduces to 1. In the final column we indicate with a 1 or 2 whether the expression is for the Verschiebung itself, or its square. In the latter case, we must extract a square root in the course of computing the norm.

Table 2. Modular Action of Verschiebung

	m-th power of Verschiebung	Norm-equivalent expression	m
$\underline{X_0(2)}$:			
	$\dfrac{(256s_2+1)(-256s_2(256s_2+1)+16s_1+1)}{(256s_1+1)(512s_2(64s_2+1)-8s_1+1)}$	$\dfrac{(-256s_2(256s_2+1)+16s_1+1)}{(512s_2(64s_2+1)-8s_1+1)}$	2
$\underline{X_0(4)}$:			
	$\dfrac{32t_2+1}{8t_1+1}$	$\dfrac{32t_1+1}{8t_1+1}$	2
$\underline{X_0(8)}$:			
	$\dfrac{(-4u_1+1)(4u_2+1)}{4u_2-1}$	$1+4u_1$	1
$\underline{X_0(16)}$:			
	$\dfrac{(-4v_1^2+1)(4v_2^2+1)}{4v_2^2-1}$	$1+4v_1^2$	1
$\underline{X_0(3)}$:			
	$\dfrac{(3s_1+1)(-19683s_1^2-486s_1+1)}{(243s_1+1)(-27s_1^2+18s_1+1)}$		2
$\underline{X_0(9)}$:			
	$\dfrac{(3t_1+1)(27t_1^2+1)(-243(81(27t_1^2+9t_1+1)^2t_1^2+2(27t_1^2+9t_1+1)t_1+1)}{(-27t_1^2+1)(243(27t_1^2+9t_1+1)t_1+1)(729t_1^4+486t_1^3+162t_1^2+18t_1+1)}$		2

3 Algorithm and performance

In order to construct the initial lifting of a finite field element to a p-adic element with precision w, we make use of the power series for x_{i+1} in terms of x_i as described in Table 1. Since the power series is approximated mod p by the congruence $x_{i+1} \equiv x_i^p \bmod p$, each application of this p-adic analytic function gains one coefficient of precision.

The analytic method, using a precomputed power product representation of the Hensel lifting of the power series appears to be more efficient than a naive linear Hensel lifting to compute the canonical lift to a precision of one 32-bit computer word. This is in part explained by the observation that a significant number of steps of the $f_i(x)$'s are in fact equal to 1, and so can be omitted from the product. Finally, we note that this product expression structures the Hensel lifting to use only multiplications.

The second phase of the lifting mirrors Algorithm 1 of SST [11], expressed here in terms of the Frobenius automorphism σ rather than its inverse. The algorithm of SST refers to the classical modular polynomial $\Phi_p(j_1, j_2)$ relating the j-invariants of two p-isogenous curves, but in fact applies in great generality [2] to find p-adic solutions to a bivariate polynomial $\Phi(x, y)$ for which $(x^p - y) \mid \Phi(x, y) \bmod p$.

Here we apply it to our modular correspondences $\Phi(x, y)$ on the curves $X_0(N)$. We define $\Phi_X(x, y)$ and $\Phi_Y(x, y)$ be the derivatives with respect to the first and second variable, respectively, of the modular correspondence. The algorithm is given in Table 3.

Table 3. `HeegnerPointBlockLift`

Input: The modular polynomial Φ, integers m and w, and a p-adic element x such that (x, x^σ) is a zero of Φ to precision p^w.

Output: A lift of x such that (x, x^σ) is a zero to precision p^{mw}.

$D_X = \Phi_X(x, x^\sigma) \bmod p^w$
$D_Y = \Phi_Y(x, x^\sigma) \bmod p^w$
for $(1 \le i \le m)$ {
  $R_x = \big(\Phi(x, x^\sigma) \text{ div } p^{iw}\big) \bmod p^w$
  for $(1 \le j \le w)$ {
    $\Delta_X = \big(R_x \bmod p\big)^{1/p}$ lifted to precision p^w
    $R_x = (R_x + D_X \Delta_X + D_Y \Delta_X^\sigma)/p$
    $x \mathrel{+}= p^{iw+j} \Delta_X$
  }
}
return x

The final step is to make use of the precomputed form of the action Frobenius on the differentials for an elliptic curve parametrization by $X_0(N)$. This action will be a rational function $\pi_1 = \pi(x_1)$ in the value x_1 of the lifted point. The Frobenius endomorphism is the product of the Galois conjugate Frobenius isogenies, so the norm $\mathrm{N}(\pi_1)$ of this value gives the action of the Frobenius endomorphism on the differentials. Since the minimal polynomial $X^2 - tX + q$ for this element it congruent to $X(X - t)$ modulo q, we see that $\mathrm{N}(\pi_1) \bmod q \equiv 0$

[2] This observation was already used by Gaudry [5] in extending this algorithm to a modified AGM modular equation.

and $(q \text{ div } \mathrm{N}(\pi_1)) \equiv t \bmod q$. In a now standard trick, the norm is computed using the identity $\mathrm{N}(\pi_1) = \exp(\mathrm{Tr}(\log(\pi_1)))$, using the efficiency of trace computation [11].

Table 4. Timing Data for AGM-$X_0(N)$

p = 2: m	$\log_2(q)$	$X_0(2)$	$X_0(4)$	$X_0(8)$	$X_0(16)$
163	163.00	0.48s	0.46s	0.45s	0.55s
193	193.00	0.61s	0.59s	0.60s	0.72s
239	239.00	0.91s	0.88s	0.91s	1.08s
p = 3:		$X_0(3)$	$X_0(9)$		
103	163.25	8.95s	10.8s		
121	191.78	19.7s	19.8s		
127	201.29	21.1s	21.2s		
151	239.33	43.5s	46.6s		
p = 5:		$X_0(5)$	$X_0(25)$		
71	164.86	8.06s	8.75s		
83	192.72	12.6s	13.5s		
103	239.16	30.5s	30.9s		
p = 7:		$X_0(7)$			
59	165.63	5.13s			
69	193.70	10.9s			
71	199.32	11.3s			
83	233.01	19.8s			
85	238.63	21.6s			
p = 13:		$X_0(13)$			
43	159.12	4.18s			
53	196.12	8.66s			
61	225.73	14.3s			
65	240.53	19.1s			

An generic implementation [7] of the method in Magma [8] yields the following timing data of Table 4 on an 1.4GHz AMD machine. The algorithm makes use of the internal Magma implementation of an efficient Galois action on unramified cyclotomic extensions when $p = 2$, and otherwise falls back on Hensel lifting to determine Galois images when the residue characteristic is odd. The timings listed are independent of the one-time setup costs for initializing the p-adic lifing rings. Further specific optimizations for $p = 2$ make this case comparatively faster than for odd residue characteristic.

4 Relations with other algorithms

The chosen model curve for $X_0(8)$ is the equation $u_1^2(4u_2 + 1)^2 = u_2$, which has the property that its reduction modulo 2 takes the form $u_1^2 = u_2$, so that u_2 is the Galois image of u_1. Over a field of characteristic zero, this equation becomes isomorphic to the equation arising in the "univariate" version of the

AGM recursion $4xy^2 = (x+1)^2$ via the change of variables[3]

$$x = \frac{1+4u_1}{1-4u_1} \text{ and } y = \frac{1+4u_2}{1-4u_2}.$$

Thus the use of 2-adic Heegner point lifts on $X_0(8)$ to determine the number of points on an elliptic curve over $\mathbb{F}_{2^m}$ could fall under a purported patent application on the AGM point counting method.[4]

In contrast, the modular curves $X_0(1)$, $X_0(2)$, $X_0(4)$, $X_0(8)$, or $X_0(16)$ are nonisomorphic as moduli spaces, and only the modular correspondence for $X_0(8)$ transforms by change of variables into the univariate AGM method. In fact if j is a root of the polynomial x^3+x+1 in $\mathbb{F}_2$, then the canonical lift of j on $X_0(1)$ is a root of the polynomial:

$$x^3 + 3491750x^2 - 5151296875x + 12771880859375.$$

The original method of Satoh, extended to characteristic 2 as in [4] or [11] finds some 2-adic approximation to a root of this polynomial. In contrast, in terms of the functions s, t, u, and v, the minimal polynomials over $\mathbb{Q}$ of a canonical lift are respectively:

$$\begin{gathered}
2^{36}x^6 + 2^{25}83x^5 + 14351421440x^4 + 412493295x^3 + 3503765x^2 + 166x + 1,\\
2^{24}x^6 + 2^{17}59x^5 + 1561856x^4 + 143007x^3 + 6101x^2 + 118x + 1,\\
2^6x^6 + 2^4 17x^5 + 572x^4 + 203x^3 + 13x^2 + 2x + 1,\\
2^3x^6 - 4x^5 + 18x^4 + 13x^3 + 9x^2 + 4x + 1.
\end{gathered}$$

The above polynomials are examples of class invariants obtained by modular correspondences on $X_0(1)$, $X_0(2)$, $X_0(4)$, $X_0(8)$, and $X_0(16)$, the latter examples naturally generalizing the construction of Couveignes and Henocq [2] for $X_0(1)$.

References

1. D. V. Bailey, C. Paar, Efficient arithmetic in finite field extensions with application in elliptic curve cryptography. *J. Cryptology*, **14** (2001), no. 3, 153–176.
2. J.-M. Couveignes and T. Henocq. Action of modular correspondences around CM points, *Algorithmic Number Theory (ANTS V, Sydney)*, 234–243, Lecture Notes in Computer Science, **2369**, Springer, Berlin, 2002.
3. N. Elkies. Elliptic and modular curves over finite fields and related computational issues, *Computational perspectives on number theory (Chicago, IL, 1995)*, 21–76, AMS/IP Stud. Adv. Math., **7**, Amer. Math. Soc., Providence, RI, 1998.

[3] Gaudry [5] makes a similar change of variables $x = 1/(1+8u)$ and $y = 1/(1+8v)$, from which he obtains the relation $(u+2v+8uv)^2 + (4u+1)v$, having the similar property of giving rise to an equation $u^2 = v$ between Galois conjugates modulo 2.

[4] Note however that the U.S. patent application concerns the "non-converging AGM iteration" (refer to `http://argote.ch`), as in Mestre's original binary AGM recursion [9], as distinct from Satoh's prior algorithm, the subsequent published univariate AGM recursions, and the variants described herein.

4. M. Fouquet, P. Gaudry, and R. Harley. An extension of Satoh's algorithm and its implementation, *J. Ramanujan Math. Soc.*, **15** (2000), 281–318.
5. P. Gaudry, A comparison and a combination of SST and AGM algorithms for counting points of elliptic curves in characteristic 2, *Advances in Cryptology – ASIACRYPT 2002*, 311–327, Lecture Notes in Computer Science, **2501**, Springer, Berlin, 2002.
6. A. Kato, T. Kobayashi, and T. Saito. Use of the Odd Characteristic Extension Field in the Internet X.509 Public Key Infrastructure, PKIX Working Group, Internet Draft, `http://www.ietf.org/internet-drafts/draft-kato--pkix-ecc-oef-00.txt`
7. D. Kohel, `http://magma.maths.usyd.edu.au/~kohel/magma`, 2003.
8. Magma Handbook, J. Cannon and W. Bosma, eds., `http://magma.maths.usyd.-edu.au/magma/htmlhelp/MAGMA.htm`, 2003
9. J.-F. Mestre, Lettre à Gaudry et Harley. `http://www.math.jussieu/~mestre`, 2001.
10. T. Satoh. The canonical lift of an ordinary elliptic curve over a finite field and its point counting. *J. Ramanujan Math. Soc.* **15** (2000), no. 4, 247–270.
11. T. Satoh, B. Skjernaa, and Y. Taguchi. Fast computation of canonical lifts of elliptic curves and its application to point counting, *Finite Fields Appl.* **9** (2003), no. 1, 89–101.
12. T. Satoh. On p-adic point counting algorithms for elliptic curves over finite fields, *Algorithmic number theory (ANTS V, Sydney)*, 43–66, Lecture Notes in Computer Science, **2369**, Springer, Berlin, 2002,
13. R. Schoof. Elliptic curves over finite fields and the computation of square roots mod p. *Math. Comput.* **44** (1985) 483–494.

5 Appendix of equations of higher level

In this appendix we give the equations for the modular correspondences and action of Verschiebung necessary to implement the AGM-$X_0(N)$ for $N = 5$, 25, 7, and 13. The modular correspondences on $X_0(5)$, $X_0(25)$, $X_0(7)$, and $X_0(13)$ with respect to a degree one function on the curve are as follows.

$\underline{X_0(5)}$:

$$\begin{aligned}&s_1^5 - 244140625 s_1^4 s_2^5 - 58593750 s_1^4 s_2^4 - 4921875 s_1^4 s_2^3 - 162500 s_1^4 s_2^2\\ &\quad - 1575 s_1^4 s_2 - 1953125 s_1^3 s_2^4 - 468750 s_1^3 s_2^3 - 39375 s_1^3 s_2^2 - 1300 s_1^3 s_2\\ &\quad - 15625 s_1^2 s_2^3 - 3750 s_1^2 s_2^2 - 315 s_1^2 s_2 - 125 s_1 s_2^2 - 30 s_1 s_2 - s_2 = 0\end{aligned}$$

$\underline{X_0(25)}$:

$$\begin{aligned}&t_1^5 - 625 t_1^4 t_2^5 - 625 t_1^4 t_2^4 - 375 t_1^4 t_2^3 - 125 t_1^4 t_2^2 - 25 t_1^4 t_2 - 625 t_1^3 t_2^5 - 625 t_1^3 t_2^4\\ &\quad - 375 t_1^3 t_2^3 - 125 t_1^3 t_2^2 - 25 t_1^3 t_2 - 375 t_1^2 t_2^5 - 375 t_1^2 t_2^4 - 225 t_1^2 t_2^3 - 75 t_1^2 t_2^2\\ &\quad - 15 t_1^2 t_2 - 125 t_1 t_2^5 - 125 t_1 t_2^4 - 75 t_1 t_2^3 - 25 t_1 t_2^2 - 5 t_1 t_2 - 25 t_2^5 - 25 t_2^4\\ &\quad - 15 t_2^3 - 5 t_2^2 - t_2 = 0\end{aligned}$$

The functions s on $X_0(5)$ and t on $X_0(25)$ are linked by the relation

$$s = 25t^5 + 25t^4 + 15t^3 + 5t^2 + t.$$

$\underline{X_0(7)}$:

$$
\begin{aligned}
&s_1^7 - 13841287201 s_1^6 s_2^7 - 7909306972 s_1^6 s_2^6 - 1856265922 s_1^6 s_2^5 - 224003696 s_1^6 s_2^4 \\
&\quad - 14201915 s_1^6 s_2^3 - 422576 s_1^6 s_2^2 - 4018 s_1^6 s_2 - 282475249 s_1^5 s_2^6 - 161414428 s_1^5 s_2^5 \\
&\quad - 37882978 s_1^5 s_2^4 - 4571504 s_1^5 s_2^3 - 289835 s_1^5 s_2^2 - 8624 s_1^5 s_2 - 5764801 s_1^4 s_2^5 \\
&\quad - 3294172 s_1^4 s_2^4 - 773122 s_1^4 s_2^3 - 93296 s_1^4 s_2^2 - 5915 s_1^4 s_2 - 117649 s_1^3 s_2^4 \\
&\quad - 67228 s_1^3 s_2^3 - 15778 s_1^3 s_2^2 - 1904 s_1^3 s_2 - 2401 s_1^2 s_2^3 - 1372 s_1^2 s_2^2 - 322 s_1^2 s_2 \\
&\quad - 49 s_1 s_2^2 - 28 s_1 s_2 - s_2 = 0
\end{aligned}
$$

$\underline{X_0(13)}$:

$$
\begin{aligned}
&s_1^{13} - 23298085122481 s_1^{12} s_2^{13} - 46596170244962 s_1^{12} s_2^{12} - 44804009850925 s_1^{12} s_2^{11} \\
&\quad - 27020264402404 s_1^{12} s_2^{10} - 11283187332872 s_1^{12} s_2^9 - 3409754413780 s_1^{12} s_2^8 \\
&\quad - 758378576462 s_1^{12} s_2^7 - 123855918940 s_1^{12} s_2^6 - 14548002326 s_1^{12} s_2^5 \\
&\quad - 1174999540 s_1^{12} s_2^4 - 59916584 s_1^{12} s_2^3 - 1623076 s_1^{12} s_2^2 - 15145 s_1^{12} s_2 \\
&\quad - 1792160394037 s_1^{11} s_2^{12} - 3584320788074 s_1^{11} s_2^{11} - 3446462296225 s_1^{11} s_2^{10} \\
&\quad - 2078481877108 s_1^{11} s_2^9 - 867937487144 s_1^{11} s_2^8 - 262288801060 s_1^{11} s_2^7 \\
&\quad - 58336813574 s_1^{11} s_2^6 - 9527378380 s_1^{11} s_2^5 - 1119077102 s_1^{11} s_2^4 \\
&\quad - 90384580 s_1^{11} s_2^3 - 4608968 s_1^{11} s_2^2 - 124852 s_1^{11} s_2 - 137858491849 s_1^{10} s_2^{11} \\
&\quad - 275716983698 s_1^{10} s_2^{10} - 265112484325 s_1^{10} s_2^9 - 159883221316 s_1^{10} s_2^8 \\
&\quad - 66764422088 s_1^{10} s_2^7 - 20176061620 s_1^{10} s_2^6 - 4487447198 s_1^{10} s_2^5 \\
&\quad - 732875260 s_1^{10} s_2^4 - 86082854 s_1^{10} s_2^3 - 6952660 s_1^{10} s_2^2 - 354536 s_1^{10} s_2 \\
&\quad - 10604499373 s_1^9 s_2^{10} - 21208998746 s_1^9 s_2^9 - 20393268025 s_1^9 s_2^8 \\
&\quad - 12298709332 s_1^9 s_2^7 - 5135724776 s_1^9 s_2^6 - 1552004740 s_1^9 s_2^5 \\
&\quad - 345188246 s_1^9 s_2^4 - 56375020 s_1^9 s_2^3 - 6621758 s_1^9 s_2^2 - 534820 s_1^9 s_2 \\
&\quad - 815730721 s_1^8 s_2^9 - 1631461442 s_1^8 s_2^8 - 1568712925 s_1^8 s_2^7 - 946054564 s_1^8 s_2^6 \\
&\quad - 395055752 s_1^8 s_2^5 - 119384980 s_1^8 s_2^4 - 26552942 s_1^8 s_2^3 - 4336540 s_1^8 s_2^2 \\
&\quad - 509366 s_1^8 s_2 - 62748517 s_1^7 s_2^8 - 125497034 s_1^7 s_2^7 - 120670225 s_1^7 s_2^6 \\
&\quad - 72773428 s_1^7 s_2^5 - 30388904 s_1^7 s_2^4 - 9183460 s_1^7 s_2^3 - 2042534 s_1^7 s_2^2 \\
&\quad - 333580 s_1^7 s_2 - 4826809 s_1^6 s_2^7 - 9653618 s_1^6 s_2^6 - 9282325 s_1^6 s_2^5 - 5597956 s_1^6 s_2^4 \\
&\quad - 2337608 s_1^6 s_2^3 - 706420 s_1^6 s_2^2 - 157118 s_1^6 s_2 - 371293 s_1^5 s_2^6 - 742586 s_1^5 s_2^5 \\
&\quad - 714025 s_1^5 s_2^4 - 430612 s_1^5 s_2^3 - 179816 s_1^5 s_2^2 - 54340 s_1^5 s_2 - 28561 s_1^4 s_2^5 \\
&\quad - 57122 s_1^4 s_2^4 - 54925 s_1^4 s_2^3 - 33124 s_1^4 s_2^2 - 13832 s_1^4 s_2 - 2197 s_1^3 s_2^4 \\
&\quad - 4394 s_1^3 s_2^3 - 4225 s_1^3 s_2^2 - 2548 s_1^3 s_2 - 169 s_1^2 s_2^3 - 338 s_1^2 s_2^2 \\
&\quad - 325 s_1^2 s_2 - 13 s_1 s_2^2 - 26 s_1 s_2 - s_2 = 0
\end{aligned}
$$

We note that a canonical lift only exists for the invariants of ordinary curves. The supersingular points, in contrast, fail to converge, and are in fact poles of each the chosen functions for the lifting process. In characteristics 2 and 3 the j-invariant 0 is supersingular, which explains why we take as starting point of our canonical lifting algorithm $1/j \equiv s_1 \equiv t_1 \cdots \bmod p$. For p equal to 5 the j-invariant of a supersingular curve is also 0, and the starting point of lifting is therefore also $1/j \equiv s_1 \equiv t_1 \bmod 5$. However for 7 and 13 the starting points of the lifing algorithms are $1/(j+1) \equiv s_1 \bmod 7$ and $1/(j-5) \equiv s_1 \bmod 13$, corresponding to the supersingular j-invariants 6 and 5, respectively.

To complete the specification of the algorithms for $X_0(5)$, $X_0(7)$, and $X_0(13)$, it remains to give the action of Verschiebung on the differentials of a generic curve as in Table 2. In terms of a special value s_1 which is the canonical lift of the invariants of an ordinary elliptic curve, we find the following form for square of the action of pullback by the Verschiebung on two parametrized curves.

$$\underline{X_0(5)}\colon \quad -\frac{G_5(s_1,1)H_5(5^3s_1,1)}{G_5(5^2s_1,1)H_5(1,s_1)}, \text{ where } \begin{cases} G_5(X,Y) = 5X^2+10XY+Y^2, \\ H_5(X,Y) = -X^2-4XY+Y^2. \end{cases}$$

$$\underline{X_0(7)}\colon \quad -\frac{F_7(s_1,1)(-7^7s_1^4+G_7(7^2s_1,1)+1)}{F_7(7^2s_1,1)(-7s_1^4+7G_7(1,s_1)+1)}, \text{ where}$$

$$\begin{aligned} F_7(X,Y) &= X^2+5XY+Y^2, \\ G_7(X,Y) &= (2X^2+9XY+10Y^2)XY. \end{aligned}$$

$$\underline{X_0(13)}\colon \quad -\frac{G_{13}(1,s_1)H_{13}(13s_1,1)}{G_{13}(13s_1,1)H_{13}(1,s_1)}, \text{ where}$$

$$\begin{aligned} G_{13}(X,Y) &= X^4+7X^3Y+20X^2Y^2+19XY^3+Y^4, \\ H_{13}(X,Y) &= X^6+10X^5Y+46X^4Y^2+108X^3Y^3+122X^2Y^4+38XY^5-Y^6. \end{aligned}$$

The action of Frobenius with respect to t_1 on $X_0(25)$ is determined by means of the expression for the function $s_1 = 25t_1^5+25t_1^4+15t_1^3+5t_1^2+t_1$ on $X_0(5)$ in terms of the function t_1 on $X_0(25)$.