Guillaume Hanrot
François Morain
Emmanuel Thomé (Eds.)

LNCS 6197

Algorithmic Number Theory

9th International Symposium, ANTS-IX
Nancy, France, July 2010
Proceedings

Springer

Lecture Notes in Computer Science 6197

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Guillaume Hanrot François Morain
Emmanuel Thomé (Eds.)

Algorithmic Number Theory

9th International Symposium, ANTS-IX
Nancy, France, July 19-23, 2010
Proceedings

Springer

Volume Editors

Guillaume Hanrot
LIP/ENS-Lyon, 46, allée d'Italie
69364 Lyon Cedex 07, France
E-mail: Guillaume.Hanrot@ens-lyon.fr

François Morain
LIX/École polytechnique
91128 Palaiseau Cedex, France
E-mail: Francois.Morain@lix.polytechnique.fr

Emmanuel Thomé
INRIA Nancy, projet CARAMEL
615 rue du jardin botanique
54602 Villers-lès-Nancy Cedex, France
E-mail: Emmanuel.Thome@inria.fr

Library of Congress Control Number: 2010930653

CR Subject Classification (1998): F.2, G.2, E.3, I.1

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

ISSN 0302-9743
ISBN-10 3-642-14517-5 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-14517-9 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper 06/3180

Preface

ANTS-IX was the ninth edition of the biennial International Symposium on Algorithmic Number Theory. The first edition of this symposium was held at Cornell University in 1994. ANTS-IX was held July 19-23, 2010 at INRIA in Nancy, France.

The ANTS-IX Program Committee consisted of 12 members whose names are listed on the next page. The selection of the accepted papers among the submissions was made from mid-January to end of March 2010. Each paper was thoroughly reviewed by at least two experts, including a Program Committee member. The Program Committee selected 25 high-quality articles, which are excellent representatives of the current state of the art in various areas of algorithmic number theory. The Selfridge Prize in computational number theory was awarded to the authors of the best contributed paper presented at the conference. We gratefully thank the authors of all submitted papers for their hard work which made the selection of a varied program possible. We also thank the authors of the accepted papers for their cooperation in the timely production of the revised versions.

Each submitted paper was presented by one of its co-authors at the conference. Besides contributed papers, the conference included five invited talks by Henri Darmon (McGill University), Jean-François Mestre (Université Paris 7), Gabriele Nebe (RWTH Aachen), Carl Pomerance (Dartmouth College), and Oded Regev (Tel-Aviv University). We thank the invited speakers for having been able to provide abstracts of their talk, which are reproduced in this volume. This list of invited speakers originally included Fritz Grunewald (HHU Düsseldorf), who unfortunately passed away on March 21, 2010, four months before the conference. A special lecture was held to honor his memory.

The conference organizers wish to thank all the people who made the conference possible. In particular, we gratefully acknowledge the support of the funding institutions.

May 2010

Guillaume Hanrot
François Morain
Emmanuel Thomé

Organization

Organizing Committee

Anne-Lise Charbonnier	INRIA, Nancy, France
Jérémie Detrey	INRIA, Nancy, France
Pierrick Gaudry (Chair)	CNRS, Nancy, France
Emmanuel Thomé	INRIA, Nancy, France
Paul Zimmermann	INRIA, Nancy, France

Program Committee

Nigel Boston	University of Wisconsin, USA
John Cremona	Warwick Mathematics Institute, UK
Claus Fieker	University of Sydney, Australia
Guillaume Hanrot (PC Chair)	École Normale Supérieure, Lyon, France
Kevin Hare	University of Waterloo, Canada
Thorsten Kleinjung	École Polytechnique Fédérale de Lausanne, Switzerland
Kamal Khuri-Makdisi	American University of Beirut, Lebanon
François Morain (PC Chair)	École Polytechnique, France
Takakazu Satoh	Tokyo Institute of Technology, Japan
Igor Shparlinski	Macquarie University, Australia
Alice Silverberg	University of California at Irvine, USA
Frederik Vercauteren	Katholieke Universiteit Leuven, Belgium

Poster Session

Benjamin Smith	INRIA Saclay, École Polytechnique, France

Sponsoring Institutions

Institut National de Recherche en Informatique et Automatique (INRIA)
Laboratoire Lorrain de Recherche en Informatique et Applications (LORIA)
École Polytechnique
Centre National de la Recherche Scientifique (CNRS)
Microsoft Research, USA
Nancy Université
Groupement de Recherches en Informatique Mathématique (GDR IM)
Communauté Urbaine du Grand Nancy
Conseil Régional de Lorraine

Conference Website

The names of the winners of the Selfridge Prize, material supplementing the contributed papers, and errata for the proceedings (if relevant), as well as the abstracts of the posters and the posters presented at ANTS-IX, can be found at `http://ants9.org/`.

Table of Contents

Invited papers

Contributed papers

Putting the Hodge and Tate Conjectures to the Test

Henri Darmon

Department of Mathematics,
McGill University, Burnside Hall, Montreal, QC, Canada
henri.darmon@mcgill.ca

The Hodge conjecture asserts that the presence of algebraic cycles on a (smooth, projective) variety over the complex numbers can be detected in its Betti cohomology equipped with the Hodge structure arising from its relation with complex deRham cohomology. The Tate conjecture makes a similar assertion with ℓ-adic cohomology replacing Betti cohomology. One of the difficulties with these conjectures is that the predictions that they make are often hard to test numerically, even in specific concrete instances. Unlike closely related parts of number theory (a case in point being the Birch and Swinnerton-Dyer conjecture) the study of algebraic cycles has therefore not been as strongly affected by the growth of the experimental and computational community as it perhaps could be. In this lecture, I will describe some numerical experiments that are designed to "test" the Hodge and Tate conjectures for certain varieties (of arbitrarily large dimension) which arise from elliptic curves with complex multiplication and theta series of CM Hecke characters.

G. Hanrot, F. Morain, and E. Thomé (Eds.): ANTS-IX 2010, LNCS 6197, p. 1, 2010.

Curves of Genus 3 with a Group of Automorphisms Isomorphic to S_3

Jean-François Mestre

Centre de Mathématiques de Jussieu Projet Théorie des Nombres
mestre@math.jussieu.fr

In this talk, we construct curves of genus 3 with automorphism group equal to S_3; we give some applications of this construction to the problem of optimal curves, i.e. of curves over a finite field $\mathbb{F}_q$ having a number of points equal to the Serre-Weil bound M_q; in particular, we prove that there exists infinitely many fields $\mathbb{F}_{3^n}$ having optimal curves; we prove also that there exists an integer C such that, for any finite field $\mathbb{F}_{7^n}$, there exists a curve of genus 3 defined over having at least $M_q - C$ points.

G. Hanrot, F. Morain, and E. Thomé (Eds.): ANTS-IX 2010, LNCS 6197, p. 2, 2010.

Learning with Errors over Rings

Oded Regev

Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel
odedr@post.tau.ac.il

The "learning with errors" (LWE) problem is to distinguish random linear equations, which have been perturbed by a small amount of noise, from truly uniform ones. The problem has been shown to be as hard as worst-case lattice problems, and in recent years it has served as the foundation for a plethora of cryptographic applications.

Unfortunately, these applications are rather inefficient due to an inherent quadratic overhead in the use of LWE. After a short introduction to the area, we will discuss recent work on making LWE and its applications truly efficient by exploiting extra algebraic structure. Namely, we will define the ring-LWE problem, and prove that it too enjoys very strong hardness guarantees.

Based on joint work with Vadim Lyubashevsky and Chris Peikert.

G. Hanrot, F. Morain, and E. Thomé (Eds.): ANTS-IX 2010, LNCS 6197, p. 3, 2010.

Lattices and Spherical Designs

Gabriele Nebe

Lehrstuhl D für Mathematik, RWTH Aachen University, Germany
nebe@math.rwth-aachen.de

A **lattice** is a finitely generated discrete subgroup of Euclidean space. Lattices are an important algorithmic tool in number theory, integral representation theory, geometry, information theory, cryptography, crystallography and have various other applications within mathematics and beyond. Any lattice has only finitely many vectors of a given length, they form the **layers** of the lattice, which are finite subsets of spheres in the underlying Euclidean space.

A **spherical design** of strength t is a finite set $X \neq \emptyset$ in the Euclidean sphere for which the mean value $\frac{1}{|X|}\sum_{x\in X} f(x)$ equals the integral of f over the sphere for all polynomials f of degree up to t. This condition is equivalent to $\sum_{x\in X} f(x) = 0$ for all non-constant harmonic polynomials of degree $\leq t$. Spherical designs hence consist of well distributed points on a sphere and are relevant for numerical integration, in information theory, geometry, statistics and have applications for instance in medicine.

Boris Venkov combined these two concepts in a very fruitful way that allows to use lattices to classify spherical designs and to use designs for finding good lattices. An introduction to this subject as well as some applications are given in "Réseaux euclidiens, designs sphériques et formes modulaires", Enseignement Math., Geneva, 2001. There Venkov introduces the notion of a **strongly perfect lattice**, which is a lattice whose minimal vectors form a spherical 4-design. Using the characterization by Korkine, Voronoi and Zolotarev one shows that strongly perfect lattices realise local maxima of the sphere packing density function on the space of all similarity classes of n-dimensional lattices (in fact in the space of all periodic packings as proved by Schürmann). All local maxima of this function are known up to dimension 8. In dimension 8 Dutour, Schürmann, Vallentin and Riener proved that there are 2408 local maxima. The densest lattice sphere packings are known up to dimension 8 and, thanks to recent results by Elkies and Kumar, in dimension 24, where the Leech lattice is the densest lattice.

Combining number theory and geometry with combinatorial methods allows classify strongly perfect lattices, where a full classification up to dimension 12 is obtained in joined work with Venkov. With one exception all known strongly perfect lattices Λ have the additional property that also the dual lattice Λ^* is strongly perfect. Such lattices are called **dual strongly perfect**, the classification of dual strongly perfect lattices in small dimension has been completed in dimension 14 and is an ongoing PhD project by Elisabeth Nossek in Aachen.

There are two general approaches to study and construct strongly perfect lattices: by modular forms and by invariant theory of finite groups. Both concepts usually allow to show that all non-empty layers of the lattice form spherical

G. Hanrot, F. Morain, and E. Thomé (Eds.): ANTS-IX 2010, LNCS 6197, pp. 4–5, 2010.

4-designs. Such lattices are called **universally perfect** and play a role in Riemannian geometry. If Λ is a universally perfect lattice then the torus $\mathbb{R}^n/\Lambda^*$ defined by the dual lattice Λ^* provides a strict local minimum of the height function on the set of all n-dimensional flat tori. R. Coulangeon also shows that universally perfect lattices Λ achieve local minima of Epstein's zeta function, they are so called ζ-extreme lattices. The question to find ζ-extreme lattices has a long history going back to Sobolev's work on numerical integration and to work of Deloné. Universally perfect lattices are dual strongly perfect.

The relation with modular forms arises, because the condition that the minimal vectors of the lattice form a 4-design means the annihilation of certain coefficients in its theta series with harmonic coefficients. In this way one can prove the strong perfectness of many extremal lattices of small level. For example there are more than 10^6 even unimodular lattices without roots in dimension 32 (by work of Oliver King) and the theory of modular forms shows that all of them are universally perfect; this is the only known method to prove that all these lattices are locally densest lattices.

If a lattice Λ has a big automorphism group $G := \mathrm{Aut}(\Lambda)$ which has no invariant harmonic polynomials of degree 2 and 4, a condition easily expressed in terms of the character of $G \leq O(n)$, then Λ is universally perfect. There are many interesting lattices such as the Barnes-Wall lattices, the 248-dimensional Thompson-Smith lattice and others which are strongly perfect by this reason. Tiep and others used representation theory to classify certain matrix groups G for which all orbits form spherical 4-designs.

On the other hand lattices are an important tool to find and classify good spherical designs. Fixing the strength t and the dimension n, one tries to find spherical t-designs $X \subset S^{n-1}$ of minimal possible cardinality. If $t = 2m$ is even, then

$$|X| \geq \binom{n-1+m}{m} + \binom{n-2+m}{m-1}$$

and if $t = 2m+1$ is odd then

$$|X| \geq 2\binom{n-1+m}{m}.$$

A t-design X for which equality holds is called a **tight** t-design.

Tight t-designs in $\mathbb{R}^n$ with $n \geq 3$ are very rare. Bannai has shown that such tight designs only exist if $t \leq 5$ and $t = 7, 11$. The tight t-designs with $t = 1, 2, 3$ as well as $t = 11$ are completely classified whereas their classification for $t = 4, 5, 7$ is still an open problem. It is conjectured that there are just seven tight t-designs of dimension $n \geq 3$ and strength $4, 5, 7$, namely in dimensions 6,22 (t=4), 3,7,23 (t=5) respectively 8,23 (t=7); each of these is known to be unique.

One possible approach to prove that there are no further tight designs X is to investigate the Euclidean lattice Λ generated by X and to obtain properties of Λ (such as its determinant or its minimum) from the design properties of X and then prove the non existence of such a lattice Λ. This strategy has been successfully applied by Bannai, Munemasa and Venkov to show that there are no further tight designs up to dimension 103.

Fixed Points for Discrete Logarithms*

Mariana Levin[1], Carl Pomerance[2], and K. Soundararajan[3]

[1] Graduate Group in Science and Mathematics Education
University of California
Berkeley, CA 94720, USA
levin@berkeley.edu
[2] Department of Mathematics
Dartmouth College
Hanover, NH 03755, USA
carl.pomerance@dartmouth.edu
[3] Department of Mathematics
Stanford University
Stanford, CA 94305, USA
ksound@math.stanford.edu

Abstract. We establish a conjecture of Brizolis that for every prime $p > 3$ there is a primitive root g and an integer x in the interval $[1, p-1]$ with $\log_g x = x$. Here, $\log_g$ is the discrete logarithm function to the base g for the cyclic group $(\mathbb{Z}/p\mathbb{Z})^\times$. Tools include a numerically explicit "smoothed" version of the Pólya–Vinogradov inequality for the sum of values of a Dirichlet character on an interval, a simple lower bound sieve, and an exhaustive search over small cases.

1 Introduction

If g is an element in a group G and $t \in \langle g \rangle$, there is some integer n with $g^n = t$. Finding a valid choice for n is known as the discrete logarithm problem. Note that if g has finite order m, then n is actually a residue class modulo m. We write

$$\log_g t = n \text{ (or } \log_g t \equiv n \pmod{m})$$

in analogy to usual logarithmic notation. Thus, the problem in the title of this paper does not seem to make good sense, since if $\log_g x = x$, then the first x is a member of the group $\langle g \rangle$ and the second x is either an integer or a residue class modulo m. However, sense is made of the equation through the traditional conflation of members of the ring $\mathbb{Z}/k\mathbb{Z}$ with least nonnegative members of residue classes.

* The work for this paper was begun at Bell Laboratories in 2001 while the first author was a summer student working with the second author. A version of this work was presented as the 2003 Master's Thesis of the first author at U. C. Berkeley, see [3]. The second author was supported in part by NSF grant DMS-0703850. The third author was supported in part by NSF grant DMS-0500711.

G. Hanrot, F. Morain, and E. Thomé (Eds.): ANTS-IX 2010, LNCS 6197, pp. 6–15, 2010.

In particular, suppose $G = (\mathbb{Z}/p\mathbb{Z})^\times$, where p is a prime number. This is known to be a cyclic group of order $p-1$. Suppose g is a cyclic generator of this group, known as a primitive root for p. A fixed point for the discrete logarithm modulo p to the base g is then an integer x in the interval $[1, p-1]$ such that $\log_g x = x$, that is, $g^x \equiv x \pmod{p}$. (Note that if x is not restricted to the interval $[1, p-1]$ it is easy to find fixed points. Namely, if x is a solution to the Chinese remainder problem $x \equiv 1 \pmod{p-1}$, $x \equiv g \pmod{p}$, then $g^x \equiv x \pmod{p}$.)

Brizolis (see Guy [6, Section F9]) made the conjecture that for every prime $p > 3$ there is a primitive root g and an integer x in $[1, p-1]$ with $\log_g x = x$, that is, $g^x \equiv x \pmod{p}$. In this paper we prove this conjecture in a somewhat stronger form. Brizolis had noticed that if there is a primitive root x for p with x in $[1, p-1]$ and $\gcd(x, p-1) = 1$, then with y the multiplicative inverse of x modulo $p-1$ and $g = x^y$, we would have that g is a primitive root for p as well, and

$$g^x \equiv x^{xy} \equiv x \pmod{p},$$

that is, there is a solution to the fixed point problem. We shall prove then the stronger result that for each prime $p > 3$ there is a primitive root x for p in $[1, p-1]$ that is coprime to $p-1$.

Several authors have shown that the Brizolis property holds for all sufficiently large primes p. In particular, Zhang [12] showed the strong conjecture holds for all sufficiently large primes p, but did not give an estimate of what "sufficiently large" is. Cobeli and Zaharescu [4] also showed that the strong conjecture holds for sufficiently large primes p, and gave the details that it holds for all $p > 10^{2070}$, but they indicated that their method would support a bound around 10^{50}.

Our method is similar to that of Zhang, who used the Pólya–Vinogradov inequality for character sums on an interval. Here we introduce a numerically explicit "smoothed" version of this inequality, see §2. In addition, we combine the traditional character-sum approach with a simple lower bound sieve. There is still some need for direct calculation for smaller values of p, which are easily handled by a short Mathematica program. In particular, we directly verified the strong conjecture for each prime $p < 1.25 \cdot 10^9$.

We mention the article by Holden and Moree [8], which considers some related problems. The total number of solutions to $g^x \equiv x \pmod{p}$ as p runs up to some high bound N, where either g is restricted to be a primitive root, and where it is not so restricted, is considered in Bourgain, Konyagin, and Shparlinski [2].

The smoothed version of the Pólya–Vinogradov inequality that we introduce in the next section is quite simple and the proof is routine, so it may be known to others. We have found it to be quite useful numerically; we hope it will find applications in "closing the gap" in other problems where character sums arise.

Some notation: $\omega(n)$ denotes the number of distinct prime divisors of n.

2 A "Smoothed" Pólya–Vinogradov Inequality

Let χ be a non-principal Dirichlet character to the modulus q. The Pólya–Vinogradov inequality (independently discovered by Pólya and Vinogradov in 1918) asserts that there is a universal constant c such that

$$\left| \sum_{M \le a \le M+N} \chi(a) \right| \le c\sqrt{q} \log q \tag{1}$$

for any choice of numbers M, N.

Let $N(p)$ denote the numer of primitive roots g for p with $g \in [1, p-1]$ and $\gcd(g, p-1) = 1$. Using (1) one can show (see Zhang [12] and Campbell [3]) that

$$N(p) = \frac{\varphi(p-1)^2}{p-1} + O(p^{1/2+\epsilon}),$$

for every fixed $\epsilon > 0$, and so $N(p) > 0$ for all sufficiently large p. The aim of this paper is to close the gap and find the complete set of primes p with $N(p) > 0$. Towards this end it would be useful to have a numerically explicit version of (1). In [3], the theorem of Bachman and Rachakonda [1] was used (plus a small unpublished improvement on a secondary term in their inequality due to the second author of the present paper). Recently, elaborating on the work in an early paper of Landau [10], plus an idea of Bateman as mentioned in Hildebrand [7], the second author in [11] proved a stronger numerically explicit version of (1). Using this simplifies the approach in [3]. However, we have found a way to simplify even further by using a "smoothed" version of (1). In this section we prove the following theorem.

Theorem 1. *Let χ be a primitive Dirichlet character to the modulus $q > 1$ and let M, N be real numbers with $0 < N \le q$. Then*

$$\left| \sum_{M \le a \le M+2N} \chi(a) \left(1 - \left| \frac{a-M}{N} - 1 \right| \right) \right| \le \sqrt{q} - \frac{N}{\sqrt{q}}.$$

Proof. We use Poisson summation, see [9, §4.3]. Let

$$H(t) = \max\{0, 1 - |t|\}.$$

We wish to estimate $|S|$, where

$$S := \sum_{a \in \mathbb{Z}} \chi(a) H\left(\frac{a-M}{N} - 1 \right).$$

Towards this end we use the identity

$$\chi(a) = \frac{1}{\tau(\bar{\chi})} \sum_{j=0}^{q-1} \bar{\chi}(j) e(aj/q),$$

where $\tau(\bar{\chi})$ is the Gauss sum for $\bar{\chi}$ and $e(x) := e^{2\pi i x}$. Thus,

$$S = \frac{1}{\tau(\bar{\chi})} \sum_{j=0}^{q-1} \bar{\chi}(j) \sum_{a \in \mathbb{Z}} e(aj/q) H\left(\frac{a-M}{N} - 1\right).$$

The Fourier transform of H is

$$\hat{H}(s) = \int_{-\infty}^{\infty} H(t) e(-st)\, dt = \frac{1-\cos 2\pi s}{2\pi^2 s^2} \text{ when } s \neq 0,\ \hat{H}(0) = 1,$$

which is nonnegative for s real. By a change of variables in the integral, we see that the Fourier transform of $e(jt/q)H((t-M)/N-1)$ is

$$Ne\big(-(M+N)(s-j/q)\big)\hat{H}\big((s-j/q)N\big).$$

Hence, by Poisson summation, we have

$$S = \frac{N}{\tau(\bar{\chi})} \sum_{j=0}^{q-1} \bar{\chi}(j) \sum_{n \in \mathbb{Z}} e\big(-(M+N)(n-j/q)\big)\hat{H}\big((n-j/q)N\big).$$

Estimating trivially (that is, taking the absolute value of each term) and using $\hat{H}$ nonnegative and $\chi(0) = 0$, we have

$$|S| \leq \frac{N}{\sqrt{q}} \sum_{j=1}^{q-1} \sum_{n \in \mathbb{Z}} \hat{H}\big((n-j/q)N\big) = \frac{N}{\sqrt{q}} \sum_{k \in \mathbb{Z} \setminus q\mathbb{Z}} \hat{H}\left(\frac{kN}{q}\right).$$

Since $(N/q)\hat{H}(sN/q)$ is the Fourier transform of $H(qt/N)$, from the last calculation we have

$$\begin{aligned} |S| &\leq \sqrt{q} \sum_{k \in \mathbb{Z} \setminus q\mathbb{Z}} \frac{N}{q} \hat{H}\left(\frac{kN}{q}\right) \leq \sqrt{q}\left(-\frac{N}{q}\hat{H}(0) + \sum_{k \in \mathbb{Z}} \frac{N}{q}\hat{H}\left(\frac{kN}{q}\right)\right) \\ &= \sqrt{q}\left(-\frac{N}{q} + \sum_{l \in \mathbb{Z}} H\left(\frac{ql}{N}\right)\right) = -\frac{N}{\sqrt{q}} + \sqrt{q}H(0) = \sqrt{q} - \frac{N}{\sqrt{q}}, \end{aligned}$$

by another appeal to Poisson summation and the definition of H. This completes the proof of the theorem.

In our application we will need a version of Theorem 1 with the variable a satisfying a coprimality condition. We deduce such a result below.

Corollary 2. *Let k be a square-free integer and let χ be a primitive character to the modulus $q > 1$. For $0 < N \leq q$, we have*

$$\left| \sum_{\substack{0 \leq a \leq 2N \\ (a,k)=1}} \chi(a)\left(1 - \left|\frac{a}{N} - 1\right|\right) \right| \leq \begin{cases} 2^{\omega(k)}\sqrt{q} & \text{always} \\ 2^{\omega(k)-1}\sqrt{q} & \text{if } k \text{ is even.} \end{cases}$$

Proof. Since $\sum_{d|(k,a)} \mu(d)$ gives 1 if $(a,k) = 1$ and 0 otherwise, the sum in question equals

$$\sum_{d|k} \mu(d)\chi(d) \sum_{a \le 2N/d} \chi(a)\Big(1 - \Big|\frac{ad}{N} - 1\Big|\Big)$$

and using Theorem 1 this is bounded in size by $2^{\omega(k)}\sqrt{q}$ as desired. If (k,q) is even, then $\chi(d) = 0$ for even divisors d of k, so that we achieve the bound $2^{\omega(k)-1}\sqrt{q}$, again as desired. Suppose now that k is even and q is odd. For each odd divisor d of k, we group together the contribution from d and $2d$, and so we may write the sum in question as

$$\sum_{d|k/2} \mu(d)\chi(d) \sum_{\substack{a \le 2N/d \\ a \text{ odd}}} \chi(a)\Big(1 - \Big|\frac{ad}{N} - 1\Big|\Big).$$

We replace a in the inner sum by $q + a$, and since q is now odd, the condition that a is odd may be replaced with the condition that $q + a = 2b$ is even. Thus, the above sum becomes

$$\sum_{d|k/2} \mu(d)\chi(d)\chi(2) \sum_{q/2 \le b \le q/2 + N/d} \chi(b)\left(1 - \left|\frac{2d(b - q/2)}{N} - 1\right|\right),$$

and appealing again to Theorem 1 we obtain the Corollary in this case.

Though we will not need it for our proof, we record the following corollary of Theorem 1.

Corollary 3. *Let χ be a primitive Dirichlet character to the modulus $q > 1$ and let M, N be real numbers with $N > 0$. Then, with θ the fractional part of N/q,*

$$\left|\sum_{M \le a \le M+2N} \chi(a)\left(1 - \left|\frac{a - M}{N} - 1\right|\right)\right| \le \frac{q^{3/2}}{N}\theta(1-\theta).$$

3 A Criterion for the Brizolis Property

Let us write the largest square-free divisor of $p - 1$ as uv where u and v will be chosen later. We shall assume that u is even, and have in mind the situation that u is composed of the small prime factors of $p - 1$, and that v is composed of the large prime factors; we also allow for the possibility that $v = 1$. For the rest of the paper, the letter ℓ will denote a prime number.

Let $\mathcal{S}$ denote the set of primitive roots in $[1, p-1]$ that are coprime to $p - 1$. Thus, an integer $g \in [1, p-1]$ is in $\mathcal{S}$ if and only if for each prime $\ell \mid p - 1$ we have both $\ell \nmid g$ and g is not an ℓ-th power (mod p). Let $\mathcal{S}_1$ denote the set of integers in $[1, p-1]$ that are coprime to u and which are not equal to an ℓ-th power (mod p) for any prime ℓ dividing u. Let $\mathcal{S}_2$ denote the set of integers in $\mathcal{S}_1$ which are divisible by some prime ℓ which divides v. Let $\mathcal{S}_3$ denote the set

of integers in $\mathcal{S}_1$ which equal an ℓ-th power (mod p) for some prime ℓ dividing v. Now $\mathcal{S} \subset \mathcal{S}_1$, and the elements in $\mathcal{S}_1$ that are not in $\mathcal{S}$ are precisely those that, for some prime $\ell \mid v$, are either divisible by ℓ or are an ℓ-th power (mod p). Thus, $\mathcal{S} = \mathcal{S}_1 \backslash (\mathcal{S}_2 \cup \mathcal{S}_3)$. We seek a positive lower bound for

$$N := \sum_{g \in \mathcal{S}} \Big(1 - \Big|\frac{2g}{p-1} - 1\Big|\Big),$$

since if $N > 0$, then $\mathcal{S} \neq \emptyset$. By our observation above we have

$$N \geq N_1 - N_2 - N_3,$$

where, for $j = 1, 2, 3$,

$$N_j = \sum_{g \in \mathcal{S}_j} \Big(1 - \Big|\frac{2g}{p-1} - 1\Big|\Big).$$

If d is a square-free divisor of $p-1$ and g is an integer in $[1, p-1]$, let $C_d(g)$ be 1 if g is a d-th power (mod p) and 0 otherwise. Thus,

$$\begin{aligned} C_d(g) &= \prod_{\ell \mid d} C_\ell(g) = \prod_{\ell \mid d} \frac{1}{\ell} \sum_{\chi^\ell = \chi_0} \chi(g) \\ &= \frac{1}{d} \prod_{\ell \mid d} \left(1 + \sum_{\chi \text{ of order } \ell} \chi(g)\right) = \frac{1}{d} \sum_{m \mid d} \sum_{\chi \text{ of order } m} \chi(g). \end{aligned}$$

Note that

$$\sum_{d \mid u} \mu(d) C_d(g)$$

is 1 if, for each $\ell \mid u$, g is *not* an ℓ-th power (mod p), and is 0 otherwise. By the above calculation, this expression is

$$\sum_{d \mid u} \frac{\mu(d)}{d} \sum_{m \mid d} \sum_{\chi \text{ of order } m} \chi(g) = \sum_{m \mid u} \sum_{\chi \text{ of order } m} \chi(g) \sum_{n \mid u/m} \frac{\mu(nm)}{nm}.$$

The inner sum here is $(\varphi(u)/u)\mu(m)/\varphi(m)$, so that

$$N_1 = \frac{\varphi(u)}{u} \sum_{\substack{1 \leq g \leq p-1 \\ (g,u)=1}} \Big(1 - \Big|\frac{2g}{p-1} - 1\Big|\Big) \sum_{m \mid u} \frac{\mu(m)}{\varphi(m)} \sum_{\chi \text{ of order } m} \chi(g). \tag{2}$$

Let $m \mid u$ with $m > 1$. Using Corollary 2, the terms above contribute an amount bounded in magnitude by

$$\frac{\varphi(u)}{u} 2^{\omega(u)-1} \sqrt{p},$$

so the total contribution over all $m \mid u$ with $m > 1$ has magnitude at most

$$\frac{\varphi(u)}{u}\left(2^{\omega(u)}-1\right)2^{\omega(u)-1}\sqrt{p}.$$

The sum over g in (2) with $m = 1$ (and so $\chi = \chi_0$) is

$$\frac{\varphi(u)}{u}\sum_{\substack{1\le g\le p-1\\(g,u)=1}}\left(1-\left|\frac{2g}{p-1}-1\right|\right)=\frac{\varphi(u)}{u}\sum_{d\mid u}\mu(d)\sum_{h\le (p-1)/d}\left(1-\left|\frac{2dh}{p-1}-1\right|\right).$$

The inner sum over h can be evaluated explicitly: it equals $(p-1)/(2d)$ if $(p-1)/d$ is even, and it equals $(p-1)/(2d) - d/(2(p-1))$ if $(p-1)/d$ is odd. It follows that the contribution when $m = 1$ is

$$\left(\frac{\varphi(u)}{u}\right)^2\frac{p-1}{2}-\frac{\varphi(u)}{u}\frac{1}{2(p-1)}\sum_{\substack{d\mid u\\(p-1)/d \text{ odd}}}d\mu(d)$$
$$\ge\left(\frac{\varphi(u)}{u}\right)^2\frac{p-1}{2}-\frac{\varphi(u)^2}{u(p-1)}\ge\left(\frac{\varphi(u)}{u}\right)^2\frac{p}{2}-\frac{\varphi(u)}{u}.$$

We conclude that

$$N_1\ge\left(\frac{\varphi(u)}{u}\right)^2\frac{p}{2}-\frac{\varphi(u)}{u}-\frac{\varphi(u)}{u}\left(2^{\omega(u)}-1\right)2^{\omega(u)-1}\sqrt{p}$$
$$>\left(\frac{\varphi(u)}{u}\right)^2\frac{p}{2}-\frac{\varphi(u)}{2u}4^{\omega(u)}\sqrt{p}.$$

Next we turn to N_2. Since an element in $\mathcal{S}_2$ must be divisible by some prime $\ell|v$ we have that

$$N_2\le\sum_{\ell\mid v}\sum_{\substack{h\le(p-1)/\ell\\(h,u)=1}}\left(1-\left|\frac{2h\ell}{p-1}-1\right|\right)\frac{\varphi(u)}{u}\sum_{m\mid u}\frac{\mu(m)}{\varphi(m)}\sum_{\chi \text{ of order } m}\chi(h\ell).$$

If $v = 1$, then $N_2 = 0$, so assume $v > 1$. The terms with $m > 1$ contribute, using Corollary 2, an amount bounded in size by

$$\frac{\varphi(u)}{u}\omega(v)\left(2^{\omega(u)}-1\right)2^{\omega(u)-1}\sqrt{p}.$$

The main term $m = 1$ above contributes (arguing as in our evaluation of the main term for N_1 above)

$$\frac{\varphi(u)}{u}\sum_{\ell\mid v}\sum_{\substack{h\le(p-1)/\ell\\(h,u)=1}}\left(1-\left|\frac{2h\ell}{p-1}-1\right|\right)\le\left(\frac{\varphi(u)}{u}\right)^2\sum_{\ell\mid v}\left(\frac{p-1}{2\ell}+\frac{\ell}{v}\right).$$

Since $\sum_{\ell|v} \ell \le v$, and using $v > 1$, we conclude that

$$N_2 \le \left(\frac{\varphi(u)}{u}\right)^2 \frac{p-1}{2} \sum_{\ell|v} \frac{1}{\ell} + \left(\frac{\varphi(u)}{u}\right)^2 + \frac{\varphi(u)}{u}\omega(v)\left(2^{\omega(u)}-1\right)2^{\omega(u)-1}\sqrt{p}$$
$$\le \left(\frac{\varphi(u)}{u}\right)^2 \frac{p}{2} \sum_{\ell|v} \frac{1}{\ell} + \frac{\varphi(u)}{2u} 4^{\omega(u)}\omega(v)\sqrt{p}.$$

Lastly we consider N_3. An element g of $\mathcal{S}_3$ must be an ℓ-th power for some prime $\ell|v$, and the indicator function for this condition is $\frac{1}{\ell}\sum_{\psi^\ell=\chi_0}\psi(g)$, as seen above. Therefore we have that N_3 is at most

$$\sum_{\ell|v} \sum_{\substack{g\le p-1\\(g,u)=1}} \Big(1-\Big|\frac{2g}{p-1}-1\Big|\Big)\Big(\frac{\varphi(u)}{u}\sum_{m|u}\frac{\mu(m)}{\varphi(m)} \sum_{\chi \text{ of order } m} \chi(g)\Big)\Big(\frac{1}{\ell}\sum_{\psi^\ell=\chi_0}\psi(g)\Big).$$

Appealing to Corollary 2 for the terms above with $\chi\psi \ne \chi_0$ we find that the contribution of such terms is bounded in magnitude by

$$\frac{\varphi(u)}{u} 2^{2\omega(u)-1}\omega(v)\sqrt{p}.$$

The main term $\chi = \psi = \chi_0$ gives

$$\frac{\varphi(u)}{u}\sum_{\ell|v}\frac{1}{\ell}\sum_{\substack{g\le p-1\\(g,u)=1}}\Big(1-\Big|\frac{2g}{p-1}-1\Big|\Big) \le \frac{\varphi(u)}{u}\Big(\frac{\varphi(u)}{u}\frac{p-1}{2}+\frac{\varphi(u)}{p-1}\Big)\sum_{\ell|v}\frac{1}{\ell}$$
$$= \left(\frac{\varphi(u)}{u}\right)^2\left(\frac{p-1}{2}+\frac{1}{v}\right)\sum_{\ell|v}\frac{1}{\ell} \le \left(\frac{\varphi(u)}{u}\right)^2\frac{p}{2}\sum_{\ell|v}\frac{1}{\ell}.$$

Thus,

$$N_3 \le \left(\frac{\varphi(u)}{u}\right)^2 \frac{p}{2}\sum_{\ell|v}\frac{1}{\ell} + \frac{\varphi(u)}{2u}4^{\omega(u)}\omega(v)\sqrt{p}.$$

Combining these bounds for N_1, N_2 and N_3 we obtain that

$$N \ge \left(\frac{\varphi(u)}{u}\right)^2 \frac{p}{2}\Big(1-2\sum_{\ell|v}\frac{1}{\ell}\Big) - \frac{\varphi(u)}{2u}4^{\omega(u)}(1+2\omega(v))\sqrt{p}.$$

We may conclude as follows: The Brizolis property holds for the prime $p \ge 5$, if we may write the largest square-free divisor of $p-1$ as uv with u even, $\sum_{\ell|v} 1/\ell < 1/2$, and with

$$\sqrt{p} > \frac{4^{\omega(u)}u}{\varphi(u)} \cdot \frac{1+2\omega(v)}{1-2\sum_{\ell|v}1/\ell}. \tag{3}$$

4 Completing the Proof

Our criterion (3) can be used in a straightforward way with $v = 1$ to get an upper bound for possible counterexamples to the Brizolis conjecture. Indeed, after a small calculation (using $4^{\omega(n)} < 1404n^{1/3}$ and $n/\varphi(n) < 2\log\log n$ for n larger than the product of the first eleven primes), it is seen that the Brizolis property holds for all $p > 10^{25}$. It is not pleasant to contemplate checking each prime to this point, so instead we use (3) with $v > 1$.

Suppose $\omega(p-1) = k \geq 10$, and take v to be the product of the six largest primes dividing $p-1$, and u to be the product of the other smaller primes. Since $\omega(p-1) \geq 10$, the primes dividing v are all at least 11, and we have that

$$1 - 2\sum_{\ell \mid v} \frac{1}{\ell} \geq 1 - 2\Big(\frac{1}{11} + \frac{1}{13} + \frac{1}{17} + \frac{1}{19} + \frac{1}{23} + \frac{1}{29}\Big) > 0.28.$$

If p_j denotes the j-th prime, then $4^{\omega(u)}u/\varphi(u) \leq \prod_{j=1}^{k-6}(4p_j/(p_j-1))$, and $p > p-1 \geq \prod_{j=1}^{k} p_j$. So from our criterion (3), if we have

$$\prod_{j=1}^{k} \sqrt{p_j} \geq \frac{13}{0.28} \prod_{j=1}^{k-6} \frac{4p_j}{p_j - 1},$$

then the Brizolis property holds for all p with $\omega(p-1) = k$. We verified that the inequality above holds for $k = 10$. If k is increased by 1 then the LHS of our inequality is increased by a factor of at least $\sqrt{31} > 5$, but the RHS is increased only by a factor of at most $4 \times (11/10) = 4.4$. Thus, the inequality holds for all $k \geq 10$.

Suppose now that $k = \omega(p-1) \leq 9$. If $k \geq 4$, we take u to be the product of the four smallest primes dividing $p-1$, and otherwise, we take u to be the product of all the primes dividing $p-1$. Then v has at most 5 prime factors, and $1 - 2\sum_{\ell \mid v} 1/\ell \geq 1 - 2(1/11 + 1/13 + 1/17 + 1/19 + 1/23) \geq 0.35$. Further $\prod_{p \mid u} 4p/(p-1) \leq \prod_{j=1}^{4} 4p_j/(p_j-1) = 1120$. Our criterion (3) shows that if

$$p \geq \Big(1120 \times \frac{11}{0.35}\Big)^2 = 1{,}239{,}040{,}000,$$

then p satisfies the Brizolis property.

Using the functions Prime[ ] and PrimitiveRoot[ ] in Mathematica, we were able to directly exhibit a primitive root g for each prime $3 < p < 1.25 \cdot 10^9$ with g in $[1, p-1]$ and coprime to $p-1$. Our program runs as follows. The function Prime[ ] allows us to sequentially step through the primes up to our bound. For each prime p returned by Prime[ ], we invoke PrimitiveRoot[p] to find the least positive primitive root r for p. We then sequentially check r^{2k-1} mod p for $k = 1, 2, \ldots$ until we find a value coprime to $p-1$ with $2k-1$ also coprime to $p-1$. The exponent being coprime to $p-1$ guarantees that the power is a primitive root, and the residue being coprime to $p-1$ then guarantees that we

have found a member of $\mathcal{S}$. If no such primitive root exists, this algorithm would not terminate, but it did, thus verifying the Brizolis property for the given range.

There are various small speed-ups that one can use to augment the program. For example, if $r = 2$ is a primitive root and $p \equiv 1 \pmod 4$, then note that $p-2$ is a primitive root coprime to $p-1$, and so work with this prime p is complete. The augmented program ran in about 90 minutes on a Dell workstation.

This completes our proof of the Brizolis conjecture.

Acknowledgment. We thank Richard Crandall for some technical assistance with the Mathematica program and the referees for some helpful comments.

References

1. Bachman, G., Rachakonda, L.: On a problem of Dobrowolski and Williams and the Pólya–Vinogradov inequality. Ramanujan J. 5, 65–71 (2001)
2. Bourgain, J., Konyagin, S.V., Shparlinski, I.E.: Product sets of rationals, multiplicative translates of subgroups in residue rings, and fixed points of the discrete logarithm. Int. Math. Res. Notices, art. ID rnn090, 29 (2008) (Corrigendum: ibid. 2009, No. 16, 3146–3147)
3. Campbell, M.E.: On fixed points for discrete logarithms, Master's Thesis, U. C. Berkeley Department of Mathematics (2003)
4. Cobeli, C., Zaharescu, A.: An exponential congruence with solutions in primitive roots. Rev. Romaine Math. Pures Appl. 44, 15–22 (1999)
5. Crandall, R., Pomerance, C.: Prime numbers: a computational perspective, 2nd edn. Springer, New York (2005)
6. Guy, R.K.: Unsolved problems in number theory. Springer, Berlin (1984)
7. Hildebrand, A.: On the constant in the Pólya–Vinogradov inequality. Canad. Math. Bull. 31, 347–352 (1988)
8. Holden, J., Moree, P.: Some heuristics and results for small cycles of the discrete logarithm. Math. Comp. 75, 419–449 (2006)
9. Iwaniec, H., Kowalski, E.: Analytic number theory. American Math. Soc., Providence (2004)
10. Landau, E.: Abschätzungen von Charaktersummen, Einheiten und Klassenzahlen. Nachrichten Königl. Ges. Wiss. Göttingen, 79–97 (1918)
11. Pomerance, C.: Remarks on the Pólya–Vinogradov inequality (submitted for publication 2010)
12. Zhang, W.-P.: On a problem of Brizolis. Pure Appl. Math. 11(Suppl.), 1–3 (1995) (Chinese. English, Chinese summary)

Explicit Coleman Integration for Hyperelliptic Curves

Jennifer S. Balakrishnan[1], Robert W. Bradshaw[2], and Kiran S. Kedlaya[1]

[1] Massachusetts Institute of Technology, Cambridge, MA 02139, USA
jen@math.mit.edu, kedlaya@mit.edu
[2] University of Washington, Seattle, WA 98195, USA
robertwb@math.washington.edu

Abstract. Coleman's theory of p-adic integration figures prominently in several number-theoretic applications, such as finding torsion and rational points on curves, and computing p-adic regulators in K-theory (including p-adic heights on elliptic curves). We describe an algorithm for computing Coleman integrals on hyperelliptic curves, and its implementation in Sage.

1 Introduction

One of the fundamental difficulties of p-adic analysis is that the totally disconnected topology of p-adic spaces makes it hard to introduce a meaningful form of antidifferentiation. It was originally discovered by Coleman that this problem can be circumvented using the principle of *Frobenius equivariance*. Using this idea, Coleman introduced a p-adic integration theory first on the projective line [9], then (partly jointly with de Shalit) on curves and abelian varieties [10], [8]. Alternative treatments have been given by Besser [3] using methods of p-adic cohomology, and by Berkovich [2] using the nonarchimedean Gel'fand transform.

Although Coleman's construction is in principle quite suitable for machine computation, this had only been implemented previously in the genus 0 case [5]. The purpose of this paper is to present an algorithm for computing single Coleman integrals on hyperelliptic curves of good reduction over $\mathbb{C}_p$ for $p > 2$, based on the third author's algorithm for computing the Frobenius action on the de Rham cohomology of such curves [17]. We also describe an implementation of this algorithm in the Sage computer algebra system.

For context, we indicate some of the many potential applications of explicit Coleman integration. Some of these will be treated, with additional numerical examples, in the first author's upcoming PhD thesis. (Some of these applications will require additional refinements of our implementation; see Section 5.)

- *Torsion points on curves.* Coleman's original application of p-adic integration was to find torsion points on curves of genus greater than 1. This could potentially be made effective and automatic.
- *p-adic heights on curves.* Investigations into p-adic analogues of the conjecture of Birch and Swinnerton-Dyer for Jacobians of hyperelliptic curves

G. Hanrot, F. Morain, and E. Thomé (Eds.): ANTS-IX 2010, LNCS 6197, pp. 16–31, 2010.

require computation of the Coleman-Gross height pairing [11]. This global p-adic height pairing can, in turn, be decomposed into a sum of local height pairings at each prime. In particular, for C a hyperelliptic curve over $\mathbb{Q}_p$ with p a prime of good reduction and for $D_1, D_2 \in \mathrm{Div}^0(C)$ with disjoint support, the Coleman-Gross p-adic height pairing at p is given in terms of the Coleman integral [10]

$$h_p(D_1, D_2) = \int_{D_2} \omega_{D_1},$$

for an appropriately constructed differential ω_{D_1} associated to the divisor D_1. This pairing is effectively computable by work of the first author [1]. Using this work, it should be possible (using ideas of Besser [4]) to add in local heights away from p, and thus compute the Coleman-Gross height pairing on Jacobians of hyperelliptic curves. (In genus 1, one can then compare to an alternate computation based on work of Mazur-Stein-Tate [22] and Harvey [16].)

- *p-adic regulators.* A related topic to the previous one is the computation of p-adic regulators in higher K-theory of arithmetic schemes, which are expected to relate to special values of L-functions. Some computations in genus 0 have been made by Besser and de Jeu [5].
- *Rational points on curves: Chabauty's method.* For C a smooth proper curve over $\mathbb{Z}[\frac{1}{N}]$, the *Chabauty condition* on C is that $\mathrm{rank}\, J(C)\left(\mathbb{Z}\left[\frac{1}{N}\right]\right) < \dim J(C)$, where $J(C)$ denotes the Jacobian of the curve. When the Chabauty condition holds, there exists a 1-form ω on $J(C)^{\mathrm{an}}$ with $\int_0^P \omega = 0$ for all points $P \in J(C)\left(\mathbb{Z}\left[\frac{1}{N}\right]\right)$. We might be able to compute $C(\mathbb{Z}[\frac{1}{N}])$ if we can find all points $P \in C^{\mathrm{an}}$ such that $\int_0^P \omega = 0$. This method has already been used in many cases, by Coleman and many others; see [23] for a survey (circa 2007). To apply Chabauty's method in a typical case, one needs the integral of ω at some point in a residue disc, with which one can find all zeroes of the integral in the residue disc. Several methods are suggested in [23, Remark 8.3] for doing this, including Coleman integration. However, no serious attempt has been made to use numerical Coleman integration in Chabauty's method; it seems likely that it can handle cases where the other methods suggested in [23, Remark 8.3] for finding constants of integration prove to be impractical.
- *Rational points on curves: nonabelian Chabauty.* It may be possible to use (iterated) Coleman integration to find rational points on curves failing the Chabauty condition, using Kim's nonabelian Chabauty method [18]. As a demonstration of the method, Kim [19] gives an explicit double integral which vanishes on the integral points of the minimal regular model of a genus 1 curve over $\mathbb{Q}$ of Mordell-Weil rank 1. The erratum to [19] includes a corrected formula, together with some numerical examples computed using the methods of this paper.
- *p-adic polylogarithms and multiple zeta values.* These have been introduced recently by Furusho [13], but little numerical data exists so far.

2 Coleman's Theory of p-adic Integration

In this section, we recall Coleman's p-adic integration theory (for single integrals only) in the case of curves with good reduction. This theory involves some concepts from rigid analytic geometry which it would be hopeless to introduce in such limited space; some standard references are [6] and [12]. (See also [10, §1].)

Let $\mathbb{C}_p$ be a completed algebraic closure of $\mathbb{Q}_p$, and let $\mathcal{O}$ be the valuation subring of $\mathbb{C}_p$. Choose once and for all a *branch of the p-adic logarithm*, i.e., a homomorphism $\mathrm{Log} : \mathbb{C}_p^\times \to \mathbb{C}_p$ whose restriction to the disc $\{x \in \mathbb{C}_p : |x-1| < 1\}$ is given by the logarithm series $\log(x) = \sum_{i=1}^{\infty} (1-x)^i/i$. (The choice of branch has no effect on the integrals on differentials of the second kind, i.e., everywhere meromorphic differentials with all residues zero.)

We first introduce integrals on discs and annuli within $\mathbb{P}^1$.

Definition 1. *Let I be an open subinterval of $[0, +\infty)$. Let $A(I)$ denote the annulus (or disc) $\{t \in \mathbb{A}^1_{\mathbb{C}_p} : |t| \in I\}$. For $\sum_{i \in \mathbb{Z}} c_i t^i \, dt \in \Omega^1_{A(I)/\mathbb{C}_p}$ and $P, Q \in A(I)$, define*

$$\int_P^Q \sum_{i \in \mathbb{Z}} c_i t^i \, dt = c_{-1} \mathrm{Log}(Q/P) + \sum_{i \neq -1} \frac{c_i}{i+1} (Q^{i+1} - P^{i+1}).$$

This is easily shown not to depend on the choice of the coordinate t.

Remark 2. Note that because of the division by $i + 1$ in the formula for the integral, we are unable to integrate on *closed* discs or annuli.

We next turn to curves of good reduction.

Definition 3. *By a* curve *over $\mathcal{O}$, we will mean a smooth proper connected scheme X over $\mathcal{O}$ of relative dimension* 1*. Equip the function field $K(X)$ with the p-adic absolute value, so that the elements of $K(X)$ of norm at most* 1 *constitute the local ring in X of the generic point of the special fibre $\overline{X}$ of X.*

Let $X_{\mathbb{Q}}$ denote the generic fibre of X as a rigid analytic space. There is a natural specialization map from $X_{\mathbb{Q}}$ to $\overline{X}$; the inverse image of any point of $\overline{X}$ is a subspace of $X_{\mathbb{Q}}$ isomorphic to an open unit disc. We call such a disc a residue disc *of X.*

Definition 4. *Let X be a curve over $\mathcal{O}$. By a* wide open subspace *of $X_{\mathbb{Q}}$, we will mean a rigid analytic subspace of $X_{\mathbb{Q}}$ of the form $\{x \in X_{\mathbb{Q}} : |f(x)| < \lambda\}$ for some $f \in K(X)$ of absolute value* 1 *and some $\lambda > 1$.*

Coleman made the surprising discovery that there is a well-behaved integration theory on wide open subspaces of curves over $\mathcal{O}$, exhibiting no phenomena of path dependence. (Note that one needs to consider wide open subspaces even to integrate differentials which are holomorphic or meromorphic on the entire curve.) In the case of hyperelliptic curves, Coleman's construction of these integrals using Frobenius lifts will be reflected in our technique for computing the integrals. For the general case, see [10, §2], [3, §4], or [2, Theorem 1.6.1].

Theorem 5 (Coleman). *We may assign to each curve X over $\mathcal{O}$ and each wide open subspace W of $X_{\mathbb{Q}}$ a map $\mu_W : \mathrm{Div}^0(W) \times \Omega^1_{W/\mathbb{C}_p} \to \mathbb{C}_p$, subject to the following conditions. (Here $\mathrm{Div}(W)$ denotes the free group on the elements of W, and $\mathrm{Div}^0(W)$ denotes the kernel of the degree map $\deg : \mathrm{Div}(W) \to \mathbb{Z}$ taking each element of W to 1.)*

(a) (Linearity) The map μ_W is linear on $\mathrm{Div}^0(W)$ and $\mathbb{C}_p$-linear on $\Omega^1_{W/\mathbb{C}_p}$.

(b) (Compatibility) For any residue disc D of X and any isomorphism $\psi : W \cap D \to A(I)$ for some interval I, the restriction of μ_W to $\mathrm{Div}^0(W \cap D) \times \Omega^1_{W/\mathbb{C}_p}$ is compatible with Definition 1 via ψ.

(c) (Change of variables) Let X' be another curve over $\mathcal{O}$, let W' be a wide open subspace of X', and let $\psi : W \to W'$ be any morphism of rigid spaces relative to an automorphism of $\mathbb{C}_p$. Then

$$\mu_{W'}(\psi(\cdot), \cdot) = \mu_W(\cdot, \psi^*(\cdot)). \tag{1}$$

(d) (Fundamental theorem of calculus) For any $Q = \sum_i c_i(P_i) \in \mathrm{Div}^0(W)$ and any $f \in \mathcal{O}(W)$, $\mu_W(Q, df) = \sum_i c_i f(P_i)$.

Remark 6. One cannot expect path independence in the case of bad reduction. For instance, an elliptic curve over $\mathbb{C}_p$ with bad reduction admits a Tate uniformization, so its logarithm map has nonzero periods in general. In Berkovich's theory of integration, this occurs because the nonarchimedean analytic space associated to this curve X has nontrivial first homology.

3 Explicit Integrals for Hyperelliptic Curves

We now specialize to the situation where $p > 2$ and X is a genus g hyperelliptic curve over an unramified extension K of $\mathbb{Q}_p$ having good reduction. We will assume in addition that we have been given a model of X of the form $y^2 = f(x)$ such that $\deg f(x) = 2g+1$ and f has no repeated roots modulo p. (This restriction is inherited from [17], where it is used to simplify the reduction procedure. One could reduce to this case after possibly replacing K by a larger unramified extension of $\mathbb{Q}_p$, by performing a linear fractional transformation in x to put one root at infinity, thus reducing the degree from $2g + 2$ to $2g + 1$.) We will distinguish between *Weierstrass* and *non-Weierstrass* residue discs of X, which respectively correspond to Weierstrass and non-Weierstrass points of $\overline{X}$.

To discuss the differentials we will be integrating, we review a core definition from [17]. Let X' be the affine curve obtained by deleting the Weierstrass points from X, and let $A = K[x, y, z]/(y^2 - f(x), yz - 1)$ be the coordinate ring of X'.

Definition 7. *The Monsky-Washnitzer (MW) weak completion of A is the ring $A^\dagger$ consisting of infinite sums of the form*

$$\left\{ \sum_{i=-\infty}^{\infty} \frac{B_i(x)}{y^i},\ B_i(x) \in K[x], \deg B_i \leq 2g \right\},$$

further subject to the condition that $v_p(B_i(x))$ grows faster than a linear function of i as $i \to \pm\infty$. We make a ring out of these using the relation $y^2 = f(x)$.

These functions are holomorphic on wide opens, so we will integrate 1-forms

$$\omega = g(x,y)\frac{dx}{2y}, \quad g(x,y) \in A^\dagger. \tag{2}$$

Note that we only consider 1-forms which are *odd*, i.e., which are negated by the hyperelliptic involution. Even 1-forms can be written in terms of x alone, and so can be integrated directly as in Definition 1. (This last statement would fail if we had taken $A^\dagger$ to be the full p-adic completion of A, rather than the weak completion. This observation is the basis for Monsky-Washnitzer's formal cohomology, which is used in [17].)

Note that the class of allowed forms includes those meromorphic differentials on X whose poles all belong to Weierstrass residue discs. For some applications (e.g., p-adic canonical heights), it is necessary to integrate meromorphic differentials with poles in non-Weierstrass residue discs. These will be discussed in [1].

Note also that for ease of exposition, we describe all of our algorithms as if it were possible to compute exactly in $A^\dagger$. This is not possible for two reasons: the elements of $A^\dagger$ correspond to infinite series, and the coefficients of these series are polynomials with p-adic coefficients. In practice, each computation will be made with suitable p-adic approximations of the truly desired quantities, so one must keep track of how much p-adic precision is needed in these estimates in order for the answers to bear a certain level of p-adic accuracy. We postpone this discussion to § 4.1.

3.1 A Basis for de Rham Cohomology

We first note that any odd differential ω as in (2) can be written uniquely as

$$\omega = df + c_0\omega_0 + \cdots + c_{2g-1}\omega_{2g-1} \tag{3}$$

with $f \in A^\dagger$, $c_i \in K$, and

$$\omega_i = \frac{x^i\,dx}{2y} \qquad (i = 0, \ldots, 2g-1). \tag{4}$$

That is, the ω_i form a basis of the odd part of the de Rham cohomology of $A^\dagger$. The process of putting ω in the form (3), using the relations

$$\begin{aligned} y^2 &= f(x), \\ d(x^i y^j) &= \left(2ix^{i-1}y^{j+1} + jx^i f'(x) y^{j-1}\right)\frac{dx}{2y}, \end{aligned}$$

can be made algorithmic; see [17, §3]. (Briefly, one uses the first relation to reduce high powers of x, and the second to reduce large positive and negative powers of y.) Using properties from Theorem 5 (linearity and the fundamental

theorem of calculus), the integration of ω reduces effectively to the integration of the ω_i.

It may be convenient for some purposes to use a different basis of de Rham cohomology. For instance, the basis $x^i\,dx/2y^3$ $(i = 0, \ldots, 2g-1)$ is *crystalline* (see the erratum to [17]), so Frobenius will act via a matrix with p-adically integral entries.

3.2 Tiny Integrals

We refer to any Coleman integral of the form $\int_P^Q \omega$ in which P, Q lie in the same residue disc (Weierstrass or not) as a *tiny integral*. As an easy first case, we give an algorithm to compute tiny integrals of basis differentials.

Algorithm 8 (Tiny Coleman integrals).
Input: *Points $P, Q \in X(\mathbb{C}_p)$ in the same residue disc (neither equal to the point at infinity) and a basis differential ω_i.*
Output: *The integral $\int_P^Q \omega_i$.*

1. *Construct a linear interpolation from P to Q. For instance, in a non-Weierstrass residue disc, we may take*

$$x(t) = (1-t)x(P) + tx(Q)$$
$$y(t) = \sqrt{f(x(t))},$$

   where $y(t)$ is expanded as a formal power series in t.
2. *Formally integrate the power series in t:*

$$\int_P^Q \omega_i = \int_P^Q x^i \frac{dx}{2y} = \int_0^1 \frac{x(t)^i}{2y(t)} \frac{dx(t)}{dt} dt.$$

Remark 9. One can similarly integrate any ω holomorphic in the residue disc containing P and Q. If ω is only meromorphic in the disc, but has no pole at P or Q, we can first make a polar decomposition, i.e., write ω as a holomorphic differential on the disc plus some terms of the form $c/(t-r)^i$, and integrate the latter terms directly. (If ω is everywhere meromorphic, this is achieved by a partial fractions decomposition.)

3.3 Non-Weierstrass Discs

We next compute integrals of the form $\int_P^Q \omega_i$ in which $P, Q \in X(\mathbb{C}_p)$ lie in distinct non-Weierstrass residue discs. The method of tiny integrals is not available; we instead employ Dwork's principle of analytic continuation along Frobenius, in the form of Kedlaya's algorithm [17] for calculating the action of Frobenius on de Rham cohomology. Note that we calculate the integrals $\int_P^Q \omega_i$ for all i simultaneously. (We modify the presentation in [17] by keeping track of exact differentials, which are irrelevant for computing zeta functions.)

Algorithm 10 (Kedlaya's algorithm).
Input: *The basis differentials* $\{\omega_i\}_{i=0}^{2g-1}$.
Output: *Functions* $f_i \in A^\dagger$ *and a* $2g \times 2g$ *matrix* M *over* K *such that* $\phi^*(\omega_i) = df_i + \sum_{j=0}^{2g-1} M_{ij}\omega_j$ *for a* p*-power lift of Frobenius* ϕ.

1. *Since* K *is an unramified extension of* $\mathbb{Q}_p$*, it carries a unique automorphism* ϕ_K *lifting the Frobenius automorphism* $x \mapsto x^p$ *on its residue field. Extend* ϕ_K *to a Frobenius lift on* $A^\dagger$ *by setting*

$$\begin{aligned}\phi(x) &= x^p,\\ \phi(y) &= y^p\left(1 + \frac{\phi_K(f)(x^p) - f(x)^p}{f(x)^p}\right)^{1/2}\\ &= y^p \sum_{i=0}^{\infty} \binom{1/2}{i} \frac{(\phi_K(f)(x^p) - f(x)^p)^i}{y^{2pi}},\end{aligned}$$

   noting the series converges in $A^\dagger$ *because* $\phi_K(f)(x^p) - f(x)^p$ *has positive valuation. (This choice of* $\phi(y)$ *ensures that* $\phi(y)^2 = \phi(f(x))$*, so that the action on* $A^\dagger$ *is well-defined.*
2. *Use a Newton iteration to compute* $y/\phi(y)$*. Then for* $i = 0, \dots, 2g-1$*, proceed as in* § 3.1 *to write*

$$\phi^*(\omega_i) = px^{pi+p-1}\frac{y}{\phi(y)}\frac{dx}{2y} = df_i + \sum_{j=0}^{2g-1} M_{ij}\omega_j \tag{5}$$

   for some $f_i \in A^\dagger$ *and some* $2g \times 2g$ *matrix* M *over* K.

We may use Algorithm 10 to compute Coleman integrals between endpoints in non-Weierstrass residue discs, as follows. (Note that our recipe is essentially Coleman's construction of the integrals in this case.)

Algorithm 11 (Coleman integration in non-Weierstrass discs).
Input: *The basis differentials* $\{\omega_i\}_{i=0}^{2g-1}$*, points* $P, Q \in X(\mathbb{C}_p)$ *in non-Weierstrass residue discs, and a positive integer* m *such that the residue fields of* P, Q *are contained in* $\mathbb{F}_{p^m}$.
Output: *The integrals* $\{\int_P^Q \omega_i\}_{i=0}^{2g-1}$.

1. *Calculate the action of the* m*-th power of Frobenius on each basis element (see Remark 12):*

$$(\phi^m)^*\omega_i = df_i + \sum_{j=0}^{2g-1} M_{ij}\omega_j. \tag{6}$$

2. *By change of variables (see Remark 13), we obtain*

$$\sum_{j=0}^{2g-1} (M - I)_{ij} \int_P^Q \omega_j = f_i(P) - f_i(Q) - \int_P^{\phi^m(P)} \omega_i - \int_{\phi^m(Q)}^{Q} \omega_i \tag{7}$$

(the fundamental linear system*). As the eigenvalues of the matrix M are algebraic integers of $\mathbb{C}_p$-norm $p^{m/2} \neq 1$ (see [17, §2]), the matrix $M - I$ is invertible, and we may solve (7) to obtain the integrals $\int_P^Q \omega_i$.*

Remark 12. To compute the action of ϕ^m, first perform Algorithm 10 to write

$$\phi^* \omega_i = dg_i + \sum_{j=0}^{2g-1} B_{ij} \omega_j.$$

If we view f, g as column vectors and M, B as matrices, we then have

$$f = \phi^{m-1}(g) + B\phi^{m-2}(g) + \cdots + B\phi_K(B) \cdots \phi_K^{m-2}(B)g$$
$$M = B\phi_K(B) \cdots \phi_K^{m-1}(B).$$

Remark 13. We obtain (7) as follows. By change of variables,

$$\begin{aligned} \int_{\phi^m(P)}^{\phi^m(Q)} \omega_i &= \int_P^Q (\phi^m)^* \omega_i \\ &= \int_P^Q (df_i + \sum_{j=0}^{2g-1} M_{ij} \omega_j) \\ &= f_i(Q) - f_i(P) + \sum_{j=0}^{2g-1} M_{ij} \int_P^Q \omega_j. \end{aligned}$$

Adding $\int_P^{\phi^m(P)} \omega_i + \int_{\phi^m(Q)}^Q \omega_i$ to both sides of this equation yields

$$\int_P^Q \omega_i = \int_P^{\phi^m(P)} \omega_i + \int_{\phi^m(Q)}^Q \omega_i + f_i(Q) - f_i(P) + \sum_{j=0}^{2g-1} M_{ij} \int_P^Q \omega_j,$$

which is equivalent to (7).

Definition 14. *A* Teichmüller point *of $X_{\mathbb{Q}}$ is a point fixed by some power of ϕ. Each non-Weierstrass residue disc contains a unique such point: if $(\overline{x}, \overline{y}) \in \overline{X}$ is a non-Weierstrass point, the Teichmüller point in its residue disc has x-coordinate equal to the usual Teichmüller lift of x. This leaves two choices for the y-coordinate, exactly one of which has the correct reduction modulo p. Note that Teichmüller points are always defined over finite* unramified *extensions of $\mathbb{Q}_p$.*

Remark 15. A variant of Algorithm 11 is to first find the Teichmüller points P', Q' in the residue discs of P, Q, then note that from the fundamental linear system (7), we have

$$\sum_{j=0}^{2g-1} (M - I)_{ij} \int_{P'}^{Q'} \omega_j = f_i(P') - f_i(Q'). \tag{8}$$

From (8), we obtain the integrals $\int_{P'}^{Q'} \omega_i$. Finally, write $\int_P^Q \omega_i - \int_{P'}^{Q'} \omega_i$ as the sum $\int_P^{P'} \omega_i + \int_{Q'}^Q \omega_i$ of tiny integrals.

3.4 Weierstrass Endpoints of Integration

Suppose now that P, Q lie in different residue discs, at least one of which is Weierstrass. Since a differential ω of the form (2) is not meromorphic over Weierstrass residue discs, we cannot always even define $\int_P^Q \omega$, let alone compute it. We will thus assume (to cover most cases arising in applications) that ω is everywhere meromorphic, with no pole at either P or Q. We then make the following observation.

Lemma 16. *Let ω be an odd, everywhere meromorphic differential on X. Choose $P, Q \in X(\mathbb{C}_p)$ which are not poles of ω, with P Weierstrass. Then for ι the hyperelliptic involution, $\int_P^Q \omega = \frac{1}{2}\int_{\iota(Q)}^Q \omega$. In particular, if Q is also a Weierstrass point, then $\int_P^Q \omega = 0$.*

Proof. Let $I := \int_P^Q \omega = \int_P^{\iota(Q)}(-\omega) = \int_{\iota(Q)}^P \omega$. Then by additivity in the endpoints, we have $\int_{\iota(Q)}^Q \omega = 2I$, from which the result follows.

If P belongs to a Weierstrass residue disc while Q does not, we find the Weierstrass point P' in the disc of P, then apply Lemma 16 to write

$$\int_P^Q \omega = \int_P^{P'} \omega + \frac{1}{2}\int_{\iota(Q)}^Q \omega. \tag{9}$$

The first integral on the right side of (9) is tiny, while the second integral involves two points in non-Weierstrass residue discs, and so may be computed as in the previous section. The situation is even better if P, Q both belong to residue discs containing respective Weierstrass points P', Q': in this case, by Lemma 16, $\int_P^Q \omega$ equals the sum $\int_P^{P'} \omega + \int_{Q'}^Q \omega$ of tiny integrals.

Remark 17. Beware that Lemma 16 does not generalize to iterated integrals. For instance, for double integrals, if both integrands are odd, the total integrand is even, so the argument of Lemma 16 tells us nothing. It is thus worth considering alternate approaches for dealing with Weierstrass discs, which may generalize better to the iterated case. We concentrate on the case where P lies in a Weierstrass residue disc but Q does not, as we may reduce to this case by splitting $\int_P^Q \omega = \int_P^R \omega + \int_R^Q \omega$ for some auxiliary point R in a non-Weierstrass residue disc.

In Algorithm 11, the form f_i belongs to $A^\dagger$ and so need not converge at P. However, it does converge at any point R near the boundary of the disc, i.e., in the complement of a certain smaller disc which can be bounded explicitly. We may thus write $\int_P^Q \omega_i = \int_P^R \omega_i + \int_R^Q \omega_i$ for suitable R in the disc of P, to obtain an analogue of the fundamental linear system (7). Similarly, when we write

ω as in (3), we can find R close enough to the boundary of the disc of P so that f converges at R, use (3) to evaluate $\int_R^Q \omega$, then compute $\int_P^R \omega$ as a tiny integral. One defect of this approach is that forcing R to be close to the boundary of the residue disc of P forces R to be defined over a highly ramified extension of $\mathbb{Q}_p$, over which computations are more expensive.

An alternate approach exploits the fact that for P in the infinite residue disc but distinct from the point at infinity, we may compute $\int_P^Q \omega$ directly using Algorithm 11. This works because both the Frobenius lift and the reduction process respect the subring of $A^\dagger$ consisting of functions which are meromorphic at infinity. When P lies in a finite Weierstrass residue disc, we may reduce to the previous case using a change of variables on the x-line to move P to the infinite disc. However, one still must use the approach of the previous paragraph to reduce evaluation of $\int_P^Q \omega$ to evaluation of the $\int_P^Q \omega_i$.

4 Implementation Notes and Precision

We have implemented the above algorithms in Sage [24] for curves defined over $\mathbb{Q}_p$. In doing so, we made the following observations.

4.1 Precision Estimates

For a tiny integral, the precision of the result depends on the truncation of the power series computed. Here is the analysis for a non-Weierstrass disc; the analysis for a Weierstrass disc, using a different local interpolation, is similar. (For points over ramified extensions, one must also account for the ramification index in the bound, but it should be clear from the proof how this is done.)

Proposition 18. *Let $\int_P^Q \omega$ be a tiny integral in a non-Weierstrass residue disc, with P, Q defined over an unramified extension of K and accurate to n digits of precision. Let $(x(t), y(t))$ be the local interpolation between P and Q defined by*

$$x(t) = x(P)(1-t) + x(Q)t = x(P) + t(x(Q) - x(P))$$
$$y(t) = \sqrt{f(x(t))}.$$

Let $\omega = g(x,y)dx$ be a differential of the second kind such that $h(t) = g(x(t), y(t))$ belongs to $\mathcal{O}[[t]]$. If we truncate $h(t)$ modulo t^m, then the computed value of the integral $\int_P^Q \omega$ will be correct to $\min\{n, m+1-\lfloor \log_p(m+1) \rfloor\}$ digits of (absolute) precision.

Proof. Let $t' = t(x(Q) - x(P))$. As P, Q are in the same residue disc and are defined over an unramified extension of K, we have $v_p(x(Q) - x(P)) \geq 1$. If we expand $g(x(t'), y(t')) = \sum_{i=0}^{\infty} c_i (t')^i$, then by hypothesis $c_i \in \mathcal{O}$. Thus

$$\begin{aligned}\int_P^Q \omega &= \int_P^Q g(x,y)dx \\ &= \int_0^1 g(x(t),y(t))dx(t) \\ &= \int_0^{x(Q)-x(P)} g(x(t'),y(t'))dt' \\ &= \int_0^{x(Q)-x(P)} \sum_{i=0}^{\infty} c_i (t')^i dt' \\ &= \sum_{i=0}^{\infty} \frac{c_i}{i+1}(x(Q)-x(P))^{i+1}.\end{aligned}$$

The effect of omitting $c_i(t')^i$ from the expansion of $g(x(t'),y(t'))$ for some $i \geq m$ is to change the final sum by a quantity of valuation at least $i+1-\lfloor \log_p(i+1) \rfloor \geq m+1-\lfloor \log_p(m+1) \rfloor$. The effect of the ambiguity in P and Q is that the computed value of $(x(Q)-x(P))^{i+1}$ differs from the true value by a quantity of valuation at least $i+1-\lfloor \log_p(i+1) \rfloor + n - 1 \geq n$.

For Coleman integrals between different residue discs, which we may assume are non-Weierstrass thanks to § 3.4, one must first account for the precision loss in Algorithm 10. According to [17, Lemmas 2,3] and the erratum to [17] (or [15]), working to precision p^N in Algorithm 10 produces the f_i, M_{ij} accurately modulo p^{N-n} for $n = 1 + \lfloor \log_p \max\{N, 2g+1\} \rfloor$.

We must then take into account the objects involved in the linear system (7), as follows.

Proposition 19. *Let $\int_P^Q \omega$ be a Coleman integral, with ω a differential of the second kind and with P, Q in non-Weierstrass residue discs, defined over an unramified extension of $\mathbb{Q}_p$, and accurate to n digits of precision. Let* Frob *be the matrix of the action of Frobenius on the basis differentials. Set $B = \mathrm{Frob}^t - I$, and let $m = v_p(\det(B))$. Then the computed value of the integral $\int_P^Q \omega$ will be accurate to $n - \max\{m, \lfloor \log_p n \rfloor\}$ digits of precision.*

Proof. By the linear system (7), the Coleman integral is expressed in terms of tiny integrals, integrals of exact forms evaluated at points, and a matrix inversion. Suppose that the entries of $B = \mathrm{Frob}^t - I$ are computed to precision n. Then taking B^{-1}, we have to divide by $\det(B)$, which lowers the precision by $m = v_p(\det(B))$. By Proposition 18, computing tiny integrals (with the series expansions truncated modulo t^{n-1}) gives a result precise up to $n - \lfloor \log_p n \rfloor$ digits. Thus the value of the integral $\int_P^Q \omega$ will be correct to $n - \max\{m, \lfloor \log_p n \rfloor\}$ digits of precision.

4.2 Complexity Analysis

We assume that asymptotically fast integer and polynomial multiplication algorithms are used; specifically addition, subtraction, multiplication, and division take $\widetilde{O}(\log N)$ bit operations in $\mathbb{Z}/N\mathbb{Z}$ and $\widetilde{O}(n)$ basering operations in

$R[x]/x^nR[x]$. In particular, this allows arithmetic operations in $\mathbb{Q}_p$ to n (relative) digits of precision, hereafter called field operations, in time $\widetilde{O}(n \log p)$. Using Newton iteration, both square roots and the Teichmüller character can be computed to n digits of precision using $\widetilde{O}(\log n)$ arithmetic operations. (We again consider only points in non-Weierstrass discs defined over unramified fields.)

Proposition 20. *Let $\int_P^Q \omega$ be a Coleman integral on a curve of genus g over $\mathbb{Q}_p$, with $\omega = df_\omega + \sum_{i=1}^{2g-i} c_i\omega_i$ a differential of the second kind and with P, Q in non-Weierstrass residue discs, defined over $\mathbb{Q}_p$, and accurate to n digits of precision. Let* Frob *be the matrix of the action of Frobenius on the basis differentials, and let $m = v_p(\det(\text{Frob}^t - I))$. Let $F(n)$ be the running time of evaluating f_ω at P and Q to n digits of precision. The value of the integral $\int_P^Q \omega$ can be computed to $n - \max\{m, \lfloor \log_p n \rfloor\}$ digits of precision in time $F(n) + \widetilde{O}(pn^2g^2 + g^3 n \log p)$. (Over a degree N unramified extension of $\mathbb{Q}_p$, the analysis is the same with the runtime multiplied by a factor of N.)*

Proof. An essential input to the algorithm is the matrix of the action of Frobenius, which can be computed by Kedlaya's algorithm to n digits of precision in running time $\widetilde{O}(pn^2g^2)$. Inverting the resulting matrix can be (naïvely) done with $O(g^3)$ arithmetic operations in $\mathbb{Q}_p$. It remains to be shown that no other step exceeds these running times. For the tiny integral on the first basis differential, the power series $x(t)/y(t) = x(t)f(x(t))^{-1/2}$ can be computed modulo t^{n-1} using Newton iteration, requiring $\widetilde{O}(n \log n)$ field operations. Each other basis differential can be computed from the first by multiplication by the linear polynomial $x(t)$ and the definite integral evaluated with $\widetilde{O}(n)$ field operations, for a total of $\widetilde{O}(gn^2)$ bit operations. Computing $\phi(P)$ and $\phi(Q)$ to n digits of precision is cheap; directly using the formula in Algorithm 10 uses $\widetilde{O}(g + \log p)$ field operations. The last potentially significant step is computing and evaluating the f_i at each P and/or Q. The coefficients of the f_i can be read off in the reduction phase of Kedlaya's algorithm, and have $O(png)$ terms each. Evaluating (or even recording) all g of these forms takes $\widetilde{O}(png^2)$ field operations, or $\widetilde{O}(pn^2g^2)$ bit operations, which is proportional to the cost of doing the reduction.

4.3 Numerical Examples

Here are some sample computations made using our Sage implementation. Additional examples will appear in the first author's upcoming PhD thesis.

Example 21. Leprévost [21] showed that the divisor $(1, -1) - \infty^+$ on the genus 2 curve $y^2 = (2x - 1)(2x^5 - x^4 - 4x^2 + 8x - 4)$ over $\mathbb{Q}$ is torsion of order 29. Consequently, the integrals of holomorphic differentials against this divisor must vanish. We may observe this vanishing numerically, as follows. Let

$$C : y^2 = x^5 + \frac{33}{16}x^4 + \frac{3}{4}x^3 + \frac{3}{8}x^2 - \frac{1}{4}x + \frac{1}{16}$$

be the pullback of Leprévost's curve by the linear fractional transformation $x \mapsto (1-2x)/(2x)$ taking ∞ to $1/2$. The original points $(1,-1), \infty^+$ correspond to the points $P = (-1,1)$, $Q = (0, \frac{1}{4})$ on C. The curve C has good reduction at $p = 11$, and we compute

$$\int_P^Q \omega_0 = \int_P^Q \omega_1 = O(11^6), \int_P^Q \omega_2 = 7 \cdot 11 + 6 \cdot 11^2 + 3 \cdot 11^3 + 11^4 + 5 \cdot 11^5 + O(11^6),$$

consistent with the fact that $Q - P$ is torsion and ω_0, ω_1 are holomorphic but ω_2 is not.

Example 22. We give an example arising from the Chabauty method, taken from [23, § 8.1]. Let X be the curve

$$y^2 = x(x-1)(x-2)(x-5)(x-6),$$

whose Jacobian has Mordell-Weil rank 1. The curve X has good reduction at 7, and

$$X(\mathbb{F}_7) = \{(0,0), (1,0), (2,0), (5,0), (6,0), (3,6), (3,-6), \infty\}.$$

By [23, Theorem 5.3(2)], we know $|X(\mathbb{Q})| \leq 10$. However, we can find 10 rational points on X: the six rational Weierstrass points, and the points $(3, \pm 6), (10, \pm 120)$. Hence $|X(\mathbb{Q})| = 10$.

Since the Chabauty condition holds, there must exist a holomorphic differential ω for which $\int_\infty^Q \omega = 0$ for all $Q \in X(\mathbb{Q})$. We can find such a differential by taking Q to be one of the rational non-Weierstrass points, then computing $a := \int_\infty^Q \omega_0, b := \int_\infty^Q \omega_1$ and setting $\omega = b\omega_0 - a\omega_1$. For $Q = (3,6)$, we obtain

$$\begin{aligned} a &= 6 \cdot 7 + 6 \cdot 7^2 + 3 \cdot 7^3 + 3 \cdot 7^4 + 2 \cdot 7^5 + O(7^6) \\ b &= 4 \cdot 7 + 2 \cdot 7^2 + 6 \cdot 7^3 + 4 \cdot 7^5 + O(7^6). \end{aligned}$$

We then verify that $\int_Q^R \omega$ vanishes for each of the other rational points R.

Remark 23. It is worth pointing out some facts not exposed by Example 22. For instance, since ω is already determined by a single rational non-Weierstrass point, we could have used it instead of a brute-force search to find other rational points. More seriously, in other examples, the integral ω may vanish at a point defined over a number field which has a rational multiple in the Jacobian. Such points may be difficult to find by brute-force search; it may be easier to reconstruct them from p-adic approximations, obtained by writing $\int_\infty^* \omega$ as a function of a linear parameter of a residue disc, then finding the zeroes of that function.

5 Future Directions

Here are some potential extensions of our computation of Coleman integrals.

5.1 Iterated Integrals

Coleman's theory of integration is not limited to single integrals; it gives rise to an entire class of locally analytic functions, the *Coleman functions*, on which antidifferentiation is well-defined. In other words, one can define integrals

$$\int_P^Q \omega_n \cdots \omega_1$$

which behave formally like iterated path integrals

$$\int_0^1 \int_0^{t_1} \cdots \int_0^{t_{n-1}} f_n(t_n) \cdots f_1(t_1)\, dt_n \cdots dt_1.$$

These appear in several applications of Coleman integration, e.g., p-adic regulators in K-theory, and the nonabelian Chabauty method.

As in the case of a single integral, one can use Frobenius equivariance to compute iterated Coleman integrals on hyperelliptic curves. One obtains a linear system expressing all n-fold integrals of basis differentials in terms of lower order integrals. Note that the number of such n-fold integrals is $(2g)^n$, so this is only feasible for small n. The cases $n \leq 4$ are already useful for applications, but ideas for reducing the combinatorial explosion for larger n would also be of interest. (One must be slightly careful in dealing with Weierstrass residue discs; see Remark 17.)

We have made some limited experiments with double Coleman integrals in Sage. The Fubini identity

$$\int_P^Q \omega_2\omega_1 + \int_P^Q \omega_1\omega_2 = \left(\int_P^Q \omega_1\right)\left(\int_P^Q \omega_2\right)$$

turns out to be a useful consistency check for both single and double integrals.

5.2 Beyond Hyperelliptic Curves

It should be possible to convert other algorithms for computing Frobenius actions on de Rham cohomology, for various classes of curves, into algorithms for computing Coleman integrals on such curves. Candidate algorithms include the adaptation of Kedlaya's algorithm to superelliptic curves by Gaudry and Gürel [14], or the general algorithm for nondegenerate curves due to Castryck, Denef, and Vercauteren [7]. It should also be possible to compute Coleman integrals using Frobenius structures on Picard-Fuchs (Gauss-Manin) connections, extending Lauder's *deformation method* for computing Frobenius matrices [20].

5.3 Heights After Harvey

We noted earlier that our algorithms for Coleman integration over $\mathbb{Q}_p$ have linear runtime dependence on the prime p, arising from the corresponding dependence

in Kedlaya's algorithm. In [15], Harvey gives a variant of Kedlaya's algorithm with only square-root dependence on p (but somewhat worse dependence on other parameters), by reorganizing the computation so that the dominant step is finding the p-th term of a linear matrix recurrence whose coefficients are polynomials in the sequence index. Harvey demonstrates the practicality of his algorithm for primes greater than 2^{50}, which may have some relevance in cryptography for finding curves of low genus with nearly prime Jacobian orders.

It should be possible to use similar ideas to obtain square-root dependence on p for Coleman integration, by constructing a recurrence that computes not just the entries of the Frobenius matrix but also the values $f_i(P)$ and $f_i(Q)$. However, this is presently a purely theoretical question, as we do not know of any applications of Coleman integration for very large p.

Acknowledgments. The authors thank William Stein for access to his computer `sage.math.washington.edu` (funded by NSF grant DMS-0821725), and Robert Coleman and Bjorn Poonen for helpful conversations. Balakrishnan was supported by a National Defense Science and Engineering Graduate Fellowship and an NSF Graduate Research Fellowship. Bradshaw was supported by NSF grant DMS-0713225. Kedlaya was supported by NSF CAREER grant DMS-0545904, the MIT NEC Research Support Fund, and the MIT Cecil and Ida Green Career Development Professorship. Some development work was carried out at the 2006 MSRI Summer Graduate Workshop on computational number theory, and the 2007 Arizona Winter School on p-adic geometry.

References

1. Balakrishnan, J.S.: Local heights on hyperelliptic curves (2010) (in preparation)
2. Berkovich, V.G.: Integration of one-forms on p-adic analytic spaces. Annals of Mathematics Studies, vol. 162. Princeton University Press, Princeton (2007)
3. Besser, A.: Coleman integration using the Tannakian formalism. Math. Ann. 322(1), 19–48 (2002)
4. Besser, A.: On the computation of p-adic height pairings on Jacobians of hyperelliptic curves, Sage Days 5 (2007), `http://wiki.sagemath.org/days5/sched`
5. Besser, A., de Jeu, R.: $\mathrm{Li}^{(p)}$-service? An algorithm for computing p-adic polylogarithms. Math. Comp. 77(262), 1105–1134 (2008)
6. Bosch, S., Güntzer, U., Remmert, R.: Non-Archimedean analysis: A systematic approach to rigid analytic geometry. Springer, Berlin (1984)
7. Castryck, W., Denef, J., Vercauteren, F.: Computing zeta functions of nondegenerate curves. IMRP Int. Math. Res. Pap., Art. ID 72017, 57 (2006)
8. Coleman, R., de Shalit, E.: p-adic regulators on curves and special values of p-adic L-functions. Invent. Math. 93(2), 239–266 (1988)
9. Coleman, R.F.: Dilogarithms, regulators and p-adic L-functions. Invent. Math. 69(2), 171–208 (1982)
10. Coleman, R.F.: Torsion points on curves and p-adic abelian integrals. Ann. of Math. (2) 121(1), 111–168 (1985)
11. Coleman, R.F., Gross, B.H.: p-adic heights on curves. In: Algebraic Number Theory – in honor of K. Iwasawa. Advanced Studies in Pure Mathematics, vol. 17, pp. 73–81 (1989)

12. Fresnel, J., van der Put, M.: Rigid analytic geometry and its applications. In: Progress in Mathematics, vol. 218. Birkhäuser Boston Inc., Boston (2004)
13. Furusho, H.: p-adic multiple zeta values. II. Tannakian interpretations. Amer. J. Math. 129(4), 1105–1144 (2007)
14. Gaudry, P., Gürel, N.: An extension of Kedlaya's point-counting algorithm to superelliptic curves. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 480–494. Springer, Heidelberg (2001)
15. Harvey, D.: Kedlaya's algorithm in larger characteristic. Int Math Res Notices, Article ID No. rnm095, 2007, 29 (2007)
16. Harvey, D.: Efficient computation of p-adic heights. LMS J. Comput. Math. 11, 40–59 (2008)
17. Kedlaya, K.S.: Counting points on hyperelliptic curves using Monsky-Washnitzer cohomology. J. Ramanujan Math. Soc. 16, 323–338 (2001); erratum ibid 18, 417–418 (2003)
18. Kim, M.: The unipotent Albanese map and Selmer varieties for curves. Publ. Res. Inst. Math. Sci. 45(1), 89–133 (2009)
19. Kim, M.: Massey products for elliptic curves of rank 1. J. Amer. Math. Soc. 23, 725–747 (2010); Erratum by Balakrishnan, J.S., Kedlaya, K.S., Kim, M., http://www.ucl.ac.uk/~ucahmki/
20. Lauder, A.G.B.: Deformation theory and the computation of zeta functions. Proc. London Math. Soc. 88(3), 565–602 (2004)
21. Leprévost, F.: Jacobiennes de certaines courbes de genre 2: torsion et simplicité. J. Théor. Nombres Bordeaux 7(1), 283–306 (1995)
22. Mazur, B., Stein, W., Tate, J.: Computation of p-adic heights and log convergence. Doc. Math. Extra, 577–614 (2006) (electronic)
23. McCallum, W., Poonen, B.: The method of Chabauty and Coleman (2007) (preprint)
24. Stein, W.A., et al.: Sage Mathematics Software (Version 4.3.5), The Sage Development Team (2010), http://www.sagemath.org

Smallest Reduction Matrix of Binary Quadratic Forms

And Cryptographic Applications

Aurore Bernard[1] and Nicolas Gama[2]

[1] XLIM, Limoges, France
aurore.bernard@xlim.fr
[2] GREYC Ensicaen, Caen, France
nicolas.gama@greyc.ensicaen.fr

Abstract. We present a variant of the Lagrange-Gauss reduction of quadratic forms designed to minimize the norm of the reduction matrix within a quadratic complexity. The matrix computed by our algorithm on the input f has norm $O\left(\|f\|^{1/2}/\Delta_f^{1/4}\right)$, which is the square root of the best previously known bounds using classical algorithms. This new bound allows us to fully prove the heuristic lattice based attack against NICE Cryptosystems, which consists in factoring a particular subclass of integers of the form pq^2. In the process, we set up a homogeneous variant of Boneh-Durfee-HowgraveGraham's algorithm which finds small rational roots of a polynomial modulo unknown divisors. Such algorithm can also be used to speed-up factorization of pq^r for large r.

1 Introduction

Binary quadratic forms appeared progressively in the 17-th century, when Descartes and Fermat first introduced the concept of coordinates as a tool to algebraically solve geometric problems. Those forms have wide applications in mathematics and physics, especially in geometry, numerical analysis or algebraic topology. A binary quadratic form is a homogeneous polynomial of degree two in two variables, which can be viewed as the Cartesian equation of a surface $f(x, y) = ax^2 + bxy + cy^2$ on a given basis of $\mathbb{R}^2$. Of course, this equation varies with the basis of expression, and it is natural to define an equivalence relation to regroup all these possible equations into classes. Over the real field, there are six classes corresponding to the Sylvester's signatures. They can be distinguished by the sign of the discriminant $\Delta_f = b^2 - 4ac$, and the sign of $a + c$. Forms of strictly negative discriminant (imaginary forms) have a unique zero at the origin, which is also their unique local and global extremum. Forms of strictly positive discriminant (real forms) represent a saddle-shape.

Meanwhile, quadratic forms were also used over the integer ring by Fermat, Lagrange and Gauss to solve long standing problems from number theory. This time, binary quadratic forms are equations with integer coefficients of discrete

G. Hanrot, F. Morain, and E. Thomé (Eds.): ANTS-IX 2010, LNCS 6197, pp. 32–49, 2010.

scatter-plots on a given lattice basis of $\mathbb{Z}^2$. One defines a similar equivalence relation by base change, except that transformation matrices are now unimodular, and that they preserve the value of the discriminant. Problems related to this equivalence are more complicated than on the real field: for instance, in both real and imaginary cases, we do not know any polynomial way to compute the number of equivalence classes of a given discriminant. Deciding the equivalence of two forms is easy in the imaginary case, where each class contains a unique reduced representative computable in polynomial time. However, the problem is hard in the real case, where there are, depending on the notion of reduction, either an exponential number of polynomially computable reduced representatives, or a few representatives computable in exponential time.

A reduction algorithm takes as input a quadratic form and outputs a reduced form and the *reduction matrix*, which is a unimodular base-change matrix used to obtain this form. The most famous polynomial time reduction algorithms are Lagrange algorithm [15] (1773) commonly known as "Gauss reduction" algorithm [11] (1801). In [14] (1980), Lagarias modified the Gauss reduction algorithm for make it more efficient. This algorithm is the one used in practice, and which we refer as the Gauss reduction algorithm, or *Classical Gauss*, if we need to differentiate it from new flavors which we propose.

The cryptanalysis of [6] shows experimental evidences that the small size of reduction matrices have important applications to the factorization of some large numbers used in public key cryptosystems, especially those of the NICE cryptosystems (see [12,13]). However the best currently known upper-bounds on the size of reduction matrices [14,1] are by an order too large, and keep all these results on the factorization heuristic. In this paper, we specially design an efficient variant of the Gauss reduction algorithm to minimize the size of transformation matrix, and we prove constructive upper-bounds which are tight both in the worst case and in the average case. These bounds, combined with an improvement of the methods of [6], allows us to prove all the above mentioned heuristics of on the factorization of integers from the NICE cryptosystems.

2 Preliminaries and Notation

In this section we recall some definitions and properties concerning binary quadratic forms. For a more detailed account of the theory see [5,4,9]. Then, we summarize some results on the norm of a matrix.

Quadratic Forms. A *binary quadratic form* f is a homogeneous polynomial of degree two in two variables $f(x, y) = ax^2 + bxy + cy^2$ with $(a, b, c) \in \mathbb{Z}^3$ which we abbreviate as $f = (a, b, c)$. Throughout this paper the word *form* will be used in the sense of binary quadratic form. It is said *primitive* when $\gcd(a, b, c) = 1$. The *discriminant* of f is $\Delta_f = b^2 - 4ac$. A discriminant Δ_f is called *fundamental* if all the forms of discriminant Δ_f are necessarily primitive: for example, it is the case of all odd and square-free integers. The set of all primitive forms of discriminant Δ_f is denoted $\mathfrak{F}_{\Delta_f}$. We impose that the discriminant is not a perfect square then a and c are always non-zero. The form f can be factored as

$f(x,y) = a(x - y\zeta_f^-)(x - y\zeta_f^+)$ where ζ_f^- and ζ_f^+ are the complex roots of the univariate polynomial $f(x,1)$ which we call the *affine representation* of f. When $\Delta_f > 0$, each root of f live in $\mathbb{R}\backslash\mathbb{Q}$ and the form is *real*. In this case, ζ_f^- will denote the smallest root and ζ_f^+ the largest one. When $\Delta_f < 0$, the roots are in $\mathbb{C}\backslash\mathbb{R}$ and the form is *imaginary*. We note $\lambda(f) = \min\{|f(x,y)| : (x,y) \in \mathbb{Z}^2\backslash(0,0)\}$ the first minimum of f.

Composition Action. We note $\mathcal{M}^t$ the transpose of a matrix $\mathcal{M}$. The *polar representation* of f is the symmetric matrix $\begin{pmatrix} a & b/2 \\ b/2 & c \end{pmatrix}$ of determinant $-\Delta_f/4$. Let $\mathcal{M} = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \in \mathcal{M}_2(\mathbb{Z})$ be a 2×2 matrix with integer entries which we often abbreviate as $(\alpha, \beta; \gamma, \delta)$. We note Id the identity matrix of $\mathcal{M}_2(\mathbb{Z})$. The *composition action* of $\mathcal{M}$ on f is defined as the form $g(x,y) = f(\alpha x + \beta y, \gamma x + \delta y)$ and it is noted $g = f.\mathcal{M}$. The coefficients of g are $g = (f(\alpha,\gamma), b(\alpha\delta + \gamma\beta) + 2(a\alpha\beta + c\gamma\delta), f(\beta,\delta))$. We remark that for each root ζ_g of g, $\left(\frac{\alpha\zeta_g + \beta}{\gamma\zeta_g + \delta}\right)$ is a root of f. Finally, the polar representation of g is $\mathcal{M}^t f \mathcal{M}$ which implies that $\Delta_g = \det(\mathcal{M})^2 \Delta_f$.

Group action. Let $\mathrm{GL}_2(\mathbb{Z})$ be the *general linear group* of matrices in $\mathcal{M}_2(\mathbb{Z})$ which are invertible and its subgroup $\mathrm{SL}_2(\mathbb{Z})$ the *special linear group* of matrices which have a determinant equal to one. The action defined with either $\mathrm{GL}_2(\mathbb{Z})$ or $\mathrm{SL}_2(\mathbb{Z})$ on the set of primitive forms $\mathfrak{F}_{\Delta_f}$ of a given discriminant is a (right) group action. Two forms f and g are *equivalent* if they belong to the same $\mathrm{SL}_2(\mathbb{Z})$-orbit. In this case we note $f \sim g$. We define $Aut^+(f)$ the *group of automorphisms* of the form $f \in \mathfrak{F}_{\Delta_f}$ as $\{\mathcal{M} \in \mathrm{SL}_2(\mathbb{Z}),\ \mathrm{trace}(\mathcal{M}) > 0 \text{ and } f.\mathcal{M} = f\}$. The set of all automorphisms of f is $\pm Aut^+(f)$. The group $Aut^+(f)$ is known to be cyclic, and we call its generator the *fundamental automorphism of* f. The largest eigenvalue of the fundamental automorphism of f is the *fundamental unit*. It only depends on the discriminant Δ_f, and will be denoted ϵ_{Δ_f}.

Three specials transformations. We define the *symmetry* $\boldsymbol{S} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$, the *exchange* $\boldsymbol{E} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ and the *translation* by an integer $\boldsymbol{T}(h) = \begin{pmatrix} 1 & h \\ 0 & 1 \end{pmatrix}$. They are three (linear) transformations of $\mathrm{GL}_2(\mathbb{Z})$. All matrices in $\mathrm{GL}_2(\mathbb{Z})$ can be written as a product of powers of these three transformations and $\mathrm{SL}_2(\mathbb{Z})$ is generated by the product $\boldsymbol{ES}$ and $\boldsymbol{T}(1)$. The action of these transformations on f are $f.\boldsymbol{S} = (a, -b, c)$, $f.\boldsymbol{E} = (c, b, a)$ $f.\boldsymbol{T}(h) = (a, b + 2ah, f(h))$. Note the important fact: the roots of $f.\boldsymbol{S}$ are the opposite of the roots of f and the roots of $f.\boldsymbol{E}$ are the inverse of the roots of f, and that $\boldsymbol{T}(h)$ subtracts h to each roots of f.

Norms of matrices and forms. Let $\mathcal{M} = (\alpha, \beta; \gamma, \delta)$ be a matrix in $\mathcal{M}_2(\mathbb{Z})$. The *Euclidean norm* is $\|\mathcal{M}\|_2 = \sqrt{\alpha^2 + \beta^2 + \gamma^2 + \delta^2}$, and the *maximum norm* is $\|\mathcal{M}\| = \max(|\alpha|, |\beta|, |\gamma|, |\delta|)$. The norm $|||\mathcal{M}||| = \sup_{\|\boldsymbol{v}\|_2 = 1}(\|\mathcal{M}.\boldsymbol{v}\|_2)$ is the *induced Euclidean norm*, which is also the square root of the largest eigenvalue of $\mathcal{M}^t\mathcal{M}$. All the norms are equivalent: $\|\mathcal{M}\| \leqslant |||\mathcal{M}||| \leqslant \|\mathcal{M}\|_2 \leqslant 2\|\mathcal{M}\|$.

Additionally, the induced norm is sub-multiplicative: if $\mathcal{N} \in \mathcal{M}_2(\mathbb{Z})$ then $|||\mathcal{MN}||| \leqslant |||\mathcal{M}||| \cdot |||\mathcal{N}|||$ and $|||\mathrm{Id}||| = 1$, and it is lower-bounded by the *spectral radius* $\rho(\mathcal{M})$, which is the supremum among the absolute values of the eigenvalues of $\mathcal{M}$. By extension, we define the norms $\|f\|$,$\|f\|_2$ and $|||f|||$ of a form as the corresponding norm of its polar representation.

3 A New Reduction Algorithm for Real Quadratic Forms

A form $f = (a, b, c)$ is *reduced* if it satisfies two conditions simultaneously: a *normalization* condition, which defines the choice of the representative of $b \mod 2a$, and a *reduction* condition, which often upper-bounds the size of $|a|$ (or $|c|$). In the imaginary case, these conditions are very natural: a form is *normal* if and only if $b \in]-|a|, |a|]$ is minimal, and is *reduced* if additionally, $|a|$ is the minimum $\lambda(f)$. A single translation is needed to normalize any form. However, the reduction condition takes more steps to be achieved. The classical Gauss reduction reduces a form by successive swaps $\boldsymbol{SE}$ and normalizations $T(\lfloor -b/2a \rceil)$ (see [1]) until f is reduced. The Gauss reduction algorithm operates in quadratic time (see [1,21,18]). For each form f of discriminant $\Delta_f < -4$, there exists a unique reduced form g in each equivalence class, and a unique reduction matrix $\mathcal{M} \in \mathrm{SL}_2(\mathbb{Z})$ such that $f.\mathcal{M} = g$. In this case $Aut^+(f) = \{\mathrm{Id}\}$.

In the real case ($\Delta_f > 0$), the previous reduction conditions applied on $f = (a, b, c)$ are too restrictive, since the smallest integers $(\alpha, \beta) \neq (0, 0)$ such that $|f(\alpha, \beta)| = \lambda(f)$ are in general exponential in the size of f. No polynomial time algorithm can output an exponential reduction matrix. Thus, according to classical notions, f is *classically normalized* if and only if $b \in]-|a|, |a|]$ when $|a| \geqslant \Delta_f$ and $b \in \left]\sqrt{\Delta_f} - 2|a|, \sqrt{\Delta_f}\right[$ when $|a| < \Delta_f$, and f is *classically reduced* if additionally, $\left|\sqrt{\Delta_f} - 2|a|\right| < b < \sqrt{\Delta_f}$. It is known that only a finite subset of forms of discriminant Δ_f are classically-reduced, and that they form a *reduced cycle* in each class. The Real-Gauss reduction algorithm, which uses the classical normalization, finds a reduced form equivalent to its input in quadratic time (see [1]).

In this paper, given a normalized form f, we will bound the coefficients of the smallest reduction matrix $\mathcal{M} = (\alpha, \beta; \gamma, \delta)$ such that $g = f.\mathcal{M} = (a_g, b_g, c_g)$ is reduced. The case of imaginary forms is eased by the uniqueness of the reduction matrix. Lemma 5.6.1 in [1] give us that $\|\mathcal{M}\| \leqslant 2 \cdot \frac{\max\{|a|,|c|\}}{\sqrt{|\Delta_f|}}$. We improve this upper-bound with the following theorem:

Theorem 1 (Imaginary Bound). *Let $f = (a, b, c)$ be a normalized imaginary form of discriminant $\Delta_f < 0$, and $\mathcal{M} = (\alpha, \beta; \gamma, \delta)$ the reduction matrix such that $g = f.\mathcal{M} = (a_g, b_g, c_g)$, $\mathcal{M}$ satisfies these two upper-bounds:*

1) $\|\mathcal{M}\| \leqslant \frac{2}{\sqrt{3}} \cdot \sqrt{\frac{|c|}{|a_g|}}$

2) $|\alpha\beta\gamma\delta|^{1/4} \leqslant |\gamma\delta|^{1/2} \leqslant \frac{2}{3^{1/4}} \cdot \left(\frac{|ac|}{|\Delta_f|}\right)^{1/4}$.

Proof. One has $|a_g| = |f(\alpha,\gamma)| = |a|\gamma^2\left((\alpha/\gamma + \frac{b}{2a})^2 + \frac{|\Delta_f|}{4a^2}\right)$, which can be lower-bounded by $\frac{|\Delta_f|}{4|a|}\gamma^2$. It follows that $\gamma^2 \leqslant \frac{4|aa_g|}{|\Delta_f|}$, and similarly $\delta^2 \leqslant \frac{4|cc_g|}{|\Delta_f|}$. Therefore $|\gamma\delta| \leqslant \frac{4\sqrt{|ac|}}{\sqrt{3|\Delta_f|}}$. The first inequality comes from $3|a_g c_g| \leqslant |\Delta_f|$, because g is reduced. Unless the transformation is trivial (Id or $\boldsymbol{SE}$), the normalization condition induces the inequalities $|\alpha| \leqslant |\gamma|$ and $|\beta| \leqslant |\delta|$, which proves $|\alpha\beta\gamma\delta|^{1/4} \leqslant |\gamma\delta|^{1/2}$. □

Thus, the norm of the reduction matrix is in fact basically in $O\left(\sqrt{\|f\|/\sqrt{|\Delta_f|}}\right)$.

In the real case however, this proof would not apply directly, because the term $\left((\alpha/\gamma + \frac{b}{2a})^2 - \frac{|\Delta_f|}{4a^2}\right)$ can be exponentially close to 0. The problem is that in the real case, each reduced cycle contains a large (often exponential) number of equivalent reduced forms, and some of them are exponentially far from f. A constructive approach is needed to build a polynomial reduction matrix. The analysis of the Gauss reduction algorithm in [1,14] basically proves that the norm of the computed reduction matrix is bounded by $O(\|f\|)$. In this paper, we study a variant of this algorithm which finds a reduction matrix of norm $O\left(\sqrt{\|f\|/\sqrt{\Delta_f}}\right)$ and we verify that it is tight even in the average case.

We define new relaxed notions of reduction and normalization, and express them according to the roots of the forms, which is more intuitive than the classical conditions on the coefficients:

Definition 1. *A real binary quadratic form f is:*

- primary normalized *if* $0 < \zeta_f^+ < 1$ *and* primary reduced *if also* $\zeta_f^- < -1$
- secondary normalized *if* $-1 < \zeta_f^- < 0$ *and* secondary reduced *if also* $1 < \zeta_f^+$.

Finally f is largely reduced *if it is either primary or secondary reduced.*

Both primary and secondary notions are exchanged by the action of $\boldsymbol{S}$, which negates the roots. Furthermore, primary and secondary reductions are exchanged by $\boldsymbol{E}$, which inverts the roots. As usual, primary and secondary normalization can always be achieved by the action of some $\boldsymbol{T}(h)$. Note that a classically normalized form, which has by definition at least one root in the interval $]-1,1[$, is either primary or secondary normalized. Similarly, a classically reduced form (a,b,c) is a largely-reduced form satisfying $b > 0$, which can again be ensured by the action of $\boldsymbol{S}$. Our main contribution is to solve the following problems, which are equivalent.

Lemma 1. *The two problems are equivalent:*

1. **Smallest $\mathbf{SL}_2(\mathbb{Z})$ matrix** *Given a classically-normalized real form f, find $\mathcal{M} \in SL_2(\mathbb{Z})$ such that $f.\mathcal{M}$ is classically-reduced and $\|\mathcal{M}\|$ is minimal.*
2. **Smallest $\mathbf{GL}_2(\mathbb{Z})$ matrix** *Given a primary-normalized real form f, find $\mathcal{M} \in GL_2(\mathbb{Z})$ such that $f.\mathcal{M}$ is largely-reduced and $\|\mathcal{M}\|$ is minimal.*

Proof. From a solution $\mathcal{M} \in \mathrm{GL}_2(\mathbb{Z})$ of *Problem 2*, one deduces a solution of *Problem 1* by left-multiplication by Id or $\boldsymbol{S}$ to make the normalization of the input correspond, followed by a right-multiplication by Id or $\boldsymbol{S}$ to force the coefficient b of the reduced form to be positive, followed by a right multiplication by Id or $\boldsymbol{E}$ so that the determinant is $+1$. The reduction of *Problem 2* to *Problem 1* is similar. Since Id, $\boldsymbol{S}$ and $\boldsymbol{E}$ are permutation matrices, they do not modify these norms $|||\cdot|||$ or $\|\cdot\|$. Remark that, reducing a problem to the other also preserves the absolute value of the product of the coefficients in each row of the reduction matrices. □

Lemma 1 motivates the search of a reduction algorithm solving the less restrictive *Problem 2*, since we can use the above permutation matrices to return to classical notions in $\mathrm{SL}_2(\mathbb{Z})$.

3.1 Algorithm and Analysis

Let f be a real form. We define the two integers h_f^+ and h_f^- as $h_f^+ = \left\lfloor \zeta_f^+ \right\rfloor$ and $h_f^- = \left\lceil \zeta_f^- \right\rceil$. It is easy to show that h_f^+ and h_f^- are respectively the unique integers such that $f.\boldsymbol{T}(h_f^+)$ is primary-normalized, and $f.\boldsymbol{T}(h_f^-)$ is secondary-normalized. Among the two integers h_f^-, h_f^+ the one of smallest absolute value is noted $\boldsymbol{h}(f)$: that is $\boldsymbol{h}(f) = h_f^+$ if $|h_f^+| < |h_f^-|$, and $\boldsymbol{h}(f) = h_f^-$ otherwise. In other words, $\boldsymbol{h}(f)$ is the *shortest normalization* of f. As a comparison, there is only a single integer ν_f in the classical case such that $f.\boldsymbol{T}(\nu_f)$ is classically-normalized, ν_f being one of the integers h_f^-, h_f^+ but not necessarily the one with the smallest absolute value. Our reduction algorithm, is a variant of the Gauss reduction which operates in $\mathrm{GL}_2(\mathbb{Z})$. It alternates exchange $\boldsymbol{E}$ and the shortest normalization $T(\boldsymbol{h}(f))$ at each loop, and terminates on a largely-reduced form. As we will see later, any kind of normalization by h_f^- or h_f^+ would make a reduction algorithm terminate[1], but the choice of the shortest normalization $\boldsymbol{h}(f)$ instead of the classical ν_f (especially during the last steps) is the key element to minimize the reduction matrix. The main result of the section is the following theorem on the quality of the output of our algorithm, which is the real-case analogue of Theorem 1.

Algorithm 1. *Red*GL2

Input: $f = (a, b, c)$ a primary-normalized form
Output: $f.\mathcal{M}$ a largely-reduced form and $\mathcal{M} \in \mathrm{GL}_2(\mathbb{Z})$

1: $\mathcal{M} = \mathrm{Id}$
2: **while** f not largely-reduced **do**
3: $\quad f \leftarrow f.\boldsymbol{E}$ and $\mathcal{M} \leftarrow \mathcal{M}\boldsymbol{E}$ ▷ Exchange step
4: $\quad f \leftarrow f.\boldsymbol{T}(\boldsymbol{h}(f))$ and $\mathcal{M} \leftarrow \mathcal{M}\boldsymbol{T}(\boldsymbol{h}(f))$ ▷ Normalization step
5: **end while**
6: **return** f and $\mathcal{M}$

[1] The original Gauss algorithm of 1801 used actually the largest normalization at each step. The number of reduction steps is exponential on some entries. Lagarias introduced the classical normalization to obtain a quadratic complexity

Theorem 2 (Real bound). *Let $f = (a, b, c)$ be a primary-normalized form of discriminant $\Delta > 0$. Given f as input, RedGL2 terminates after at most $\left(\frac{\log(|a|/\sqrt{\Delta})}{2\cdot\log(\omega)} + 4\right)$ iterations where $\omega = \frac{1+\sqrt{5}}{2}$ is the gold number. Its output $\mathcal{M} = (\alpha, \beta; \gamma, \delta)$ and $f_r = f.\mathcal{M} = (a_r, b_r, c_r)$ satisfies:*

1) $\| \mathcal{M} \| \leqslant 4 \cdot \sqrt{|a|/|a_r|}$

2) $(|\alpha\beta\gamma\delta|)^{1/4} \leqslant |\gamma\delta|^{1/2} \leqslant \sqrt{21} \cdot \sqrt{|a|/\sqrt{\Delta}}$.

Before proving this theorem, we remark that the best known upper-bounds achieved by the classical Gauss algorithm under the same conditions (see theorem 4.4 of [1]) are $\|\mathcal{M}\| \leqslant |a|(1+1/\Delta)$ and $|\gamma\delta|^{1/2} \leqslant \left(|a|/\sqrt{\Delta}\right)(1+1/\sqrt{\Delta})$. They are basically the square of the upper-bounds of *Red*GL2. Figure 1 and 2 illustrate respectively the families of forms $F_n = (-n, b, 1)$ and $G_n = (n, n, 1)$ with $n \in \mathbb{N}$ and $b = \lfloor 2n/3 - 2/3 \rfloor$, which are families of forms where the Gauss reduction algorithm outputs reduction matrices $\sqrt{\Delta}$ times larger than our variant *Red*GL2. Finally, note that a multiplicative triangular inequality on the norms of the polar representations of $f = f_r.\mathcal{M}^{-1}$ yields $\sqrt{\|f\|/\|f_r\|} \leqslant \sqrt{2}\|\mathcal{M}\|$, which confirms the optimality of Theorem 2 in average. The analysis of Gauss reduction algorithm in [1] upper-bounds the number of iterations by $\left(\frac{\log(|a|/\sqrt{\Delta})}{2\cdot\log(2)} + 2\right)$ reduction steps. Our upper-bound on the number of iterations of *Red*GL2 is tight in the worst case, and is only by a multiplicative factor around 1.4 larger than the maximum number of iterations of the Gauss reduction algorithm. However the primary goal of *Red*GL2 is the minimization of the reduction matrix.

3.2 Proof of Theorem 2

To prove Theorem 2, we first study the termination cases, characterized by the presence of integers between the roots of $f.E$, and where the choice of the shortest normalization is of greatest importance. Eventually, we shall treat the general case and the complexity.

Termination cases. We first study the two cases where the algorithm terminates in a single step of reduction. The first one deals with normal form f containing exactly one integer between its roots. This is the only case where $h_f^- = h_f^+$, so all notions of normalizations (classical, primary, secondary, shortest) coincide.

Lemma 2. *Let $f = (a, b, c)$ be a real form satisfying $-1 < \zeta_f^- < 0 < \zeta_f^+ < 1$, and $h = \boldsymbol{h}(f.\boldsymbol{E})$. The form $f_r = f.\boldsymbol{ET}(h) = (a_r, b_r, c_r)$ is largely-reduced, and its coefficients satisfy $a_r = c$, $|c_r| \leqslant |a|$, and $h^2|a_r| \leqslant |a|$.*

Proof. The reduction matrix from f to f_r is $\boldsymbol{ET}(h) = (0, 1; 1, h)$. Consider the parabola $p(x) = cx^2 + bx + a$ which is the affine representation of $g = f.\boldsymbol{E}$. Then we have $h = \boldsymbol{h}(g)$, and $\zeta_g^- < h_g^- \leqslant -1 < 1 \leqslant h_g^+ < \zeta_g^+$, $c_r = p(\boldsymbol{h}(g))$ and $p(0) = a$. By definition of h we have two cases: if $-b/2c > 0$ then we have $h = h_g^- < 0 < -b/2c$, else we have $-b/2c < 0 < h = h_g^+$. In both cases we

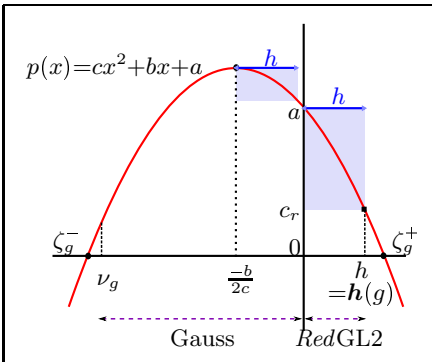


Fig. 1. Illustration of Lemma 2
This figure illustrates the convexity inequalities of Lemma 2. In this case, the shortest normalization chosen by RedGL2 is $\boldsymbol{h}(g) = \lfloor \zeta_g^+ \rfloor$, *which can be* $O(\sqrt{\Delta})$ *smaller than the classical normalization* $\nu(g) = \lceil \zeta_g^- \rceil$ *in Gauss Algorithm. It is clear that* $|c_r|$ *is in the interval* $[0, |a|]$. *Comparison of heights of the two rectangles on the same convex and decreasing branch of the parabola, gives* $|ch^2| \leqslant |a|$.

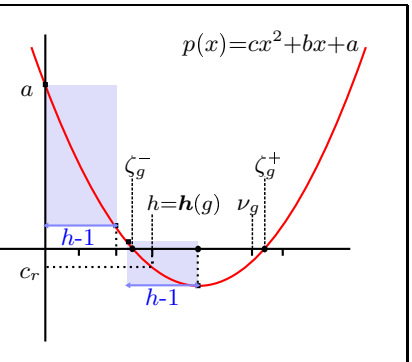


Fig. 2. Illustration of Lemma 3
This figure is the analogue for Lemma 3. In this case, the shortest normalization chosen by RedGL2 is $\boldsymbol{h}(g) = \lceil \zeta_g^- \rceil$, *which can be* $O(\sqrt{\Delta})$ *smaller than the classical normalization in Gauss algorithm is* $\nu(g) = \lfloor \zeta_g^+ \rfloor$. *Inequality on the slopes of* p *before and after* ζ_g^- *gives* $|c_r| \leqslant |a|$. *Comparison of heights of the two rectangles on the same convex and decreasing branch of the parabola, gives* $|c(h-1)^2| \leqslant |a|$.

graphically verify that $|c_r| = |p(h)| < |p(0)| = |a|$ (see Figure 1). A convexity inequality on p between $[0, h]$ and $[-b/2c, -b/2c + h]$ shows $|a - c_r| \geqslant |c|h^2$. Since a and c_r have the same sign and $|a|$ is larger, then $|a| \geqslant |a_r|h^2$. □

Theorem 2 holds in this termination case: the reduction matrix is $\mathcal{M} = (0, 1; 1, h)$. By Lemma 2, its norm satisfies $\|\mathcal{M}\| = h \leqslant \sqrt{|a|/|a_r|}$. Since $f = f_r.\mathcal{M}^{-1}$, its first coefficient is $a = a_r h^2 - b_r h + c_r$, thus $b_r h = -a + c_r + a_r h^2$ and $(b_r h)^2 - 4a_r c_r h^2 = \Delta \cdot h^2 = a^2 + c_r^2 + a_r^2 h^4 - 2ac_r - 2aa_r h^2 - 2a_r c_r h^2 \leqslant (|a| + |c_r| + |a_r|h^2)^2 \leqslant 9|a|^2$, which proves the second point of Theorem 2.

The second case of single-step termination concerns normalized form f such that at least two integers lie between the roots of $f.\boldsymbol{E}$ (namely $h^-_{f.\boldsymbol{E}} < h^+_{f.\boldsymbol{E}}$). We just write a proof for primary-normalized forms, but it can be easily extended to secondary-normalized forms.

Lemma 3. *Let* $f = (a, b, c)$ *be a real form satisfying* $0 < \zeta_f^- < \zeta_f^+ < 1$, *and such that* $h^-_{f.\boldsymbol{E}} < h^+_{f.\boldsymbol{E}}$. *If* $h = \boldsymbol{h}(f.\boldsymbol{E})$, *then* $f_r = f.\boldsymbol{E}\boldsymbol{T}(h) = (a_r, b_r, c_r)$ *is secondary-reduced, and its coefficients satisfy* $a_r = c$, $|c_r| \leqslant |a|$, *and* $h^2|a_r| \leqslant 4|a|$.

Proof. The proof of this lemma is also based on convexity inequalities. Let $g = f.\boldsymbol{E}$, of affine representation $p(x) = cx^2 + bx + a$. Note that $h = \lceil \zeta_g^- \rceil \geqslant 2$. Again, one has $p(0) = a$, $p(h) = c_r$. It follows from the definition that f_r is

secondary-reduced. The reduction matrix is $\mathcal{M} = (0, 1; 1, h)$, which proves $a_r = c$. Application of a convexity inequality (see Figure 2) on p in the two intervals $[0; h-1]$ and $\left[-\frac{b}{2c} - (h-1); -\frac{b}{2c}\right]$ of same length yields $|a_r|(h-1)^2 \leqslant |a - p(h-1)| \leqslant |a|$, therefore $|a_r|h^2 \leqslant 4|a_r|(h-1)^2 \leqslant 4|a|$. Finally, another convexity inequality centered on ζ_g^- gives $\frac{p(0)-p(\zeta_g^-)}{0-\zeta_g^-} \leqslant \frac{p(h)-p(\zeta_g^-)}{h-\zeta_g^-}$, so $|a| = p(0) \geqslant \frac{\zeta_g^-}{h-\zeta_g^-} \cdot (-p(h)) \geqslant |c_r|$. □

Once again, Theorem 2 holds in this termination case, but this time, $\|\mathcal{M}\| = h \leqslant 2\sqrt{|a|/|a_r|}$ and $\Delta \cdot h^2 \leqslant (|a| + |c_r| + |a_r|h^2)^2 \leqslant (6|a|)^2$.

General case. We now prove the general case of Theorem 2. We call $f_i = (a_i, b_i, c_i)$ the successive values of f at the beginning of the while loop of Algorithm 1, and $h_i = \boldsymbol{h}(f_i.\boldsymbol{E})$. We suppose that the primary-normalized form f_0 does not have any integer between its roots (otherwise it would either already be reduced or as in Lemma 2). Thus $0 < \zeta_{f_0}^- < \zeta_{f_0}^+ < 1$. For each iteration i in the loop, if there is at least one integer between the roots of $f_i.E$, then we set $m = i + 1$ and the algorithm reaches one of the two termination cases above. Otherwise the shortest normalization h_i is the primary one $h_i = h_{f_i.\boldsymbol{E}}^+ < h_{f_i.\boldsymbol{E}}^-$. Thus f_i is also primary-normalized and $0 < \zeta_{f_i}^- < \zeta_{f_i}^+ < 1$. Note that the distance between the roots strictly increases $\left|\zeta_{f_i}^+ - \zeta_{f_i}^-\right| = \left|\zeta_{f_{i-1}.\boldsymbol{E}}^+ - \zeta_{f_{i-1}.\boldsymbol{E}}^-\right| = \left|\zeta_{f_{i-1}}^+ \times \zeta_{f_{i-1}}^-\right|^{-1} \cdot \left|\zeta_{f_{i-1}}^+ - \zeta_{f_{i-1}}^-\right| \geqslant \left|\zeta_{f_{i-1}}^+ - \zeta_{f_{i-1}}^-\right|$. Such process can not hold forever, otherwise the integer sequence of the first coefficients $|a_i| = \sqrt{\Delta}/\left|\zeta_{f_i}^+ - \zeta_{f_i}^-\right|$ would be strictly decreasing. This proves the termination of the algorithm. The integer m is the smallest index, such that $f_{m-1}.\boldsymbol{E}$ contains at least one integer between its roots. The shortest normalization $h_{m-1} = h_{f_{m-1}.\boldsymbol{E}}^- \leqslant h_{f_{m-1}.\boldsymbol{E}}^+$ is in this case secondary, and satisfies $h_{m-1} \geqslant 2$.

We eventually use the following lemma to conclude the proof of Theorem 2.

Lemma 4. *Let $f = (a, b, c)$ and $g = (a_g, b_g, c_g)$ be two real forms and $\mathcal{M} = (\alpha, \beta; \gamma, \delta) \in GL_2(\mathbb{Z})$ such that $f.\mathcal{M} = g$. If all the roots of g are positive and $\gamma \geqslant 0$ and $\delta \geqslant 1$ then $|a_g|\delta^2 \leqslant |a|$.*

Proof. If $\gamma = 0$, then $\mathcal{M}$ is triangular, so $|\alpha| = |\delta| = 1$ and $|a_g| = |a|$. We now suppose $\gamma > 0$. Let ζ_g be a root of g, then $\zeta_f = \frac{\alpha\zeta_g + \beta}{\gamma\zeta_g + \delta}$ is a root of f. We have $|\alpha/\gamma - \zeta_f| = 1/|\gamma^2\zeta_g + \gamma\delta| < 1/\gamma\delta$ thanks to the positivity conditions. Since this bound holds for both roots of f, $|a_g| = \gamma^2|a| \left|\alpha/\gamma - \zeta_f^-\right| \left|\alpha/\gamma - \zeta_f^+\right| < |a|/\delta^2$. □

We continue the proof of Theorem 2 by applying this lemma to the main loop of *Red*GL2. Note that for each $i \in [1; m]$, the reduction matrix from f_0 to f_i is

$$\mathcal{M}_i = \begin{pmatrix} 0 & 1 \\ 1 & h_0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & h_1 \end{pmatrix} \cdots \begin{pmatrix} 0 & 1 \\ 1 & h_{i-1} \end{pmatrix} = \begin{pmatrix} \alpha_i & \beta_i \\ \gamma_i & \delta_i \end{pmatrix}. \tag{1}$$

Their coefficients are all positive, and satisfy these recurrence equalities for $i \geqslant 2$:

$$\begin{array}{lll} \gamma_{i+1} = \delta_i = h_{i-1}\delta_{i-1} + \delta_{i-2} & \text{and} & (\delta_0, \delta_1) = (1, h_1) \\ \alpha_{i+1} = \beta_i = h_{i-1}\beta_{i-1} + \beta_{i-2} & \text{and} & (\beta_0, \beta_1) = (0, 1) \end{array}$$

Since all the $(h_j)_{j=0..i}$ are greater than 1, it follows that $\alpha_i \leqslant \min(\beta_i, \gamma_i) \leqslant \max(\beta_i, \gamma_i) \leqslant \delta_i$ and $\|\mathcal{M}_i\| = \delta_i \geqslant \omega^{i-2}$ by induction and comparison to the Fibonacci sequence [2]. Applying Lemma 4 on f_0 and f_{m-1} implies that $\|\mathcal{M}_{m-1}\|^2 \leqslant |a_0|/|a_{m-1}|$. At iteration m, Lemma 4 can be applied to $f_m.T(-1)$, which has positive roots and shares its first coefficient a_m with f_m. The transformation matrix $\mathcal{M}_m T(-1) = \mathcal{M}_{m-1}\,(0, 1; 1, h_{m-1} - 1)$ still satisfies the conditions of Lemma 4 because $h_{m-1} \geqslant 2$. We obtain $\|\mathcal{M}_m T(-1)\|^2 \leqslant |a_0|/|a_m|$, and finally $\|\mathcal{M}_m\|^2 \leqslant 4|a_0|/|a_m|$ after a backwards translation by $T(1)$.

We already know that f_m is secondary-normalized and that the largest root of f_m is positive. There are two cases:

1. If the largest root of f_m is strictly greater than 1, then $r = m$, f_r is secondary-reduced, and the reduction matrix is $\mathcal{M}_m = (\alpha_m, \beta_m; \gamma_m, \delta_m)$. One already has $\|\mathcal{M}_m\|^2 \leqslant 4|a_0|/|a_r|$. From $f_0 = f_r.\mathcal{M}^{-1}$, we draw $a_0 = a_r\delta_m^2 - b_r\delta_m\gamma_m + c_r\gamma_m^2$, so $\Delta\delta_m^2\gamma_m^2 = (b_r\delta_m\gamma_m)^2 - 4a_rc_r\delta_m^2\gamma_m^2 \leqslant (|a_0| + |a_r\delta_m^2| + |c_r\gamma_m^2|)^2$. Since by construction $\gamma_m^2 = \delta_{m-1}^2 = \|\mathcal{M}_{m-1}\|^2$ and by Lemma 3 applied on f_{m-1} and f_r, $|c_r| \leqslant |a_{m-1}|$, one finds $\Delta\delta_m^2\gamma_m^2 \leqslant (6 \cdot |a_0|)^2$.
2. If the second root of f_m is strictly lower than 1, then by Lemma 2, f_{m+1} is reduced. The matrix of reduction is $\mathcal{M} = \begin{pmatrix} \alpha_r & \beta_r \\ \gamma_r & \delta_r \end{pmatrix} = \mathcal{M}_m \cdot \begin{pmatrix} 0 & 1 \\ 1 & h_m \end{pmatrix}$, and $r = m + 1$. Thus $\|\mathcal{M}\|^2 \leqslant \|\mathcal{M}_m\|^2(1 + |h_m|)^2 \leqslant 4|a_0|/|a_m| \cdot 4h_m^2 \leqslant 16|a_0|/|a_r|$. One still has $\Delta\delta_r^2\gamma_r^2 \leqslant (|a_0| + |a_r\delta_r^2| + |c_r\gamma_r^2|)^2 \leqslant (21|a_0|)^2$, because $|c_r| \leqslant |a_m|$ by Lemma 2.

This concludes the proof of items 1) and 2) of Theorem 2. It remains the complexity issue, proved in the following paragraph.

Complexity. We now prove the number of iterations performed by *Red*GL2. Two steps before the end, at iteration $r - 2$ of *Red*GL2, we know that the form $f_{r-2} = (a_{r-2}, b_{r-2}, c_{r-2})$ satisfies $\sqrt{\Delta} < |a_{r-2}|$, because the distance between the roots of f_{m+1} is smaller than 1. By Lemma 4 we have $\omega^{r-4} \leqslant \|\mathcal{M}_{r-2}\| \leqslant \sqrt{|a_0/a_r - 2|} \leqslant \sqrt{|a_0/\sqrt{\Delta}|}$. It follows that $r - 4$ is upper-bounded by $\left(\dfrac{\log\left(|a|/\sqrt{\Delta}\right)}{2\log(\omega)}\right)$ steps where $\omega = \frac{1+\sqrt{5}}{2}$.

The worst case complexity of algorithm *Red*GL2 is reached when all the normalizations occurring in the algorithm until the index $r - 2$ are by $h = 1$. For instance, we experimentally verify that it is the case on this family of inputs $g.(\boldsymbol{T}(-1)\boldsymbol{E})^n$ where g is reduced and n grows.

[2] The ith number of the sequence of Fibonacci numbers is bigger than ω^{i-2}.

4 Proof of Heuristic Cryptanalysis of the NICE Cryptosystems

We propose an application of the results of the previous section to the cryptanalyses of the NICE cryptosystems. There are two variants, which are by chronological order NICE Imaginary [12] (with imaginary forms), and NICE Real [13] (with real forms). Their security relies on the intractability of factorization of the public discriminant N. They were designed for a similar level of security as RSA, but with faster decryption, since the decryption process has quadratic complexity. Both are now considered as broken. The first one succumbed by a proved arithmetic attack in [7]. However, the more general attack against both versions of NICE (in [6]) using lattice reduction remains only experimental and relies on two heuristic assumptions. In this paper, we provide an alternative point of view on the lattice attack, which allows to avoid the use of these heuristics and to prove the attack entirely.

Both variants of NICE (Real and Imaginary) have originally been described in terms of ideals of quadratic orders, and are based on a morphism between classes of primitive forms of fundamental discriminant p and classes of primitive forms of non-fundamental discriminant $N = q^2p$. These notions are actually not needed here to understand the lattice attack, therefore we will here give a simple description solely in term of quadratic forms.

4.1 Lifting Quadratic Orders

We summarize some important properties on the relation between the sets $\mathfrak{F}_p$ and $\mathfrak{F}_N$ of primitives forms of discriminants respectively p and $N = q^2p$, using the terminology we introduced in the last section. For the cryptographic interest we restrict ourselves to the case where q is an odd prime. The following background theory can be found in [5,4,9].

Integer matrices of determinant q. We define an equivalence relation modulo $\mathrm{SL}_2(\mathbb{Z})$ between two integer matrices A and $B \in \mathcal{M}_2(\mathbb{Z})$ by $A \equiv B \iff \exists \mathcal{M} \in \mathrm{SL}_2(\mathbb{Z}),\ A\mathcal{M} = B$. The 2×2 integer matrices of determinant q correspond to matrices of rank 1 mod q, they fall into $q+1$ equivalence class, which are characterized by the (projective) direction from $\{0, 1, ..., q-1, \infty\}$ of their image mod q. Each class contains a unique Hermite normal form: $Q_k = \begin{pmatrix} q & k \\ 0 & 1 \end{pmatrix},\ k \in \{0, \ldots, q-1\}$ or $Q_\infty = \begin{pmatrix} 1 & 0 \\ 0 & q \end{pmatrix}$.

Lift. As we can see in [5, section 7], for each form f of discriminant $N = pq^2$ and each $\mathcal{M} \in \mathcal{M}_2(\mathbb{Z})$ of determinant q, there exists a (non-unique) form $g \in \mathfrak{F}_p$ such that $f = g.\mathcal{M}$. When $\mathcal{M} = Q_\infty$, we define a particular function φ (also called *lift*) which computes such $g \in \mathfrak{F}_p$ from $f = (a, b, c) \in \mathfrak{F}_N$ such that $\gcd(a, q) = 1$ as follows: $\varphi(f) = (a, \frac{b+2ah}{q}, \frac{ah^2+bh+c}{q^2})$ where $h \in [1-q, \ldots, 0]$ and $h = -b/2a \mod q$. Note that all the divisions are exact since f is primitive of discriminant $N = 0 \mod q^2$ and q is an odd prime. It must be noted that the

lift preserves the first coefficient a of the form. It is also clear that φ preserves primary normalization, because its action on the roots of f is a translation by $-h \in [0, q-1]$ followed by a division by q, which stabilizes the interval $]0,1[$ of the largest root. Finally, equivalence of forms is stable by lift $\forall f, f' \in \mathfrak{F}_N,\ f \sim f' \implies \varphi(f) \sim \varphi(f')$.

The converse is in general false. Given a form $g \in \mathfrak{F}_p$ and U its fundamental automorphism, there are exactly $q - (p/q)$ primitive forms (in $\mathfrak{F}_N$) among $\{g.Q_0, \ldots, g.Q_{q-1}, g.Q_\infty\}$ where (p/q) denotes the Legendre symbol. These forms split into $(q-(p/q))/s_q$ sets of s_q equivalent forms (see [5] theorem 7.4), where s_q is the order of U modulo q. The fundamental unit ϵ_N is equal to the power $(\epsilon_p)^{s_q}$. These $(q-(p/q))/s_q$ different classes of equivalence are the only ones to be lifted to the class of g.

Reduced cycle. Let $g \in \mathfrak{F}_\Delta$ be a classically-reduced form of discriminant $\Delta > 0$, the *right neighbour* of g is the classical normalization of $g.\boldsymbol{SE}$. If we note $\boldsymbol{H}(g)$ the *largest normalization* of g (by the integer among h_g^-, h_g^+ of largest absolute value), then the right neighbour of g is $g.\boldsymbol{SET}(\boldsymbol{H}(g.\boldsymbol{SE}))$. Successive iterations of the right neighbour enumerates all the reduced forms equivalent to g, and define the reduced cycle of the class of g. The cardinality of such reduced cycle is in $O(\log(\epsilon_\Delta))$ where ϵ_Δ is the fundamental unit.

Principal cycle, and q-belt. The *principal class* of a discriminant $\Delta > 0$ is the class containing $(1, 1, *)$. The *principal form* is the classical-normalization of this form, and the *principal cycle* $\mathbb{1}_\Delta$ is the reduced cycle of the principal class. Note that the principal class is the only class containing a form of first (or last) coefficient equals to 1.

We define the *q-belt* of a discriminant $N = pq^2$ as the set of all primary normalized forms $(q^2, kq, *)$ of the principal class. Necessarily, $k \in [-\sqrt{p}, 2q-\sqrt{p}]$. There are exactly $s_q - 1$ forms in the q-belt of N: let g_0 be the principal form $(1, *, *)$ of $\mathfrak{F}_N$ and $f = \varphi(g_0)$ is (necessarily) the principal form of $\mathfrak{F}_p$. Let U be the fundamental automorphism of f, we set by induction $k_0 = \infty$ and k_i the unique integer such that $UQ_{k_{i-1}} \equiv Q_{k_i}$ for $i \geqslant 1$. Note that $Q_{k_i} \equiv U^i Q_{k_0}$, and that the order of $U \mod q$ is precisely s_q, therefore the sequence (k_i) is periodic and $k_{s_q} = k_0 = \infty$. Finally, the q-belt of N is the set $\{g_1 = f.Q_{k_1}, \ldots, g_k = f.Q_{k_{s_q-1}}\}$. They are indeed primary-normalized and equivalent by construction. A transformation matrix from g_i to g_{i-1} is by construction $Q_{k_i}^{-1} U Q_{k_{i-1}} \in \mathrm{SL}_2(\mathbb{Z})$, because $UQ_{k_{i-1}} \equiv Q_{k_i}$.

4.2 Cryptosystem Real NICE

We now describe the NICE Real encryption and decryption. The public key is a composite integer $N = pq^2$ and the secret key (p, q) with p and q two distinct primes of the same size, satisfies two conditions:

- p is a Schinzel prime [19] which is a positive squarefree integer of the form $p = A^2x^2 + 2Bx + C$ with $A, B, C, x \in \mathbb{Z}$, $A \neq 0$ and $B^2 - 4AC$ dividing $4\gcd(A^2, B)^2$. Such special primes implies a very low number of reduced

forms in each class, namely there are $O(\log(p))$ reduced forms in $\mathfrak{F}_p$ in each equivalence class ([8] and [22], theorem 5.8, p. 52]). It is therefore practical to enumerate every reduced form equivalent to a given one. With a generic discriminant, the number of reduced forms per cycle would be exponential, around $O(\sqrt{p})$ (see [3]). To avoid any confusion, please note that even for a Schinzel prime, the number of classes in $\mathfrak{F}_p$ remains exponential.

– q is such that s_q is linear in q. This imply that the number of reduced forms of discriminant $N = q^2p$ in each equivalence class is at least linear in q and upper-bounded by $O(q\log(p))$, which is exponential.

The encryption of a message m works as follows: m is embedded into a (usually prime) integer $a \leqslant \sqrt{p}/2$ which satisfies some low-probability pattern, and such that q^2p is a square modulo a. This integer is expanded into a quadratic form $f_s = (a, b', c')$ of discriminant q^2p (which is not printed). The ciphertext is a random reduced form f_c equivalent to f_s (there are exponentially many). It can be generated from f_s by successive multiplications by random unimodular matrices and reductions.

The decryption algorithm lifts the ciphertext in $\mathfrak{F}_p$ and enumerate all the reduced forms equivalent to $\varphi(f_c)$, looking for the pattern. Of course, the knowledge of q is needed to compute φ. There are only $O(\log(p))$ of them. It will necessarily find it, because the (unknown) lift of $f_s \sim f_c$ is an equivalent form $\varphi(f_s) = (a, *, *)$, whose normalization $(a, *, *)$ is reduced due to the small size of a, and it satisfies the pattern by construction. Due to the small number of reduced forms, it is likely the only one of the small reduced cycle to satisfy the pattern, and the plaintext m is eventually extracted from a.

4.3 Cryptanalysis

The cryptanalysis of NICE Real presented in [6] works as follows. The authors present an algorithm inspired of Coppersmith methods (see [10,17]), which solves in polynomial time the equation $au^2 + buv + xv^2 = 0 \mod q^2$ in the variables (u, v, q) where $N = pq^2$ is known and $\max(|u|, |v|) = O(N^{1/9})$. They call this algorithm Homogeneous-Coppersmith in [6]. Their cryptanalysis of NICE Real is: Pick[3] a form g of the *principal cycle*, and try to solve the equation $g(u, v) = 0 \mod q^2$ with Homogeneous-Coppersmith. Repeat this until it finds a solution (u, v, q) and return the private key q.

The proof of the attack of [6] relies on this heuristic assumption:

Assumption 1. *The cardinality of the set* $\mathcal{A} = \{g \in \mathbb{1}_N,\ \exists(u, v)\ \max(|u|, |v|) \leqslant O(N^{\frac{1}{9}})$ *and* $g(u, v) = 0 \mod q^2\}$ *is linear in* s_q.

[3] The authors of [6] enumerates the forms sequentially, until it finds a solvable one. They need an assumption not only on the large number of such forms, but also on their regular repartition on the principal cycle. Randomizing the enumeration avoids to prove the assumption on regular repartition (Heuristic 2 in [6]), which is feasable using the distance introduced in Theorem 3, but is beyond the scope of this paper.

The authors of [6] experimentally verify this assumption. Namely, if $\bar{g}_k$ denotes the reduction of the form $g_k = (q^2, *, *)$ of the q-belt by Classical Gauss reduction. The bottom two coefficients of the reduction matrix satisfy $\bar{g}_k(\delta, -\gamma) = q^2$. Homogeneous-Coppersmith experimentally recovers $(\delta, -\gamma)$ for most of the $\bar{g}_k$ and even a few of their direct left or right neighbours on the principal cycle. This indicates that the norm of the reduction matrix is in general upper-bounded by $O(N^{1/9})$. However we also found rare cases of $\bar{g}_k$ where the norm of reduction matrix was by an order greater than $N^{1/9}$, and on which Homogeneous-Coppersmith algorithm cannot find any solution. We call these particular forms *unbalanced*, because they have in general an unusually small coefficient. The main three difficulties which prevented the authors of [6] to prove Assumption 1 were to justify that that the proportion of unbalanced forms is negligible among the set of $\{\bar{g}_k\}$, that the reduction matrix using Classical Gauss reduction is bounded by $O(N^{1/9})$, and that Classical Gauss is injective on a large enough subset of the q-belt, which prevents $\{\bar{g}_k\}$ from being too small.

Our first improvement in their analysis is to replace the Classical Gauss reduction algorithm with *Red*GL2. This allows to square-root the upper-bounds on the reduction matrix as of Theorem 2. Thus we define $\hat{g}_k$ as the reduction by *Red*GL2 of the q-belt form g_k for each k. We ensure that $\hat{g}_k$ is classically reduced and that the reduction matrix has determinant $+1$ using Lemma 1. The first point of Theorem 2 implies that the norm of the reduction matrix is in $O(N^{1/9})$ as soon as the smallest coefficient of $\hat{g}_k$ is greater than $N^{4/9}$. We can either prove that this condition is satisfied by a large proportion of the g_k , or we can also circumvent this limitation by using the second point of Theorem 2, which indicates that the size of the product $|uv|$ is always upper-bounded by $O(N^{1/6})$.

We therefore improve the Homogeneous-Coppersmith algorithm so that it also finds unbalanced solutions: namely, we design a rational variant of Boneh-Durfee-HowgraveGraham algorithm [2] which in particular solves $g(u, v) = au^2 + buv + cv^2 = 0 \mod q^2$ on (u, v, q) as soon as the product $|uv|$ is in $O(N^{2/9})$.

Our new polynomial attack on Nice Real is the following: Randomly select a form g on the principal cycle $\mathbb{1}_N$, and try to solve $g(u, v) = 0 \mod q^2$ in (u, v, q) using Rational-BonehDurfeeHowgraveGraham. Repeat until it finds a solution, and return q.

The proof of this attack works in two steps: first, we prove (in Theorem 3) that the above-defined $\hat{g}_k$ represent a non-negligible proportion of the principal cycle, and second, we prove (in Section 4.4) that Rational-BonehDurfeeHowgraveGraham finds q from any of the $\hat{g}_k$ in polynomial time.

Definition 2 (distance). *we define a notion of* distance *between two equivalent forms* $f \sim g$ *as* $\mathrm{dist}(f, g) = \min\{\log(\|\mathcal{M}\|), \ \mathcal{M} \in SL_2(\mathbb{Z})$ *and* $f.\mathcal{M} = g\}$. *Let* f, g, h *be three equivalent forms in* $\mathfrak{F}_\Delta$*, the distance function satisfies the following properties:*

1. $\mathrm{dist}(f, g) = \mathrm{dist}(g, f) \geqslant 0$
2. $\mathrm{dist}(f, g) = 0 \iff f = g$ *or* $f = g.\boldsymbol{SE}$

3. $\mathrm{dist}(f,h) \leqslant \mathrm{dist}(f,g) + \mathrm{dist}(g,h)$
4. *if* $\mathcal{M} \in SL_2(\mathbb{Z})$ *satisfies* $f.\mathcal{M} = g$ *and* $|||\mathcal{M}||| \leqslant \sqrt{\epsilon_\Delta}$, *then* $\mathrm{dist}(f,g) = \log(|||\mathcal{M}|||)$.

Proof. The first three points follow from basic properties of the induced norm, and the fact that only isometries have a unit norm. To prove the fourth statement, let U be the fundamental automorphism of f, the eigenvalues of U are ϵ_Δ and ϵ_Δ^{-1}. Any non-trivial automorphism V of f satisfies $|||V||| \geqslant \epsilon_\Delta$, because V is a non-zero power of U, and its spectral radius is a positive power of ϵ_Δ. The matrix $\mathcal{M}$ of the fourth point is necessarily the smallest transformation matrix from f to g, otherwise any matrix $X \in \mathrm{SL}_2(\mathbb{Z})$ such that $f.X = g$ and $|||X||| < |||\mathcal{M}|||$ would produce a non-trivial automorphism $\mathcal{M}X^{-1}$ of f of too small norm $|||\mathcal{M}X^{-1}||| < \epsilon_\Delta$, which is impossible. □

One of the greatest advantage of this distance is the fourth statement, which in general indicates that any polynomial transformation matrix is necessarily the smallest one. This allows to efficiently lower-bound a distance. As shown in the proof, it is essential that the group of automorphism is cyclic, the fourth statement would be false on $\mathrm{GL}_2(\mathbb{Z})$. The authors of [6] used another distance between (f,g), which could have been formalized as the smallest $k \in \mathbb{N}$ such that there exists $h_1, \ldots, h_k$ such that $\prod_{i=1}^{k} \boldsymbol{SET}(h_i)$ transforms f into g or $g.\boldsymbol{SE}$. Inside the reduced cycle, this corresponds to *Shanks distance* [20]. Unfortunately, it does not satisfy any equivalent of the fourth point: there is no way to efficiently verify that a given distance, as small as it could be, is correct. All the variants we found of this distance, which aims to approximate this statement, based either on the logarithms of the h_i or some maximum norms, break the positive definiteness or the triangular inequality. This explains why we do not base our proof on Shanks distance and introduce our own instead.

Theorem 3. *Given a NICE modulus* $N = pq^2$, *the set* $\mathcal{A}' = \{\hat{g}_k = RedGL2(g_k),\ k \in [1, \ldots s_q - 1]\}$ *of the reduced of the q-belt has at least* $K.s_q$ *elements for some constant* $K > 0$.

Proof. We now call U_p the fundamental automorphism of the principal form of $\mathfrak{F}_p$. We verify that $|||U_p^j||| \leqslant 2(\epsilon_p^j + \epsilon_p^{-j})$ and that for all i, j, $Q_{k_i}^{-1} U_p^j Q_{k_{i+j}}$ transforms g_i into g_{i+j}. Its norm is bounded by $\frac{1}{q}|||Q_{k_i}||| \cdot |||U^j||| \cdot |||Q_{k_{i+j}}||| < 4q(\epsilon_p^j + \epsilon_p^{-j})$.

Due to point 4, for all $j \in [1, (s_q/2) - 2]$, the distance $\mathrm{dist}(g_i, g_{i+j}) = \log(|||Q_{k_i}^{-1} U_p^j Q_{k_{i+j}}|||)$ is greater than $j\log(\epsilon_p) - \log(2q)$. By Theorem 2, the norm of the reduction matrix from a g_i to $\hat{g}_i$ is upper-bounded by $2 \cdot 21q^2/\sqrt{N} = 42q/\sqrt{p}$, and it follows that $\mathrm{dist}(\hat{g}_i, \hat{g}_{i+j}) \geqslant j\log(\epsilon_p) - \log(3528q^3p)$. For this reason, if $j > \log(3528q^3p)/\log(\epsilon_p)$, then $\mathrm{dist}(\hat{g}_i, \hat{g}_{i+j}) > 0$ and $\hat{g}_i \neq \hat{g}_{i+j}$. Using the NICE parameters, one has $\log(3528q^3p)/\log(\epsilon_p) < 3$, thus the forms $\hat{g}_1, \hat{g}_4, \hat{g}_7, \ldots, \hat{g}_{3n+1}$ are distinct (with $n \leqslant s_q/6$). □

4.4 Rational Improvement of the Boneh-Durfee-HowgraveGraham's Algorithm

In this section, we describe our Rational-BonehDurfeeHowgraveGraham algorithm as a variant of Boneh Durfee Howgrave-Graham algorithm [2] solving rational linear equations $u/v - C = 0 \mod q$ in the variables (u, v, q) when a multiple $N = pq^r$ is known. The description of Rational-BonehDurfeeHowgraveGraham is summarized in Algorithm 2. Among others, it can be used to solve all the equations $\hat{g}_k(u, v) = au^2 + buv + cv^2 = 0 \mod q^2$ of discriminant pq^2 of the previous section, because they are equivalent to $u/v + b/2a = 0 \mod q$. Since the solution we are looking for satisfies $|uv| = O(N^{1/6})$, the following Theorem 4 proves that Rational-BonehDurfeeHowgraveGraham finds all solutions $|uv| = O(N^{2/9})$, and concludes the proof of our new attack on Nice Real.

More generally, given a polynomial P, the technique due to Boneh Durfee Howgrave-Graham transforms the equation $P(u/v) = 0 \mod q$, into a lattice L of dimension m and bounded determinant, and whose short vectors are orthogonal to the integer vector $S = (u^m, u^{m-1}v, ..., uv^{m-1}, v^m)$. The solutions u and v can be extracted from any of those short lattice vectors. This lattice is described by a basis B, whose rows contain the coefficients of $(m-1)$-degree polynomials having u/v as a root modulo a power of q. When u and v have approximately the same size (like in Homogeneous-Coppersmith of [6]), the celebrated LLL reduction algorithm on B outputs directly the desired vector orthogonal to S. Otherwise, when u and v are unbalanced, say for instance that u is 1000 times larger than v, one first needs to re-balance the lattice by multiplying each i-th column by C^i, where C is close to 1000, and only then reduce the basis. The original Boneh-Durfee-HowgraveGraham's algorithm, which interests in integer solutions (arbitrary u and $v = 1$), follows the above rule: the lattice basis which is actually LLL-reduced is the basis of Homogeneous-Coppersmith where each i-th column has been multiplied by X^i, where X is a power of 2 just larger than the solution u. More generally, if we don't know the relative balance between u and v but only know that the size of uv is n-bits, then we can test the n possible powers of two sequentially within a linear-factor overhead. Besides, we remark that instead of multiplying the columns of the input Homogeneous-Coppersmith basis by $(1, 2, 4, ..., 2^m)$, we describe the exact same lattice by multiplying the columns of the LLL-reduced basis, and the second one is almost reduced (LLL terminates in a very few steps). Thus after the reduction of the first Homogeneous-Coppersmith basis, one obtains all the other possible balances of u and v for free.

Theorem 4. *Given any integer $N = pq^r$ (where p and q are unknown), and a bound $\beta < \frac{1}{4} \cdot q^{\log(q^r)/\log(N)}$, Algorithm 2 terminates in polynomial time, and finds a solution (if it exists) of the equation $\frac{u}{v} = c \mod q$ where (u, v) are unknown integers satisfying $|uv| < \beta$.*

Proof. Let $(U, V) \in \mathbb{R}^2$ such that $|u| \leqslant U$ and $v \leqslant V$. We use the same parameters $m \in \mathbb{N}\backslash\{0\}$ and $t = \left\lfloor \frac{(m+1)\cdot\log(q^r)}{\log(N)} \right\rfloor$.

We denote by $\mathbb{R}_m[X, Y]$ the span of homogeneous polynomials of degree m, and we define the isomorphism $\varphi : \mathbb{R}_m[X, Y] \to \mathbb{R}^{m+1}$ which computes

Algorithm 2. Rational Boneh-Durfee-HowgraveGraham

Input: An integer $N \in \mathbb{N}$ of the form pq^r (p and q are unknown), an integer $c \in [0, N-1]$ and a bound $\beta < \frac{1}{4} \cdot q^{\log(q^r)/\log(N)}$

Output: $(u, v) \in \mathbb{N}^3$ such that $\frac{u}{v} = c \mod q$ and $|u| \cdot |v| < \beta$ if it exists

1: Choose the smallest m such that $\left(N^{\frac{1}{2}+\frac{1}{8r}}(m+1)^{\frac{1}{2}}\right)^{\frac{1}{m+1}} < 1.5$, and set $t = \left\lfloor \frac{(m+1)\cdot\log(q^r)}{\log(N)} \right\rfloor$.
2: Compute the family $P_k(X,Y) = N^{\lceil \frac{t-k}{r} \rceil} \cdot (X - cY)^k \cdot Y^{m-k}$ for $k = [0..m]$
3: **for** $l = 0$ to $\lfloor \log_2(\beta) \rfloor$ **do**
4: $\quad U = 2^l$; $V = \lfloor \beta / 2^l \rfloor$
5: $\quad$ Express (or update) the family $(P_k)_{k\in 0..m}$ on the monomial basis $(\frac{X^k Y^{m-k}}{U^k V^{m-k}})_{k=0..m}$, and form a matrix $B \in \mathcal{M}_{m+1}(\mathbb{Z})$
6: $\quad$ LLL-reduce B, and call $(\alpha_0, \ldots, \alpha_m)$ the first vector
7: $\quad$ **for** each rational root $\frac{u}{v}$ of $R(X) = \sum_{k=0}^{m} \frac{\alpha_k}{U^k V^{m-k}} X^k = 0$ **do**
8: $\quad\quad$ if $|uv| \leqslant \beta$ and $\gcd(u - cv, N)$ is non-trivial return (u, v)
9: $\quad$ **end for**
10: **end for**

the coordinates of a polynomial on the basis $(\frac{X^k Y^{m-k}}{U^k V^{m-k}})_{k=0..m}$. For instance, $\varphi(X^k Y^{m-k}) = U^k V^{m-k} \boldsymbol{e}_k$ where $\boldsymbol{e}_k$ is the k-th canonical basis vector. Let $(P_k)_{k\in[0..m]}$ be the family $P_k(X,Y) = N^{\lceil \frac{t-k}{r} \rceil} \cdot (X - cY)^k \cdot Y^{m-k} \in \mathbb{R}_m[X,Y]$. By construction, any integer linear combination $R \in \sum_{k=0}^{m} \mathbb{Z} \cdot P_k$ satisfy $R(u,v) = 0 \mod q^t$ and $|R(u,v)| \leqslant \sqrt{m+1} \cdot ||\varphi(R)||_2$ (using Cauchy-Schwartz inequality). We now suppose that $\varphi(R)$ is a short vector of the lattice generated by the (triangular) basis $B = (\varphi(P_k))_{k\in[1,m]}$. By that, we mean $||\varphi(R)||_2 \leqslant (1.08)^{m+1} \det(B)^{1/(m+1)}$. Such a vector can be found by running the LLL algorithm on the lattice basis B (see [16]). The remainder of the proof is just a formal verification that when m grows, $\det(B)$ is small enough to guaranty that $|R(u,v)| < q^t$, and therefore that $R(u,v) = 0$ (in $\mathbb{Z}$). Since R is homogeneous, this allows to recover u and v. □

5 Conclusion

We saw that reduction algorithms are conceptually simpler to study in $\mathrm{GL}_2(\mathbb{Z})$, because we mostly manipulate only positive matrices, which are easy to bound. The precision of our analysis, in the worst case and also in the average case, allows us to fully prove a lattice-based total-break attack against Nice cryptosystems [12,13], which is unusual in the history of lattice based cryptology. A further lead would be to extend these results on the reduction of the forms in higher dimension.

Acknowledgements. We would like to thank Fabien Laguillaumie and Guilhem Castagnos for useful discussions and valuable comments on this paper.

References

1. Biehl, I., Buchmann, J.: An analysis of the reduction algorithms for binary quadratic forms. In: Voronoi's Impact on Modern Science, pp. 71–98 (1999)
2. Boneh, D., Durfee, G., Howgrave-Graham, N.A.: Factoring $n = p^r q$ for large r. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, p. 326. Springer, Heidelberg (1999)
3. Buchmann, J., Thiel, C., Williams, H.: Short representation of quadratic integers. Proc. of CANT 1992, Math. Appl. 325, 159–185 (1995)
4. Buchmann, J., Vollmer, U.: Binary Quadratic Forms An Algorithmic Approach. Springer, Heidelberg (2007)
5. Buell, D.A.: Binary Quadratic Forms Classical Theory and Modern Computations. Springer, Heidelberg (1989)
6. Castagnos, G., Joux, A., Laguillaumie, F., Nguyen, P.Q.: Factoring pq^2 with quadratic forms: Nice cryptanalyses. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 469–486. Springer, Heidelberg (2009)
7. Castagnos, G., Laguillaumie, F.: On the security of cryptosystems with quadratic decryption: The nicest cryptanalysis. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 260–277. Springer, Heidelberg (2010)
8. Cheng, K.H.F., Williams, H.C.: Some results concerning certain periodic continued fractions. Acta Arith. 117, 247–264 (2005)
9. Cohen, H.: A Course in Computational Algebraic Number Theory, 2nd edn. Springer, Heidelberg (1995)
10. Coppersmith, D.: Small solutions to polynomial equations, and low exponent RSA vulnerabilities. J. of Cryptology 10(4), 233–260 (1997); Revised version of two articles from Eurocrypt 1996 (1996)
11. Gauss, C.F.: Disquisitiones Arithrneticae. PhD thesis (1801)
12. Hartmann, M., Paulus, S., Takagi, T.: NICE - New Ideal Coset Encryption. In: Koç, Ç.K., Paar, C. (eds.) CHES 1999. LNCS, vol. 1717, pp. 328–339. Springer, Heidelberg (1999)
13. Jacobson, M.J., Scheidler, R., Weimer, D.: An adaptation of the NICE cryptosystem to real quadratic orders. In: Vaudenay, S. (ed.) AFRICACRYPT 2008. LNCS, vol. 5023, pp. 191–208. Springer, Heidelberg (2008)
14. Lagarias, J.C.: Worst-case complexity bounds for algorithms in the theory of integral quadratic forms. Journal of Algorithm 1, 142–186 (1980)
15. Lagrange, J.L.: Recherches d'arithmétique. Nouveaux Mémoires de l'Académie de Berlin (1773)
16. Lenstra, A.K., Lenstra Jr., H.W., Lovász, L.: Factoring polynomials with rational coefficients. Mathematische Ann. 261, 513–534 (1982)
17. May, A.: Using LLL-reduction for solving RSA and factorization problems: A survey. In: Nguyen, P., Vallee, B. (eds.) The LLL algorithm, survey and Applications, Information Security and Cryptography, pp. 315–348 (2010)
18. Nguyen, P.Q., Stehlé, D.: Low-dimensional lattice basis reduction revisited (extended abstract). In: Proceedings of ANTS VI. LNCS, Springer, Heidelberg (2004)
19. Schinzel, A.: On some problems of the arithmetical theory of continued fractions. Acta Arithmetica 6, 393–413 (1961)
20. Shanks, D.: The infrastructure of a real quadratic field and its applications. In: Proc. NTC 1992, pp. 217–224 (1972)
21. Vallee, B., Vera, A.: Lattice reduction in two dimensions: Analyses under realistic probalistic models. In: Proc. of AofA 2007, DMTCS AH, pp. 181–216 (2007)
22. Weimer, D.: An Adaptation of the NICE Cryptosystem to Real Quadratic Orders, Master's thesis. PhD thesis, Technische Universitat Darmstadt (2004)

Practical Improvements to Class Group and Regulator Computation of Real Quadratic Fields

Jean-François Biasse[1] and Michael J. Jacobson, Jr.[2,⋆]

[1] École Polytechnique, 91128 Palaiseau, France
biasse@lix.polytechnique.fr
[2] Department of Computer Science, University of Calgary
2500 University Drive NW, Calgary, Alberta, Canada T2N 1N4
jacobs@cpsc.ucalgary.ca

Abstract. We present improvements to the index-calculus algorithm for the computation of the ideal class group and regulator of a real quadratic field. Our improvements consist of applying the double large prime strategy, an improved structured Gaussian elimination strategy, and the use of Bernstein's batch smoothness algorithm. We achieve a significant speed-up and are able to compute the ideal class group structure and the regulator corresponding to a number field with a 110-decimal digit discriminant.

1 Introduction

Computing invariants of real quadratic fields, in particular the ideal class group and the regulator, has been of interest since the time of Gauss, and today has a variety of applications. For example, solving the well-known Pell equation is intimately linked to computing the regulator, and integer factorization algorithms have been developed that make use of this invariant. Public-key cryptosystems have also been developed whose security is related to the presumed difficulty of these computational tasks. See [16] for details.

The fastest algorithm for computing the ideal class group and regulator in practice is a variation of Buchmann's index-calculus algorithm [6] due to Jacobson [14]. The algorithm on which it is based has subexponential complexity in the size of the discriminant of the field. The version in [14] includes several practical enhancements, including the use of self-initialized sieving to generate relations, a single large-prime variant (based on that of Buchmann and Düllman [7] in the case of imaginary quadratic fields), and a practical version of the required linear algebra. This approach proved to work well, enabling the computation of the ideal class group and regulator of a real quadratic field with a 101-decimal digit discriminant [15]. Unfortunately, both the complexity results of Buchmann's algorithm and the correctness of the output are dependent on the Generalized Riemann Hypothesis (GRH). Nevertheless, for fields with large discriminants, this approach is the only one that works.

⋆ The second author is supported in part by NSERC of Canada.

G. Hanrot, F. Morain, and E. Thomé (Eds.): ANTS-IX 2010, LNCS 6197, pp. 50–65, 2010.

Recently, Biasse [4] presented practical improvements to the corresponding algorithm for imaginary quadratic fields. These included a double large prime variant and improved algorithms for the required linear algebra. The resulting algorithm was indeed faster then the previous state-of-the-art [14], and enabled the computation of the ideal class group of an imaginary quadratic field with 110 decimal digit discriminant.

In this paper, we describe a number of practical improvements to the index-calculus algorithm for computing the class group and regulator of a real quadratic field. In addition to adaptations of Biasse's improvements in the imaginary case, we have found some modifications designed to improve the regulator computation part of the algorithm. We also investigate applying an idea of Bernstein [3] to factor residues produced by the sieve using a batch smoothness test. Extensive computations demonstrating the effectiveness of our improvements are presented, including the computation of class group and regulator of a real quadratic field with 110 decimal digit discriminant.

This paper is organized as follows. In the next section, we briefly recall the required background of real quadratic fields, and give an overview of the index-calculus algorithm using self-initialized sieving. Our improvements to the algorithm are described in Section 3, followed by numerical results in Section 4.

2 Real Quadratic Fields

We present an overview of required concepts related to real quadratic fields and the index-calculus algorithm for computing invariants. For more details, see [16].

Let $K = \mathbb{Q}(\sqrt{\Delta})$ be the real quadratic field of discriminant Δ, where Δ is a positive integer congruent to 0 or 1 modulo 4 with Δ or $\Delta/4$ square-free. The integral closure of $\mathbb{Z}$ in K, called the maximal order, is denoted by $\mathcal{O}_\Delta$. An interesting aspect of real quadratic fields is that their maximal orders contain infinitely many non-trivial units, i.e., units that are not roots of unity. More precisely, the unit group of $\mathcal{O}_\Delta$ consists of an order 2 torsion subgroup and an infinite cyclic group. The smallest unit greater than 1, denoted by ε_Δ, is called the fundamental unit. The regulator of $\mathcal{O}_\Delta$ is defined as $R_\Delta = \log \varepsilon_\Delta$.

The fractional ideals of K play an important role in the index-calculus algorithm described in this paper. In our setting, a fractional ideal is a rank 2 $\mathbb{Z}$-submodule of K. Any fractional ideal can be represented as

$$\mathfrak{a} = \frac{s}{d}\left[a\mathbb{Z} + \frac{b+\sqrt{\Delta}}{2}\mathbb{Z}\right] \quad ,$$

where $a, b, s, d \in \mathbb{Z}$ and $4a \mid b^2 - \Delta$. The integers a, s, and d are unique, and b is defined modulo $2a$. The ideal $\mathfrak{a}$ is said to be primitive if $s = 1$, and $d\mathfrak{a} \subseteq \mathcal{O}_\Delta$ is integral. The norm of $\mathfrak{a}$ is given by $\mathcal{N}(\mathfrak{a}) = as^2/d^2$.

Ideals can be multiplied using Gauss's composition formulas for indefinite binary quadratic forms. Ideal norm respects ideal multiplication, and the set

$\mathcal{I}_\Delta$ forms an infinite abelian group with identity $\mathcal{O}_\Delta$ under this operation. The inverse of $\mathfrak{a}$ is

$$\mathfrak{a}^{-1} = \frac{d}{sa}\left[a\mathbb{Z} + \frac{-b+\sqrt{\Delta}}{2}\mathbb{Z}\right] .$$

The group $\mathcal{I}_\Delta$ is generated by the prime ideals of $\mathcal{O}_\Delta$, namely those integral ideals of the form $p\mathbb{Z}+(b_p+\sqrt{\Delta})/2\mathbb{Z}$ where p is a prime that is split or ramified in K. As $\mathcal{O}_\Delta$ is a Dedekind domain, the integral part of any fractional ideal can be factored uniquely as a product of prime ideals. To factor $\mathfrak{a}$, it suffices to factor $\mathcal{N}(\mathfrak{a})$ and, for each prime p dividing the norm, determine whether the prime ideal $\mathfrak{p}$ or $\mathfrak{p}^{-1}$ divides $\mathfrak{a}$ according to whether $b \equiv b_p$ or $-b_p$ modulo $2p$.

The ideal class group, denoted by Cl_Δ, is the factor group $\mathcal{I}_\Delta/\mathcal{P}_\Delta$, where $\mathcal{P}_\Delta \subseteq \mathcal{I}_\Delta$ is the subgroup of principal ideals. The class group is finite abelian, and its order is called the class number, denoted by h_Δ. By computing the class group we mean computing the elementary divisors $m_1, \ldots, m_l$ with $m_{i+1} \mid m_i$ for $1 \leq i < l$ such that $Cl_\Delta \cong \mathbb{Z}/m_1\mathbb{Z} \times \cdots \times \mathbb{Z}/m_l\mathbb{Z}$.

2.1 The Index-Calculus Algorithm

Like other index-calculus algorithms, the algorithm for computing the class group and regulator relies on finding certain smooth quantities, those whose prime divisors are all small in some sense. In the case of quadratic fields, one searches for smooth principal ideals for which all prime ideal divisors have norm less than a given bound B_1. The set of prime ideals $\mathcal{B} = \{\mathfrak{p}_1, \ldots, \mathfrak{p}_n\}$ with $\mathcal{N}\mathfrak{p}_i \leq B_1$ is called the factor base.

A principal ideal $(\alpha) = \mathfrak{p}_1^{e_1} \ldots \mathfrak{p}_n^{e_n}$ with $\alpha \in K$ that factors completely over the factor base yields the relation $(e_1, \ldots, e_n, \log|\alpha|)$. The key to the index-calculus algorithm is the fact, proved by Buchmann [6], that the set of all relations forms a sublattice $\Lambda \subset \mathbb{Z}^n \times \mathbb{R}$ of determinant $h_\Delta R_\Delta$ provided that the prime ideals in the factor base generate Cl_Δ. This follows, in part, due to the fact that L, the integer component of Λ, is the kernel of the homomorphism from $\mathbb{Z}^n$ to Cl_Δ given by $\mathfrak{p}_1^{e_1} \ldots \mathfrak{p}_n^{e_n}$ for $(e_1, \ldots, e_n) \in \mathbb{Z}^n$. If $\mathfrak{p}_1, \ldots, \mathfrak{p}_n$ generate Cl_Δ, then this homomorphism is surjective, and the homomorphism theorem then implies that $\mathbb{Z}^n/L \cong Cl_\Delta$.

The main idea behind the index-calculus algorithm is to find random relations until they generate the entire relation lattice Λ. Let Λ' denote the sublattice of Λ generated by the relations that have been computed. To determine whether $\Lambda' = \Lambda$, one computes an approximation h^* of $h_\Delta R_\Delta$ such that $h^* < h_\Delta R_\Delta < 2h^*$. The value h^* is obtained by approximating the L-function $L(1, \chi_\Delta)$, where χ_Δ denotes the Kronecker symbol (Δ/p), and applying the analytic class number formula. If $\Lambda' \subset \Lambda$, then $\det(\Lambda')$ is a integer multiple of $h_\Delta R_\Delta$. Thus, $\Lambda' = \Lambda$ as soon as $\det(\Lambda') < 2h^*$, because $h_\Delta R_\Delta$ is the only integer multiple of itself in the interval $(h^*, 2h^*)$.

As described in [14], an adaptation of the strategy used in the self-initialized quadratic sieve (SIQS) factoring algorithm is used to compute relations. First, compute the ideal $\mathfrak{a} = \mathfrak{p}_1^{e_1} \dots \mathfrak{p}_n^{e_n} = (1/d)[a\mathbb{Z} + (b + \sqrt{\Delta})/2\mathbb{Z}]$ with $\mathcal{N}(\mathfrak{a}) = a/d^2$. Let $\alpha = (ax + (b + \sqrt{\Delta})/2y)/d$ with $x, y \in \mathbb{Z}$ be an arbitrary element in $\mathfrak{a}$. Then

$$\mathcal{N}(\alpha) = \frac{1}{d^2}\left(ax + \frac{b+\sqrt{\Delta}}{2}y\right)\left(ax + \frac{b-\sqrt{\Delta}}{2}y\right) = (a/d^2)(ax^2 + bxy + cy^2)$$

where $c = (b^2 - \Delta)/(4a)$. Because ideal norm is multiplicative, there exists an ideal $\mathfrak{b}$ with $\mathcal{N}(\mathfrak{b}) = ax^2 + bxy + cy^2$ such that $(\alpha) = \mathfrak{a}\mathfrak{b}$. Thus, finding x and y such that $\mathcal{N}(\mathfrak{b})$ factors over the norms of the prime ideals in the factor base yields a relation. Such x and y can be found by sieving the polynomial $\varphi(x, y) = ax^2 + bxy + cy^2$, and a careful selection of the ideals $\mathfrak{a}$ yields a generalization of self-initialization, in which the coefficients of the sieving polynomials and their roots modulo the prime ideal norms can be computed quickly. In practice, we use $\varphi(x, 1)$ for sieving, so that the algorithm resembles the SIQS more closely. For more details, see [14] or [16].

The determinant of the relation lattice Λ' is computed in two stages. The first step is to compute the determinant of the integer part of this sublattice by finding a basis in Hermite normal form (HNF). Once Λ' has full rank, the determinant of this basis is computed as the product of the diagonal elements in a matrix representation of the basis vectors. The group structure is then computed by finding the Smith normal form of this matrix. The real part of $\det(\Lambda')$, a multiple of the regulator R_Δ, is computed by first finding a basis of the kernel of the matrix consisting of the integer parts of the relations. Every vector $(k_1, \dots, k_m) \in \mathbb{Z}^m$ in the kernel corresponds to a multiple of the regulator computed with $mR_\Delta = k_1 \log|\alpha_1| + \dots + k_m \log|\alpha_m|$. The "real gcd" of the multiples $m_1 R_\Delta, \dots, m_n R_\Delta$ computed from each basis vector of the kernel, defined as $\gcd(m_1, \dots, m_n)R_\Delta$, is then the real part of $\det(\Lambda')$. An algorithm of Maurer [21] can be used to compute the real gcd efficiently and with guaranteed numerical accuracy given explicit representations of the α_i and the kernel vectors.

As mentioned in the introduction, the correctness of this algorithm depends on the truth of the Generalized Riemann Hypothesis. In fact, the GRH must be invoked in two places. The first is to compute a sufficiently accurate approximation h^* of $h_\Delta R_\Delta$ via a method due to Bach [2]. Without the GRH, an exponential number of terms in the Euler product used to approximate $L(1, \chi_\Delta)$ must be used (see, for example, [20]). The second is to ensure that the factor base generates Cl_Δ. Without the GRH, an exponential size factor base is required, whereas by a theorem of Bach [1] the prime ideals of norm less than $6\log(\Delta)^2$ suffice. In practice, an even smaller factor base is often used, but in that case, the factor base must be verified by showing that every remaining prime ideal with norm less than Bach's bound can be factored over the ideals in the factor base.

3 Practical Improvements

In this section, we describe our practical improvements for computing the class group structure and the regulator of a the real quadratic field. Some of these improvements, such as the double large prime variant and structured Gaussian elimination, were used in [4] for the simpler case of imaginary quadratic number fields. On the other hand, the batch smoothness test and system solving based methods for computing the regulator had never been implemented in the context of number fields before.

3.1 Relation Collection

Improving the relation collection phase allows us to speed up every other stage of the algorithm. Indeed, the faster the relations are found, the smaller the factor base can be, thus reducing the dimensions of the relation matrix and the time taken by the linear algebra phase. In addition, the verification phase also relies on our ability to find relations and therefore benefits from improvements to the relation collection phase. Throughout the rest of the paper, M denotes the relation matrix, the matrix whose rows are the integer parts of the relations.

Large prime variants. The large prime variants were developed in the context of integer factorization to speed up the relation collection phase in both the quadratic sieve and the number field sieve. A single large prime variant was described by Buchmann and Düllman [7] for computing the class group of an imaginary quadratic field, and adapted to the real case by Jacobson [14]. Biasse [4] described how the double large prime strategy could be using in the imaginary case, and obtained a significant speed-up.

The idea is to keep relations involving one or two extra primes not in the factor base of norm less than $B_2 \geq B_1$. These relations thus have the form

$$(\alpha) = \mathfrak{p}_1^{e_1} \dots \mathfrak{p}_n^{e_n} \mathfrak{p} \quad \text{and} \quad (\alpha) = \mathfrak{p}_1^{e_1} \dots \mathfrak{p}_n^{e_n} \mathfrak{p}\mathfrak{p}'$$

for $\mathfrak{p}_i$ in $\mathcal{B}$, and for $\mathfrak{p}, \mathfrak{p}'$ of norm less than B_2. We will refer to these types of partial relations as 1-partial relations and 2-partial relations, respectively. Keeping partial relations only involving one large prime is the single large prime variant, whereas keeping those involving one or two is the double large prime variant which was first described by Lenstra and Manasse [17]. We do not consider the case of more large primes, but it is a possibility that has been studied in the context of factorization [10].

Partial relations may be identified as follows. Let m be the remainder of $\varphi(x, 1)$ after the division by all primes $p \leq B_1$, and assume that $B_2 < B_1^2$. If $m = 1$ then we have a full relation. If $m \leq B_2$ then we have a 1-partial relation. We can see here that detecting 1-partial relations is almost for free. If we also intend to collect 2-partial relations then we have to consider the following possibilities:

1. $m > B_2^2$;
2. m is prime and $m > B_2$;
3. m is prime and $m \leq B_2$;
4. m is composite and $B_1^2 < m \leq B_2^2$.

In Cases 1 and 2 we discard the relation. In Case 3 we have a 1-partial relation, and in Case 4 we have $m = pp'$ where $p = \mathcal{N}(\mathfrak{p})$ and $p' = \mathcal{N}(\mathfrak{p}')$. Cases 1, 2, and 3 can be checked very easily, but if none are satisfied we need to factor m in order to determine whether Case 4 is satisfied. We used Milan's implementation of the SQUFOF algorithm [22] based on the theoretical work of [12] to factor the m values produced.

Even though we might have to factor the remainder, partial relations are found much faster than full relations. However, the dimensions of the resulting matrix are much larger, thus preventing us from running the linear algebra phase directly on the resulting relation matrix. In addition, we have to find many more relations since we have to produce a full rank matrix. We will see in §3.2 how to reduce the dimensions of the relation matrix using Gaussian elimination techniques.

Batch smoothness test. After detecting potential candidates for smooth integers via the SIQS, one has to certify their smoothness. In [4,14], this was done by trial division with the primes in the factor base. The time taken by trial division can be shortened by using Bernstein's batch smoothness test [3], which uses a product tree structure and modular arithmetic to factor a batch of residues simultaneously in time $O\left(b(\log b)^2 \log\log b\right)$ where b is the total number of input bits.

Instead of testing the smoothness of every potential candidate as soon as they are discovered, we rather stored them and tested them at the same time using Bernstein's method as soon their number exceeded a certain limit. This improvement has an effect that is all the more important when the time spent in the trial division is long. In our algorithm, this time mostly depends on the tolerance value T, a parameter used to control the number of candidates yielded by the sieve for smoothness testing.

3.2 Structured Gaussian Elimination

As mentioned in §2.1, in order to determine whether the computed relations generate the entire relation lattice, we need to compute the HNF basis of the sublattice they generate. This can be done by putting the integer components of the relations as rows in a relation matrix, and computing the HNF.

The first step when using large primes is to compute full relations from all of the partial relations. Traditionally, rows were recombined to give full relations as follows. In the case of 1-partial relations, any pair of relations involving the same large prime $\mathfrak{p}$ were recombined into a full relation. In the case of 2-partial relations, Lenstra [17] described the construction of a graph whose vertices were the relations and whose edges linked vertices having one large prime in common.

Finding independent cycles in this graph allows us to recombine partial relations into full relations.

In this paper, we instead follow the approach of Cavallar [8], developed for the number field sieve, and adapted by the first author to the computation of ideal class group structures in imaginary quadratic number fields [4], which uses Gaussian elimination on columns. The ideas is to eliminate columns using structured Gaussian strategies until the dimensions of the matrix are small enough to allow the computation of the HNF with standard algorithms.

Let us recall a few definitions. First, subtracting two rows is called *merging*. If two relations corresponding to rows r_1 and r_2 share the same prime $\mathfrak{p}$ with coefficients c_1 and c_2 respectively, then multiplying r_1 by c_2 and r_2 by c_1 and merging is called *pivoting*. Finally, finding a sequence of pivots leading to the elimination of a column of Hamming weight k is a k-way merge.

We aim to reduce the dimensions of the relation matrix by performing k-way merges on the columns of weight $k = 1, \ldots, w$ in increasing order for a certain bound w. To limit the growth of the density and of the size of the coefficients induced by these operations, we used optimized pivoting strategies. In what follows we describe an algorithm performing k-way merges to minimize the growth of both the density and the size of the coefficients, thus allowing us to go deeper in the elimination process and delay the explosion of the coefficients.

As in [4], we define a cost function C mapping rows onto the integers. The one used in [4] satisfied

$$C(r) = \sum_{1 \leq |e_i| \leq Q} 1 + c \sum_{|e_j| > Q} 1, \tag{1}$$

where c and Q are positive numbers, and $r = [e_1, \ldots, e_n]$ is a row corresponding to $(\alpha) = \prod_i \mathfrak{p}_i^{e_i}$. This way, the heaviest rows are those which have a high density and large coefficients. In our experiments for this work, we used a different cost function, see §4.1. Then, to perform a k-way merge on a given column, we construct a complete graph $\mathcal{G}$ of size k such that

- the vertices are the rows r_i, and
- every edge linking r_i and r_j has weight $C(r_{ij})$, where r_{ij} is obtained by pivoting r_i and r_j.

Finding the best sequence of pivots with respect to the chosen cost function C is equivalent to finding the minimum spanning tree $\mathcal{T}$ of $\mathcal{G}$, and then recombining every row r with its parent starting with the leaves of $\mathcal{T}$.

Unlike in [4], we need to keep track of the permutations we apply to the relation matrix, and of the empty columns representing primes of norm less than $6 \log^2 \Delta$. This will be required for the regulator computation part of the algorithm described next.

3.3 Regulator Computation

As mentioned in §2.1, the usual way to compute the regulator is to find a basis of the kernel of the relation matrix, compute integer multiples of the regulator

from these basis vectors, and compute their real gcd using Maurer's algorithm [21]. If $\det \Lambda' > 2h^*$, then either the class number or regulator computed is too large, and we need to find extra relations corresponding to new generators, and new kernel vectors involving them.

In this section, we describe a way of taking advantage of the large number of generators involved in the different partial relations. Indeed, the dimensions of the relation matrix before the Gaussian elimination stage is much larger than in the base scenario and thus involves more generators. Consequently, given a set of $k \leq \dim(\ker M)$ kernel vectors $(u_1^j, \ldots, u_n^j)_{j \leq k}$, the probability that the corresponding elements

$$v_j := u_1^j \log |\alpha_1| + \ldots + u_n^j \log |\alpha_n| \ ,$$

where α_i is the generator of the i-th relation, can be recombined into R is much larger. On the other hand, the dimensions of the matrix prevents us from running a kernel computation directly after the relation collection phase. Thus, rather than attempting to compute the kernel, we use a method similar to that of Vollmer [24] based on solving linear systems.

The first step of our algorithm consists of putting the matrix in a pseudo-lower triangular form using a permutation obtained during the Gaussian elimination phase. Indeed, as part of this computation we obtain a unimodular matrix $U \in \mathbb{Z}^{n \times n}$ such that

$$UM = \left(\begin{array}{c:cc} A & (0) & \\ \hdashline & 1 & (0) \\ (*) & \ddots & \\ & (*) & 1 \end{array} \right) .$$

Thus, solving a linear system of the form $xM = b$ for a vector $b \in \mathbb{Z}^m$ boils down to solving a system of the form $x'A = b'$, then doing a trivial descent through the diagonal entries which equal 1 and finally permuting back the coefficients using U. To solve the small linear systems, we used the algorithm `certSolveRedLong` from the IML library [9]. It takes a single precision dense representation of A and returns an LLL-reduced solution.

Once M is in pseudo-lower triangular form, we draw a set of relations $r_1, \ldots r_d$ which are not already rows of M, and for each r_i, $i \leq d$, we solve the system $x_i A = r_i$. We then augment M with the rows r_i for $i \leq d$ and the vectors x_i with d extra coordinates, which are all set to zero except for the i-th which is set to -1.

$$M' := \begin{pmatrix} M \\ \cdots\cdots \\ r_i \end{pmatrix} \quad x'_i := \left(x_i \,\vdots\, 0 \dots 0 \; -1 \; 0 \dots 0 \right).$$

We clearly have $x'_i M' = 0$ for $i \leq d$, and the x'_i can be used to find a multiple of R_Δ as described in §2.1.

4 Numerical Results

In this section, we give numerical results showing the impact of our improvements. For each timing, we specify the architecture used. All the timings were obtained with our code in C++ based on the libraries GMP [11], NTL [23], IML [9] and Linbox [19]. All timings are in CPU seconds.

4.1 Comparative Timings

The state of the art concerning class group and regulator computation was established in [14], where all the timings were obtained with the SPARCStation II architecture. In addition, most of the code used at the time is unavailable now, including the HNF computation algorithm. Thus, providing a meaningful comparison between our methods and those of [14] is difficult. We chose to implement the HNF computation algorithm in a way that resembles the one of [14], but takes advantage of the libraries available today for computing the determinant and the modular HNF. We used this implementation in each different scenario. The relation collection phase is easier to compare, since our method relies on SIQS.

In the following, we will refer to the base case as the strategy consisting of finding the relation matrix without using the large prime variants or the smoothness batch test, and calculating the regulator by computing its kernel with the algorithm `nullspaceLong` from IML library. It differs from the 0 large prime case (0LP) where we use the algorithm described in §3.3 for computing the regulator, along with a relation collection phase that does not use large primes. We also denote the 1 large prime scenario by 1LP, the 2 large primes by 2LP and 2LP Batch when using batch smoothness test.

Relation collection phase. In Table 1, we give the time taken to collect all necessary relations. Without large primes, we collected $|\mathcal{B}| + 100$ relations, whereas when we allow large primes we need to collect enough relations to ensure that the number of rows is larger than the number of non-empty columns. We used a 2.4 GHz Opteron with 16GB of memory and took $\Delta = 4(10^n + 3)$ with $40 \leq n \leq 70$. For each discriminant, we used the optimal parameters given in [14], including the size of the factor base, even if we tend to reduce this parameter when optimizing the overall time. The only parameter we modified is the tolerance value for the SIQS, as a higher tolerance value is required for the large prime variations. In each case we took $B_2 = 12B_1$. It is shown in [4] that the ratio B_2/B_1 does not have an important impact on the sieving time.

Table 1. Comparative table of the relation collection time

n	0LP	1LP	2LP	2LP Batch
40	0.83	0.48	0.63	0.90
45	6.70	3.10	2.70	2.20
50	23.00	9.50	9.20	6.10
55	56.00	26.00	23.00	15.00
60	202.00	86.00	69.00	41.00
65	1195.00	513.00	354.00	227.00
70	4653.00	1906.00	1049.00	834.00

The timings in Table 1 correspond to the optimal value of the tolerance value in each case, found by trying values between 1.7 and 4, and keeping the optimum for each scenario. For 0LP, the optimal value is between 1.7 and 2.3 whereas it is around 2.3 for 1LP, 2.8 for 2LP and 3.0 for 2LP Batch. The latter case has a higher optimal tolerance value because using the batch smoothness test allows one to spend more time factoring the residues. When using Bernstein's smoothness test, we took batches of 100 residues. In our experiments, this value did not seem to have an important effect on the relation collection time. We observe in Table 1 that the use of the large prime variants has a strong impact on the relation collection phase, and that using the smoothness batch test strategy yields an additional speed-up of approximately 20% over the double large prime strategy.

Structured Gaussian elimination. Structured Gaussian elimination allows us to reduce the time taken by the linear algebra phase by reducing the dimensions of the relation matrix. Our method minimizes the growth of the density and of the size of the coefficients. To illustrate the impact of the algorithm described in §3.2, we monitor in Table 2 the evolution of the dimensions of the matrix, the average Hamming weight of its rows, the extremal values of its coefficients and the time taken for computing its HNF in the case of a relation matrix corresponding to $\Delta = 4(10^{60} + 3)$. We keep track of these values after all i-way merges for some values of i between 5 and 170. The original dimensions of the matrix are 2000×1700, and the timings are obtained on a 2.4 Ghz Opteron with 32GB of memory.

In [4], the first author regularly deleted the rows having the largest coefficients. To do this, we need to create more rows than in the base case. To provide a fair comparison between the two strategies, we used the same relation matrix resulting from a relation collection phase without large primes, and with as few rows as was required to use the same algorithm as in [14]. We therefore had to drop the regular row deletion. We also tuned the cost function to compensate for the resulting growth of the coefficients, using

$$C(r) = \sum_{1 \le |e_i| \le 8} 1 + 100 \sum_{|e_j| > 8} |e_j| \ ,$$

instead of (1).

The HNF computation consists of taking the GCD of the determinants of two different submatrices of the matrix after elimination using Linbox, and using the modular HNF of NTL with this value. Indeed, this GCD (which is likely to be relatively small) is a multiple of h_Δ. This method, combined with an elimination strategy due to Havas [13], was used in [14] and implemented in LiDIA [18]. As this implementation is no longer available, we instead refer to the timings of our code, which has the advantage of using the best linear algebra libraries available today.

Table 2. Comparative table of elimination strategies

Naive Gauss						
i	Row Nb	Col Nb	Average weight	max coeff	min coeff	HNF time
5	1189	1067	27.9	14	-17	357.9
10	921	799	49.3	22	-19	184.8
30	757	635	112.7	51	-50	106.6
50	718	596	160.1	81	-91	93.7
70	699	577	186.3	116	-104	85.6
90	684	562	205.5	137	-90	79.0
125	664	542	249.0	140	-146	73.8
160	655	533	282.4	167	-155	72.0
170	654	532	286.4	167	-155	222.4
With dedicated elimination strategy						
i	Row Nb	Col Nb	Average weight	max coeff	min coeff	HNF time
5	1200	1078	26.8	13	-12	368.0
10	928	806	42.6	20	-15	187.2
30	746	624	82.5	33	-27	100.8
50	702	580	107.6	64	-37	84.3
70	672	550	136.6	304	-676	73.4
90	656	534	157.6	1278	-1088	67.5
125	637	515	187.1	3360	-2942	63.4
160	619	497	214.6	5324	-3560	56.9
170	615	493	247.1	36761280	-22009088	192.6

Table 2 shows that the use of our elimination strategy leads to a matrix with smaller dimensions (493 rows with our method, 533 with the naive elimination) and lower density (the average weight of its rows is of 214 with our method and 282 with the naive elimination). These differences result in an improvement of the time taken by the HNF computation: 56.9 seconds with our method against 72.0 seconds with the naive Gaussian elimination. The regular cancellation of the rows having the largest coefficients over the course of the algorithm would delay the explosion of the coefficient size, but require more rows for the original matrix. This brutal increase in the size of the extremal values of the matrix can be seen in Table 2. At this point these higher values propagate during pivoting operations, and any further column elimination becomes counter-productive.

Factor base verification. The improvements in the relation collection phase have an impact on the factor base verification. The impact of the smoothness batch test is straightforward, whereas the large prime variants act in a more subtle way. Indeed, we create many more relations when using the large prime variants, and the relations created involve primes of larger norm. Therefore, a given prime not in $\mathcal{B}$ of norm less than $6\log^2 \Delta$ is more likely to appear in a relation, and thus not to need to be verified. Table 3 shows the impact of the large prime variants and of the batch smoothness test on the verification time. We used a 2.4 GHz Opteron with 16GB of memory. We considered discriminants of the form $\Delta = 4(10^n + 3)$ for n between 40 and 70, and we chose in every case the factor base giving the best results for the base scenario.

Table 3. Comparative table of the factor base verification time

n	0LP	1LP	2LP	2LP Batch
40	17.0	11.0	11.0	6.2
45	77.0	44.0	30.0	18.0
50	147.0	85.0	52.0	43.0
55	308.0	167.0	134.0	110.0
60	826.0	225.0	282.0	274.0
65	8176.0	1606.0	1760.0	1689.0
70	9639.0	4133.0	5777.0	2706.0

Regulator computation. Our method for computing the regulator avoids computing the relation matrix kernel. Instead, we need to solve a few linear systems involving the matrix resulting from the Gaussian elimination. To illustrate the impact of this algorithm, we used the relation matrix obtained in the base case for discriminants of the form $4(10^n + 3)$ for n between 40 and 70. The timings are obtained on a 2.4GHz Opteron with 16GB of memory.

In Table 4, the timings corresponding to our system solving approach are taken with seven kernel vectors. However, in most cases only two or three vectors are required to compute the regulator. As most of the time taken by our approach

Table 4. Comparative table of regulator computation time

n	Kernel Computation	System Solving
40	15.0	6.2
45	18.0	8.3
50	38.0	20.0
55	257.0	49.0
60	286.0	103.0
65	5009.0	336.0
70	10030.0	643.0

Table 5. Effect on the overall time

n	strategy	$\lvert\mathcal{B}\rvert$	relations	elimination	HNF	regulator	verification	total
40	base	400	0.8	0.1	3.2	14.6	16.8	35.6
	0LP	400	0.7	0.1	2.2	6.0	16.6	25.7
	1LP	300	0.8	0.2	2.5	6.4	13.1	23.1
	2LP	250	1.7	0.3	4.8	8.7	18.0	33.3
	2LP Batch	250	0.5	0.2	3.6	6.7	4.4	15.5
45	base	500	6.7	0.1	5.1	18.0	77.0	107.0
	0LP	500	5.9	0.2	4.9	10.0	85.0	106.0
	1LP	400	4.0	0.4	6.0	11.0	50.0	71.0
	2LP	350	3.8	0.5	12.0	17.0	36.0	69.0
	2LP Batch	350	2.6	1.1	9.0	14.0	30.0	57.0
50	base	750	23.0	0.3	16.0	38.0	147.0	224.0
	0LP	700	21.0	0.4	15.0	20.0	147.0	203.0
	1LP	450	20.0	0.4	10.0	17.0	108.0	155.0
	2LP	400	14.0	0.8	22.0	23.0	74.0	133.0
	2LP Batch	400	10.0	0.6	21.0	25.0	62.0	119.0
55	base	1200	129.0	1.9	60.0	257.0	308.0	756.0
	0LP	1300	47.0	0.7	52.0	49.0	265.0	414.0
	1LP	650	61.0	0.7	28.0	33.0	255.0	378.0
	2LP	550	40.0	1.1	48.0	48.0	177.0	313.0
	2LP Batch	550	34.0	1.0	47.0	48.0	141.0	271.0
60	base	1700	322.0	2.9	95.0	286.0	830.0	1535.0
	0LP	1700	187.0	1.3	106.0	103.0	846.0	1244.0
	1LP	750	309.0	1.0	45.0	64.0	865.0	1284.0
	2LP	700	143.0	2.1	152.0	137.0	365.0	799.0
	2LP Batch	700	142.0	1.8	103.0	100.0	309.0	655.0
65	base	2700	10757.0	12.0	652.0	5009.0	8176.0	24607.0
	0LP	2700	1225.0	2.8	489.0	336.0	3676.0	5730.0
	1LP	1900	1003.0	15.0	318.0	262.0	2984.0	4583.0
	2LP	1200	753.0	4.7	525.0	398.0	1943.0	3624.0
	2LP Batch	1000	1030.0	35.0	199.0	219.0	1642.0	3125.0
70	base	3700	17255.0	24.0	1869.0	10031.0	9639.0	38818.0
	0LP	3600	4934.0	19.0	1028.0	644.0	9967.0	16591.0
	1LP	2500	3066.0	17.0	845.0	646.0	9005.0	13579.0
	2LP	1700	2414.0	27.0	2054.0	1295.0	4590.0	10379.0
	2LP Batch	1700	2588.0	20.0	1372.0	934.0	5078.0	9991.0

is spent on system solving, we see that computing fewer kernel vectors would result in an improvement of the timings, at the risk of obtaining a multiple of the regulator.

Overall time. We have studied the individual impact of our improvements on each stage of the algorithm. We now present their effect on the overall time taken by the algorithm, including the factor base verification time, for discriminants of the form $\Delta = 4(10^n + 3)$ with $40 \leq n \leq 70$ on a 2.4 GHz Opteron with 16GB of memory. We used the same parameters as in [14], except for the tolerance

and the size of the factor base. We notice in Table 5 that the optimal size of the factor base is smaller when we use improvements for the sieving phase. For example the optimal size for the double large prime variant is half the one of the base case scenario. This results in an improvement in the HNF and regulator computation whereas the relation collection time can remain unchanged, or even increase. The tolerance value we chose varies only with the strategy, but not with the size of the discriminant. We chose 2.0 for the base case and 0LP whereas we set it to 2.3 for 1LP, 2.8 for 2LP and 3.0 for 2LP Batch. We eliminated columns of weight up to $w = 150$ since Table 2 indicates that further elimination is counter-productive.

Table 5 shows that there is an overall speed-up of of a factor of 2 for the smallest discriminants and 4 for the largest. The base case with the largest discriminants suffers from the necessity of finding some relations in a more randomized way. This ensures that we can get full rank submatrices of the relation matrix after the Gaussian elimination to compute a small multiple of h_Δ. Matrices produced using the large prime variants do not need this extra step, even with the largest discriminants. This naturally affects the sieving time, since we cannot use SIQS for that purpose, but also affects phases relying on linear algebra. Indeed, elimination produces a matrix with larger entries and dimensions.

4.2 Large Example

The improvements we described allow us to compute class groups and regulators of real number fields with larger discriminants than was previously possible. The key is to parallelize the relation collection and verification phase, while the linear algebra has to be performed the usual way. These methods were successfully used in [4] to compute the class group structure of an imaginary quadratic field with a 110-digit discriminant. We used a cluster with 260 2.4GHz Xeon cores to compute a relation matrix corresponding to the discriminant $\Delta_{110} := 4(10^{110}+3)$ in 4 days. We allowed two large primes, used a tolerance value of 3.0, tested batches of 100 residues, took $w = 250$ and set $|\mathcal{B}| = 13000$. Then, we used three 2.4 GHz Opterons with 32GB of memory each to compute determinants of full-rank submatrices of the relation matrix after the Gaussian elimination in 1 day, and one 2.4GHz Opteron to compute the HNF modulo the GCD of these determinants in 3 days. We had to find 4018 extra relations during the verification phase that took 4 days on 96 2.4GHz Xeon cores. We thus obtained that

$$Cl_{\Delta_{110}} \cong \mathbb{Z}/12\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z} \ , \tag{2}$$

and the corresponding regulator is

$$R_{\Delta_{110}} \approx 70795074091059722608293227655184666748799878533480399.6730200233 \ .$$

We estimate that it would take two weeks (4000 relations per day) to complete the relation collection for Δ_{120} with the same factor base as Δ_{110}, thus requiring a similar time for the linear algebra.

5 Conclusions

Recently, our work has been extended to the problems of principal ideal testing and solving the discrete logarithm problem in the ideal class group [5]. The double large prime variant and improvements to relation generation translated directly to improvements in this context. However, HNF computations are not required for this problem, and linear system solving over $\mathbb{Z}$ can be used instead. The numerical results were used to give estimates for discriminant sizes that offer equivalent security to recommended sizes of RSA moduli.

Some possibilities for further improvements remain to be investigated. For example, a lattice sieving strategy could be used to sieve $\varphi(x, y)$ instead of $\varphi(x, 1)$. Factor refinement and coprime factorization techniques may be a useful alternative to Bernstein's batch smoothness test. Multiple large primes have been successfully used for integer factorization and could also be tried in our context.

There is also still room for improvement to the linear algebra components. For example, a HNF algorithm that exploits the natural sparseness of the relation matrix, perhaps as a black-box algorithm, would be useful. If such an algorithm were available, we could reconsider using Gaussian elimination techniques since they induce a densification of the matrix. We could also study the effect of other dense HNF algorithms in existing linear algebra packages such as KASH, Pari, Sage and especially MAGMA which seems to have the most efficient HNF algorithm for our types of matrices. In that case, we would need the elimination phase regardless of how these algorithms are affected by the density and the size of the coefficients of the matrix. Indeed, we cannot afford manipulating a dense representation of the matrix before the Gaussian elimination phase.

References

1. Bach, E.: Explicit bounds for primality testing and related problems. Math. Comp. 55(191), 355–380 (1990)
2. Bach, E.: Improved approximations for Euler products. In: Number Theory: CMS Proc., vol. 15, pp. 13–28. Amer. Math. Soc., Providence (1995)
3. Bernstein, D.: How to find smooth parts of integers. Mathematics of Computation (submited)
4. Biasse, J.-F.: Improvements in the computation of ideal class groups of imaginary quadratic number fields. In: Advances in Mathematics of Communications (to appear 2010)
5. Biasse, J.-F., Jacobson Jr., M.J., Silvester, A.K.: Security estimates for quadratic field based cryptosystems. In: ACISP (to appear 2010)
6. Buchmann, J.: A subexponential algorithm for the determination of class groups and regulators of algebraic number fields. In: Séminaire de Théorie des Nombres (Paris), pp. 27–41 (1988-1989)
7. Buchmann, J., Düllmann, S.: Distributed class group computation. In: Festschrift aus Anlaß des sechzigsten Geburtstages von Herrn Prof. Dr. G. Hotz, pp. 69–79. Universität des Saarlandes (1991), Teubner, Stuttgart (1992)

8. Cavallar, S.: Strategies in filtering in the number field sieve. In: Bosma, W. (ed.) ANTS 2000. LNCS, vol. 1838, pp. 209–232. Springer, Heidelberg (2000)
9. Chen, Z., Storjohann, A., Fletcher, C.: IML: Integer Matrix Library. Software (2010), http://www.cs.uwaterloo.ca/~astorjoh/iml.html
10. Dodson, B., Leyland, P.C., Lenstra, A.K., Muffett, A., Wagstaff, S.: MPQS with three large primes. In: Fieker, C., Kohel, D.R. (eds.) ANTS 2002. LNCS, vol. 2369, pp. 446–460. Springer, Heidelberg (2002)
11. GMP, The GNU multiple precision bignum library. Software (2010), http://gmp-lib.org/
12. Gower, J.E., Wagstaff, S.: Square form factorization. Mathematics of Computation 77, 551–588 (2008)
13. Havas, G., Majewski, B.S.: Integer matrix diagonalization. Journal of Symbolic Computing 24, 399–408 (1997)
14. Jacobson Jr., M.J.: Subexponential class group computation in quadratic orders, Ph.D. thesis, Technische Universitt Darmstadt, Darmstadt, Germany (1999)
15. Jacobson Jr., M.J., Scheidler, R., Williams, H.C.: The efficiency and security of a real quadratic field based key exchange protocol. In: Public-Key Cryptography and Computational Number Theory, Warsaw, Poland, pp. 89–112. de Gruyter (2001)
16. Jacobson Jr., M.J., Williams, H.C.: Solving the Pell equation. CMS Books in Mathematics. Springer, Heidelberg (2009) ISBN 978-0-387-84922-5
17. Lenstra, A.K., Manasse, M.S.: Factoring with two large primes (extended abstract). In: Damgård, I.B. (ed.) EUROCRYPT 1990. LNCS, vol. 473, pp. 72–82. Springer, Heidelberg (1991)
18. LiDIA Group, LiDIA: a c++ library for computational number theory. Software, Technische Universität Darmstadt, Germany (1997), http://www.informatik.tu-darmstadt.de/TI/LiDIA
19. LinBox, Project LinBox: Exact computational linear algebra. Software (2010), http://www.linalg.org/
20. Louboutin, S.: Computation of class numbers of quadratic number fields. Math. Comp. 71(240), 1735–1743 (2002)
21. Maurer, M.: Regulator approximation and fundamental unit computation for real quadratic orders, Ph.D. thesis, Technische Universitt Darmstadt, Darmstadt, Germany (1999)
22. Milan, J.: Tifa. Software (2010), http://www.lix.polytechnique.fr/Labo/Jerome-Milan/tifa/tifa.xhtml
23. Shoup, V.: NTL: A Library for doing Number Theory. Software (2010), http://www-shoup.net/ntl
24. Vollmer, U.: An accelerated Buchmann algorithm for regulator computation in real quadratic fields. In: Fieker, C., Kohel, D.R. (eds.) ANTS 2002. LNCS, vol. 2369, pp. 148–162. Springer, Heidelberg (2002)

On the Use of the Negation Map in the Pollard Rho Method

Joppe W. Bos, Thorsten Kleinjung, and Arjen K. Lenstra

Laboratory for Cryptologic Algorithms
EPFL, Station 14, CH-1015 Lausanne, Switzerland

Abstract. The negation map can be used to speed up the Pollard rho method to compute discrete logarithms in groups of elliptic curves over finite fields. It is well known that the random walks used by Pollard rho when combined with the negation map get trapped in fruitless cycles. We show that previously published approaches to deal with this problem are plagued by recurring cycles, and we propose effective alternative countermeasures. As a result, fruitless cycles can be resolved, but the best speedup we managed to achieve is by a factor of only 1.29. Although this is less than the speedup factor of $\sqrt{2}$ generally reported in the literature, it is supported by practical evidence.

Keywords: Pollard's rho method, fruitless cycles, negation map.

1 Introduction

The difficulty of the elliptic curve discrete logarithm problem (ECDLP) underlies the security of cryptographic schemes based on elliptic curves over finite fields [11,13]. The best method known to solve ECDLP for curves without special properties is the parallelized [17] Pollard rho method [15]. A common optimization is to halve the search space by identifying a point with its inverse [18,9,7]. Because representatives for the equivalence classes can quickly be computed using the *negation map*, this equivalence relation may result in a speedup by a factor of up to $\sqrt{2}$ when solving ECDLP. For the elliptic curves over binary extension fields $\mathbf{F}_{2^t}$ from [12], order t equivalence relations can be used as well, resulting in a speedup by a factor of up to $\sqrt{2t}$ [18,9].

Usage of the negation map in the context of the Pollard rho method leads to *fruitless cycles*, useless cycles trapping the random walks. An analysis of their likelihood of occurrence appeared in [7]. Various methods have been proposed [18,9] to deal with them, all leading to costlier random walks and administrative overhead. The literature suggests that the resulting inefficiencies are negligible, and that a speedup by a factor of $\sqrt{2}$ is attainable [1, Section 19.5.5].

We analyze fruitless cycles and the previously published methods to avoid their ill effects and show that current approaches to escape from cycles suffer from *recurring cycles*. These may have contributed to the lack of practical usage of the negation map to solve prime field ECDLPs: it was not used for the solutions

G. Hanrot, F. Morain, and E. Thomé (Eds.): ANTS-IX 2010, LNCS 6197, pp. 66–82, 2010.

[10,6] of the 79-, 89-, 97- and 109-bit prime field Certicom challenges [5]. Neither was it used by the independent current 112-bit prime field record [3].

We present and analyze alternative methods to deal with fruitless cycles. All our analyses are supported by experiments. We found that the negation map indeed leads to a speedup, but we have not been able to reach more than a factor of 1.29, somewhat short of the $\sqrt{2}$ that we had hoped for. We also found that the best attainable speedup depends on the platform one uses: for instance, if the Pollard rho method is parallelized in SIMD fashion, it is a challenge to achieve any speedup at all. This has consequences for the applicability of the negation map in large scale prime field ECDLP solution attempts. For such efforts, all participating processors must use the same random walk definition, so one may desire to gear the implementation towards processors with the best performance/price ratio, such as graphics cards (which are SIMT, a SIMD variant).

The negation map (while dealing with cycles) slows down random walks in three ways. In the first place, on average more elliptic curve group operations are required per step of each walk. This is unavoidable and attempts should be made to minimize the number of additional operations. Secondly, dealing with cycles entails administrative overhead and branching, which cause a non-negligible slowdown when running multiple walks in SIMD-parallel fashion. Finally, the best way to counter the effect of the higher average number of group operations per step is making the walks "more random" by allowing a finer grained decision per step. However, the beneficial effects of this approach are, in most circumstances on current processors, wiped out by cache inefficiencies. It will be seen that it is best to strike a balance between the first and third of these slowdowns. The second slowdown somewhat affects regular PCs, but is a major obstacle to the negation map in SIMD environments.

This paper is organized as follows. Section 2 recalls background on ECDLP, the Pollard rho method and fruitless cycles. Section 3 introduces recurring cycles and presents and analyzes new methods to deal with them. Section 4 compares the various cycle reduction, detection, and escape methods in practice.

2 Preliminaries

2.1 The Elliptic Curve Discrete Logarithm Problem

Let $\mathbf{F}_p$ denote a finite field of odd prime characteristic p. Any $a, b \in \mathbf{F}_p$ with $4a^3+27b^2 \neq 0$ define an elliptic curve $E_{a,b}$ over $\mathbf{F}_p$. The additively written *group of points* $E_{a,b}(\mathbf{F}_p)$ *of* $E_{a,b}$ *over* $\mathbf{F}_p$ is defined as the *zero point* $\mathfrak{o}$ along with the set of pairs $(x, y) \in \mathbf{F}_p \times \mathbf{F}_p$ that satisfy the *shortened Weierstrass equation* $y^2 = x^3 + ax + b$. Let p, a, b and $\mathfrak{g} \in E_{a,b}(\mathbf{F}_p)$ of prime order q be such that the index $[E_{a,b}(\mathbf{F}_p) : \langle \mathfrak{g} \rangle]$ is small. For $\mathfrak{h} \in \langle \mathfrak{g} \rangle$, the ECDLP is to find an integer m such that $m\mathfrak{g} = \mathfrak{h}$. For curves without special properties, solving ECDLP is believed to require an effort on the order of $\sqrt{q}$. Pollard's rho method achieves this run time, while requiring more or less constant memory.

2.2 Pollard's Rho Method

If objects are selected truly at random and with replacement from q objects, the conditional probability at step $n+1$ of finding the first duplicate (or *collision*) is $\frac{n}{q}$ (if $n < q$). Via straightforward arguments this leads to $\sqrt{\pi q/2}$ for the expected number of steps until the first collision. If random objects are selected as $u\mathfrak{g} + v\mathfrak{h} \in \langle \mathfrak{g} \rangle$ for random integer multipliers u, v, a collision corresponds to $u, v, \bar{u}, \bar{v}$ such that $u\mathfrak{g} + v\mathfrak{h} = \bar{u}\mathfrak{g} + \bar{v}\mathfrak{h}$. Unless $\bar{v} \equiv v \bmod q$, the value $m = \frac{u-\bar{u}}{\bar{v}-v} \bmod q$ solves the discrete logarithm problem. The expected number of steps of this idealized version of Pollard's rho method [15] is $\sqrt{\pi q/2}$.

***r*-adding and *r*+*s*-mixed walks.** Pollard's rho method uses an approximation of a truly random walk in $\langle \mathfrak{g} \rangle$. Let, for a small integer r, an index function $\ell : \langle \mathfrak{g} \rangle \mapsto [0, r-1]$ induce an r-partition $\langle \mathfrak{g} \rangle = \cup_{i=0}^{r-1} \mathfrak{G}_i$ of $\langle \mathfrak{g} \rangle$, where $\mathfrak{G}_i = \{\mathfrak{x} : \mathfrak{x} \in \langle \mathfrak{g} \rangle, \ell(\mathfrak{x}) = i\}$ and all $\mathfrak{G}_i$ have cardinality close to $\frac{q}{r}$. For random integers u_i, v_i, elements $\mathfrak{f}_i = u_i\mathfrak{g} + v_i\mathfrak{h} \in \langle \mathfrak{g} \rangle$ are precomputed for $0 \le i < r$. Starting at a random but known multiple of $\mathfrak{g}$, the successor of a point $\mathfrak{p}$ of the walk is defined as $\mathfrak{p} + \mathfrak{f}_{\ell(\mathfrak{p})} \in \langle \mathfrak{g} \rangle$. It is easy to keep track of the u, v such that $\mathfrak{p} = u\mathfrak{g} + v\mathfrak{h}$.

Such an *r-adding walk* results in an expected number of steps until a collision occurs that is somewhat larger than $\sqrt{\pi q/2}$, as shown by Brent and Pollard [4] and expanded upon in [2]. Assume that ℓ is perfectly random. Let $p_i = \frac{\#\mathfrak{G}_i}{q}$. A point in the walk is said to belong to class i if its predecessor upon its first occurrence belongs to $\mathfrak{G}_i$. If the nth point belongs to $\mathfrak{G}_j$ (with probability p_j) and the $(n+1)$st point produces the first collision, the collision point cannot be of class j (this happens with probability p_j), since then the collision would have occurred in step n. Therefore, the probability that the first collision occurs at step $n+1$ is

$$\frac{n}{q}(1 - \sum_{j=0}^{r-1} p_j^2).$$

With $q' = \frac{q}{1-\sum_{j=0}^{r-1} p_j^2}$ this is $\frac{n}{q'}$. We get via the same arguments referred to above

$$\sqrt{\frac{\pi q'}{2}} = \sqrt{\frac{\pi q}{2(1 - \sum_{j=0}^{r-1} p_j^2)}} \tag{1}$$

for the expected number of steps until the first collision.

Pollard [15] uses $r = 3$, $\mathfrak{f}_0 = \mathfrak{h}$, and $\mathfrak{f}_2 = \mathfrak{g}$, but replaces the $i = 1$ case by the doubling $2\mathfrak{p}$. Teske [16] shows that a larger r, such as $r = 20$, leads to better performance on average, conform the analysis, even if none of the choices does an explicit doubling, as Pollard's $i = 1$ case.

Inclusion of doublings leads to $r + s$*-mixed walks*: with $\ell : \langle \mathfrak{g} \rangle \mapsto [0, r+s-1]$ partitioning $\langle \mathfrak{g} \rangle$ into $r+s$ parts of cardinality close to $\frac{q}{r+s}$, the next point equals $\mathfrak{p} + \mathfrak{f}_{\ell(\mathfrak{p})}$ if $0 \le \ell(\mathfrak{p}) < r$, but $2\mathfrak{p}$ if $\ell(\mathfrak{p}) \ge r$. Pollard's walk is a $2+1$-mixed walk. The analysis above applies again, assuming that we consider the doublings as one class, hit with probability p_D. Experiments by Teske show that best performance is achieved for $\frac{s}{r}$ between $\frac{1}{4}$ and $\frac{1}{2}$ but that apart from the case $r = 3$ mixed

walks are not significantly better. The analysis and our own experiments, as reported below, suggest that the optimal ratio $\frac{s}{r}$ is close to zero.

Per step the occurrence probability of the event $\mathfrak{p} = \mathfrak{f}_i$ (and thus a chance to solve the discrete logarithm problem) is negligible compared to the probability of a birthday collision. So, for r-adding walks doublings most likely will not occur.

Parallelized random walks. Parallelization of Pollard's rho method does not consist of running any number of random walks in parallel, until one of them collides: on M processors the expected speedup would be by a factor of $\sqrt{M}$, so overall it would require $\sqrt{M}$ more processing power than a single processor. The proper way to parallelize Pollard's rho method is presented in [17]. It achieves an M-fold speedup on M processors, thus requiring the same overall processing power as a single process, but in $\frac{1}{M}$th of the time. Different processes must be able to efficiently recognize if, probably at different points in time, their walks collide. To achieve this, each process generates a single random walk, each from its own random starting point, but all using the same index function ℓ and the same $\mathfrak{f}_i$'s. As soon as a walk hits upon a *distinguished point*, this point is reported. The idea is that when two walks collide – without noticing it – they will keep taking the same steps (because they use the same walk definition) and will thus both ultimately reach the same distinguished point. This will be noticed when the colliding distinguished point is reported. The discrete logarithm can then be computed from the two, hopefully distinct, pairs of integer multipliers u, v that correspond to the same distinguished point.

A distinguished point must be easy to recognize, occur with low enough probability to make it possible to store them all and to efficiently find collisions, but occur often enough for every walk to hit one. The distinguishing property could be that k specific bits of the point's x-coordinate are zero, in which case walks may hit a distinguished point once every 2^k steps.

The parallelized version of Pollard's rho method requires a unique, and thus affine, point representation to make the walks well-defined and to recognize distinguished points. The fastest suitable type of elliptic curve group arithmetic uses the affine Weierstrass point representation. Per group operation, it requires a (usually expensive) modular inversion. Its cost is amortized among the walks running in parallel per processor, at the cost of three modular multiplications per step per walk, using Montgomery's simultaneous inversion [14]. Point doubling requires an extra modular squaring compared to regular non-doubling point addition. This makes doubling on average about $\frac{7}{6}$ times slower than regular addition when parallelized walks and simultaneous inversion are used.

Using automorphisms. Following [18], define an equivalence relation $\sim$ on $\langle \mathfrak{g} \rangle$ by $\mathfrak{p} \sim -\mathfrak{p}$ for $\mathfrak{p} \in \langle \mathfrak{g} \rangle$ and, instead of searching $\langle \mathfrak{g} \rangle$ of size q, search $\langle \mathfrak{g} \rangle / \sim$ of size about $\frac{q}{2}$. Denoting the equivalence class containing $\mathfrak{p}$ and $-\mathfrak{p}$ by $\sim\mathfrak{p}$, it may be represented by the element with y-coordinate of least absolute value. It is trivial to calculate since $-(x, y) = (x, -y)$ for $(x, y) \in \langle \mathfrak{g} \rangle$. Thus, using this *negation map* one would expect to save a factor of $\sqrt{2}$ in the number of steps.

For r-adding and $r + s$-mixed walks the speedup by a factor of $\sqrt{2}$ is slightly too pessimistic. Let the definitions of p_i, p_D, and of class i be as above. Assume

Table 1. Number of steps required by the Pollard rho method in random elliptic curve groups of 31-bit prime order q over prime fields of random 31-bit prime characteristic p, divided by $\sqrt{\pi q/2}$ or by $\sqrt{\pi q/4}$ (without or with the negation map). Lowest and highest averages are over 10 measurements. Each measurement calculates the average number of steps taken until a collision occurs, over 100 000 collision searches where for each search a prime p and an elliptic curve over $\mathbf{F}_p$ are randomly selected until the order q of the group of points is prime. Overall average is the average of the 10 averages (thus, the average over one million searches). Expression (1) and (2) columns are the quotients as expected based on expressions (1) (with $p_i = \frac{1}{r}$ for $0 \le i < r$) and (2) (with $p_i = \frac{1}{r+s}$ for $0 \le i < r$ and $p_D = \frac{s}{r+s}$), respectively. Those expressions are for $q \to \infty$ and indeed for larger (smaller) q they give a better (worse) fit.

	Without negation map				With negation map			
	Averages			Expression	Averages			Expression
	lowest	overall	highest	(1)	lowest	overall	highest	(2)
8-adding	1.079	1.083	1.085	1.069	1.035	1.039	1.042	1.033
16-adding	1.032	1.037	1.040	1.033	1.015	1.017	1.020	1.016
32-adding	1.014	1.018	1.019	1.016	1.007	1.009	1.011	1.008
16 + 4-mixed	1.041	1.043	1.044	1.043	1.036	1.038	1.040	1.031
16 + 8-mixed	1.075	1.078	1.081	1.078	1.075	1.077	1.079	1.069

that the nth point belongs to $\mathfrak{G}_j$ and that the $(n+1)$st point produces the first collision while hitting the representative $\mathfrak{p}$, directly or after negation. If this step is a doubling then the analysis is as above. This happens with probability p_D^2. Otherwise, we only exclude the case that, as a result of just the addition, the two predecessors hit the same point ($\mathfrak{p}$ or $-\mathfrak{p}$). This happens with probability $\frac{p_j^2}{2}$. Therefore, the probability that the first collision occurs at step $n+1$ is

$$\frac{2n}{q}(1 - p_D^2 - \sum_{j=0}^{r-1} \frac{p_j^2}{2}).$$

As above we get

$$\sqrt{\frac{\pi q}{4(1 - p_D^2 - \frac{1}{2}\sum_{j=0}^{r-1} p_j^2)}} \tag{2}$$

for the expected number of steps until the first collision. For the same parameter values this expression is more than $\sqrt{2}$ smaller than Expression (1). However, usage of the negation map requires modifications to the iteration function due to the occurrence of *fruitless cycles*. This disadvantage of the negation map was already pointed out in [9,18]. It is the focus of this article.

The group $\langle \mathfrak{g} \rangle$ may admit other trivially computable maps. For Koblitz curves the Frobenius automorphism of a degree t binary extension field leads to a further $\sqrt{t}$-fold speedup. This does not apply to the case considered here.

Small scale experiments. We checked the accuracy of predictions based on expressions (1) and (2). The results, for 31-bit primes q, are listed in Table 1.

With all averages larger than 1, both r-adding and $r+s$-mixed walks on average perform worse than truly random walks. For most walks with the negation map the averages are lower than their negation-less counterparts, indicating that the reduction factor in the expected number of steps is indeed larger than $\sqrt{2}$. This does not imply a speedup by the same factor, because to obtain the figures costly fruitless cycle detection methods had to be used. It can be seen that $r+s$-mixed walks are disadvantageous if $s > \frac{r}{4}$.

2.3 Fruitless Cycles

Straightforward application of the negation map to Pollard's rho method with r-adding or $r+s$-mixed walks does not work due to fruitless cycles. This section describes the current state-of-the-art of dealing with those cycles.

Length 2 cycles. If a random walk step goes from $\mathfrak{p}$ to $-\mathfrak{p} - \mathfrak{f}_i$ (with probability $\frac{1}{2}$, for some i) and $-\mathfrak{p} - \mathfrak{f}_i \in \mathfrak{G}_i$ (with probability $\frac{1}{r}$), then the next point after $-\mathfrak{p} - \mathfrak{f}_i$ is $\mathfrak{p}$ again (with probability 1), thereby cancelling the effect of the previous step. It follows that a fruitless 2-cycle starts from a random point with probability $\frac{1}{2r}$, cf. [7, Proposition 31]. This 2-cycle is denoted as

$$\mathfrak{p} \xrightarrow{(i,-)} -(\mathfrak{p} + \mathfrak{f}_i) \xrightarrow{(i,-)} \mathfrak{p}.$$

Here "(i,s)" with $s \in \{-,+\}$ indicates that addition constant $\mathfrak{f}_i$ is added to a point $\mathfrak{p}$ after which the result is left as is ($s = +$) or negated ($s = -$) to find the correct representative ($\mathfrak{p} + \mathfrak{f}_i$ if $s = +$, or $-\mathfrak{p} - \mathfrak{f}_i$ if $s = -$). Any walk with two consecutive steps "$(i,-)$" is trapped in an infinite loop. Because this happens with probability $\frac{1}{2r}$, all walks can be expected to end up in fruitless cycles after a moderate number of steps when the negation map is used with r-adding walks.

Looking ahead to reduce 2-cycles. To reduce the occurrence of 2-cycles, Wiener and Zuccherato propose to use a more costly iteration function that results in a lower probability that two successive points belong to the same partition [18]. This can be achieved by using the first i of $\ell(\mathfrak{p})$, $\ell(\mathfrak{p})+1$, …, $\ell(\mathfrak{p})+r-1$ such that $i \bmod r \neq \ell(\sim(\mathfrak{p}+\mathfrak{f}_i))$, if such an index exists (here and in the sequel indices i in $\mathfrak{f}_i$ are understood to be taken modulo r). Thus, define the next point as $f(\mathfrak{p})$ with $f : \langle\mathfrak{g}\rangle \to \langle\mathfrak{g}\rangle$ defined by

$$f(\mathfrak{p}) = \begin{cases} E(\mathfrak{p}) & \text{if } j = \ell(\sim(\mathfrak{p}+\mathfrak{f}_j)) \text{ for } 0 \le j < r \\ \sim(\mathfrak{p}+\mathfrak{f}_i) & \text{with } i \ge \ell(\mathfrak{p}) \text{ minimal s.t. } \ell(\sim(\mathfrak{p}+\mathfrak{f}_i)) \neq i \bmod r. \end{cases}$$

The function $E : \langle\mathfrak{g}\rangle \to \langle\mathfrak{g}\rangle$ may restart the walk at a new random initial point. The latter is expected to happen once every r^r steps and will therefore not affect the efficiency. The expected cost per step of the walk is increased by a factor of $\sum_{i=0}^{r} \frac{1}{r^i}$, which lies between $1 + \frac{1}{r}$ and $1 + \frac{1}{r-1}$.

Dealing with fruitless cycles in general. Although the look-ahead technique reduces the frequency of 2-cycles, they may still occur [18]. This is elaborated upon in Section 3. Even so, it is well known that just addressing 2-cycles does

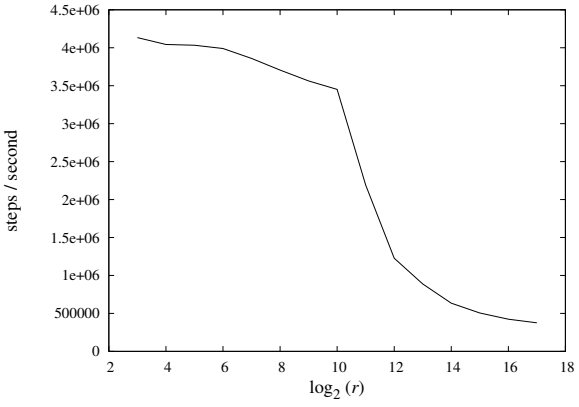


Fig. 1. Total number of steps per second as a function of r, taken by 200 parallel r-adding walks sharing the modular inversion and not using the negation map, for Pollard's rho method applied to a 131-bit prime ECDLP

not solve the problem of fruitless cycles, because longer cycles will occur as well. Reducing their occurrence requires additional overhead on top of what is already incurred to reduce 2-cycles. Given that fruitless cycles are unavoidable, they must be effectively dealt with when they occur.

In [9] a general approach is proposed to detect cycles and to escape from them: after α steps record a length β sequence of successive points and compare the next point to these β points. If a cycle is detected a cycle representative $\mathfrak{p}$ is chosen deterministically from which the cycle is escaped. One may add $\mathfrak{f}_{\ell(\mathfrak{p})+c}$ for a fixed $c \in [2, r-1]$ (the choice $c = 1$ is bad as it could lead to an immediate cycle recurrence). Instead one may add a distinct precomputed value $\mathfrak{f}'$ that does not depend on the escape-point, or one may add $\mathfrak{f}''_{\ell(\mathfrak{p})}$ from a distinct list of r precomputed values $\mathfrak{f}''_0, \mathfrak{f}''_1, \ldots, \mathfrak{f}''_{r-1}$.

In the next section we discuss fruitless cycles in greater detail and propose alternative methods that avoid problems that the method from [9] may run into.

3 Improved Fruitless Cycle Handling

The probability to enter a fruitless cycle decreases with increasing r [7]. This does not imply that it suffices to take r large enough to make the probability sufficiently low. Fig. 1 depicts the effect of increasing r-values on the performance of an r-adding walk, measured as number of steps per second. The performance deterioration can be attributed to the increasing rate of cache misses during retrieval of the addition constants $\mathfrak{f}_i$. The effect varies between processors, implementations, and elliptic curves. It is worsened for more contrived walks, such as those using the negation map where cycle reduction, detection and escape methods are unavoidable. Unless the expected overall number of steps (of order $\sqrt{q}$) is too small to be of interest, r cannot be chosen large enough to both

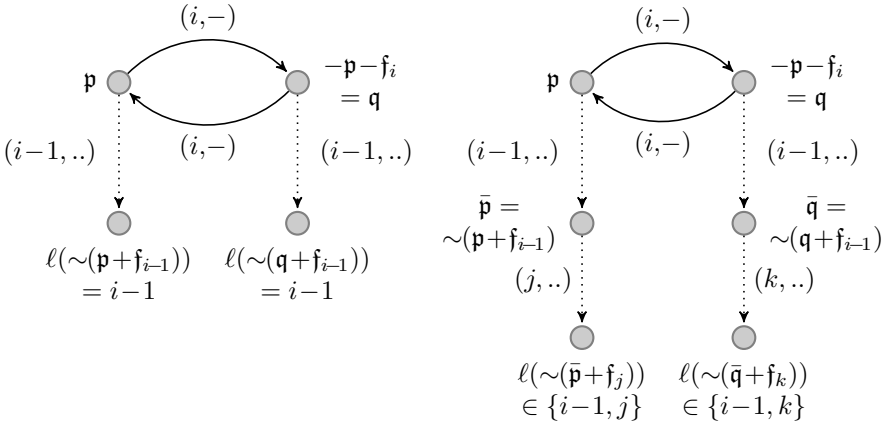


Fig. 2. 2-cycles caused by 2-cycle reduction (left) and 4-cycle reduction. The dotted steps are prevented.

avoid fruitless cycles and achieve adequate performance. Therefore, in this section we concentrate on other ways to deal with fruitless cycles. We first discuss short-cycle reduction techniques, next discuss cycle detection methods and analyze their behavior, and finally propose alternative methods.

3.1 Short Fruitless Cycle Reduction

2-cycles. Unfortunately, the look-ahead technique to reduce 2-cycles presented above introduces new 2-cycles. The dotted lines in the left example in Fig. 2 are the steps taken by the regular iteration function, the new cycle is depicted by the solid lines which are the steps taken as a result of $f(\mathfrak{p})$ and $f(\mathfrak{q})$. This new cycle occurs with probability $\frac{1}{2r^3}$. It is the most likely 2-cycle introduced by the look-ahead technique.

Lemma 1. *The probability to enter a fruitless 2-cycle when looking ahead to reduce 2-cycles while using an r-adding walk is*

$$\frac{1}{2r}\left(\sum_{i=1}^{r-1}\frac{1}{r^i}\right)^2 = \frac{(r^{r-1}-1)^2}{2r^{2r-1}(r-1)^2} = \frac{1}{2r^3} + O\left(\frac{1}{r^4}\right).$$

Proof. With i as in the definition of f, the probability is r^{-c} that $i \geq \ell(\mathfrak{p}) + c$ for $0 \leq c < r$ (considering the case $E(\mathfrak{p})$ as $i = \infty$), hence $i = \ell(\mathfrak{p}) + c$ with probability $\frac{r-1}{r}\frac{1}{r^c}$.

We compute the probability of entering a cycle consisting of points $\mathfrak{p}$ and $\mathfrak{q}$ starting at $\mathfrak{p}$. Let $j = \ell(\mathfrak{p})$ and $k = \ell(\mathfrak{q})$, and let the steps from $\mathfrak{p}$ to $\mathfrak{q}$ and back be adding $\mathfrak{f}_{j+c}$ and $\mathfrak{f}_{k+d}$, respectively. This implies that $j + c \equiv k + d \bmod r$ and that the step from $\mathfrak{p}$ to $\mathfrak{q}$ involves a negation. From the definition of f it follows

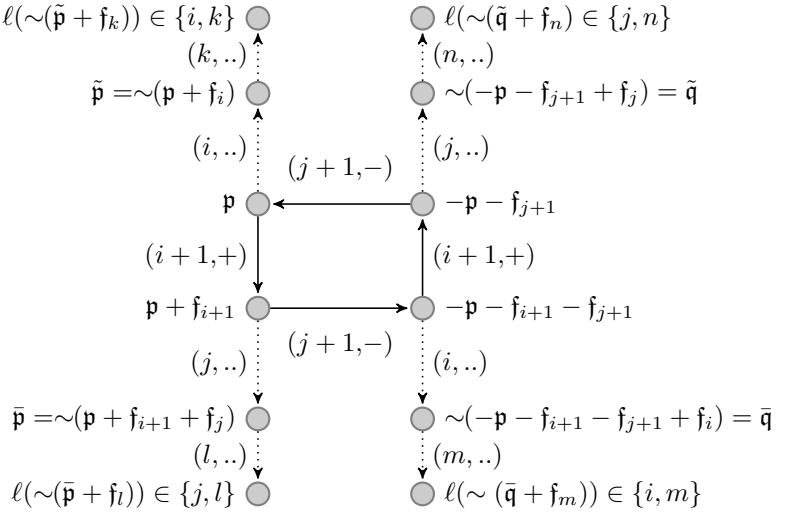


Fig. 3. A 4-cycle when the 4-cycle reduction method is used

that $\ell(\mathfrak{q}) \not\equiv j + c \bmod r$, thus $d \neq 0$ and by symmetry $c \neq 0$. Since j is given and k is determined by j, c and d, the probabilities must be summed over all possible c and d. The probability for a c, d pair is the product of the following probabilities:

- $\frac{r-1}{r}\frac{1}{r^c}$ for the first step being c;
- $\frac{1}{2}$ for the sign;
- $\frac{1}{r-1}$ for $\ell(\sim(\mathfrak{p} + \mathfrak{f}_{j+c})) = k$ (we know already that $\ell(\sim(\mathfrak{p} + \mathfrak{f}_{j+c})) \not\equiv j + c \not\equiv k \bmod r$);
- $\frac{1}{r^d}$ for the second step being d (since $\ell(\sim(\mathfrak{q} + \mathfrak{f}_{k+d})) \not\equiv k + d \bmod r$).

This results in the probability $\dfrac{1}{2r}\displaystyle\sum_{c=1}^{r-1}\sum_{d=1}^{r-1}\frac{1}{r^c}\frac{1}{r^d}$. □

We conclude that, even when the look-ahead technique is used, 2-cycles are still too likely to occur for relevant values of q and r. Some of the new 2-cycles are prevented by other short-cycle reduction methods, but the remaining ones must be dealt with using detection and escape methods. This is discussed below.

4-cycles. Unless the addition constants $\mathfrak{f}_i$ have been chosen poorly, 3-cycles do not occur as a direct result of the negation map, so that 4-cycles are the next type of short cycles to be considered. Excluding again that the $\mathfrak{f}_i$ have unlikely properties, a fruitless 4-cycle without proper sub-cycle is of the form

$$\mathfrak{p} \xrightarrow{(i,+)} \mathfrak{p} + \mathfrak{f}_i \xrightarrow{(j,-)} -\mathfrak{p} - \mathfrak{f}_i - \mathfrak{f}_j \xrightarrow{(i,+)} -\mathfrak{p} - \mathfrak{f}_j \xrightarrow{(j,-)} \mathfrak{p}.$$

The cycle may be entered at any of its four points. Hence, a fruitless 4-cycle starts from a random point with probability $\frac{r-1}{4r^3}$. This is a lower bound for the probability of occurrence of 4-cycles when looking ahead to reduce 2-cycles.

An extension of the 2-cycle reduction method looks ahead to the first two successors of a point, thereby reducing the frequency of 2-cycles and 4-cycles, while still being deterministic:

$$g(\mathfrak{p}) = \begin{cases} E(\mathfrak{p}) & \text{if } j \in \{\ell(\mathfrak{q}), \ell(\sim(\mathfrak{q} + \mathfrak{f}_{\ell(\mathfrak{q})}))\} \text{ or } \ell(\mathfrak{q}) = \ell(\sim(\mathfrak{q} + \mathfrak{f}_{\ell(\mathfrak{q})})) \\ & \text{where } \mathfrak{q} = \sim(\mathfrak{p} + \mathfrak{f}_j), \text{ for } 0 \le j < r, \\ \mathfrak{q} = \sim(\mathfrak{p} + \mathfrak{f}_i) & \text{with } i \ge \ell(\mathfrak{p}) \text{ minimal s.t.} \\ & i \bmod r \ne \ell(\mathfrak{q}) \ne \ell(\sim(\mathfrak{q} + \mathfrak{f}_{\ell(\mathfrak{q})})) \ne i \bmod r. \end{cases}$$

Compared to $f(\mathfrak{p})$, the probability that E is called increases from $(\frac{1}{r})^r$ to at least $(\frac{2}{r})^r$ because $\ell(\sim(\mathfrak{q}+\mathfrak{f}_{\ell(\mathfrak{q})})) \in \{j \bmod r, \ell(\mathfrak{q})\}$ with probability $\frac{2}{r}$ for each j. This iteration function is at least $\frac{r+4}{r}$ times slower than the standard one, because with probability $\frac{2}{r}$ at least two additional group operations need to be carried out, an effect that is slightly alleviated by a factor of $(\frac{r-1}{r})^{\frac{1}{2}}$ since the image of g is a subset of $\langle \mathfrak{g} \rangle$ of cardinality approximately $\frac{r-1}{r}q$. The value $\sim(\mathfrak{q}+\mathfrak{f}_{\ell(\mathfrak{q})})$ can be stored for use in the next iteration. Usage of g reduces the occurrence of 4-cycles, and also prevents some of the 2-cycles newly introduced by the 2-cycle reduction method (such as the one depicted on the left in Fig. 2). But g introduces new types of 2-cycles and 4-cycles as well, both of which do indeed occur in practice. A newly introduced 2-cycle is shown in the right example in Fig. 2. There the points $\bar{\mathfrak{p}}$ and $\bar{\mathfrak{q}}$ are $\notin \mathfrak{G}_{i-1} \cup \mathfrak{G}_i$. This 2-cycle occurs with probability $\frac{2(r-2)^2}{(r-1)r^4}$, which is therefore a lower bound for the probability of 2-cycles when using the 4-cycle reduction method. Fig. 3 depicts an example of a newly introduced 4-cycle: the points reached via dotted lines belong to a partition different from their predecessors. The probability that such a 4-cycle starts from a random point is at least $\frac{4(r-2)^4(r-1)}{r^{11}}$.

We have not been able to design or to find in the literature short-cycle reduction methods that do not introduce other (lower probability) short cycles. We therefore turn our attention to cycle detection and escape methods.

3.2 Cycle Detection and Escape

Recurring cycles. The cycle detection and escape method from [9] described in Section 2.3, does not prevent recurrence to the same cycle. When using $\mathfrak{f}_{\ell(\mathfrak{p})+c}$ to escape (we fixed $c = 4$ as it worked as well as any other choice $\ne 1$), Fig. 4 depicts how the (wavy) escape from the (solid) 4-cycle recurs to the 4-cycle via one of the dotted possibilities. The probability of recurrence depends on the escape method and on which point in the cycle the walk recurs to. With $\mathfrak{f}_{\ell(\mathfrak{p})+c}$ as escape, immediate recurrence to the escape point happens with probability $\frac{1}{2r}$ when no cycle reduction is used, recurrence happens with probability at least $\frac{1}{2r^2}$ with 2-cycle reduction, and with probability at least $\frac{(r-2)^2}{r^4}$ with 4-cycle and thus 2-cycle reduction. Similar recurrences occur, with lower probabilities, when $\mathfrak{f}'$ or $\mathfrak{f}''_{\ell(\mathfrak{p})}$ are used to escape.

Lemma 2. *Lower bounds for the probabilities to enter 2-cycles or 4-cycles or to recur to cycles for three different cycle escape methods are listed in Table 2*

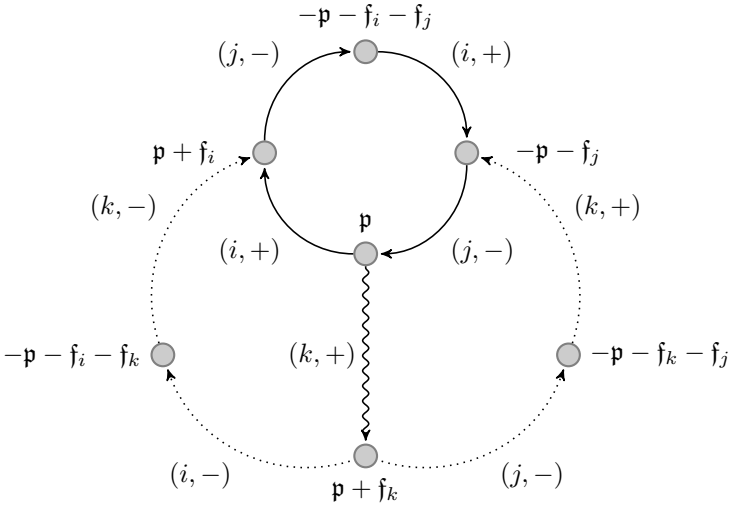


Fig. 4. Escaping from a fruitless 4-cycle, and recurring to it ($i \neq j \neq k \neq i$)

if no cycle reduction, or 2-cycle reduction (f), or 4-cycle reduction (g) is used, along with a lower bound for the slowdown factor caused by f or g.

Proof. The proofs for many entries of Table 2 were given earlier. We prove the entries in rows four and five.

Let $\mathfrak{p}$ be the escape point and let $\mathfrak{q}$ be the point it escapes to. Using $\mathfrak{f}'$ or $\mathfrak{f}''_{\ell(\mathfrak{p})}$ one can recur to the escape point $\mathfrak{p}$ by entering another cycle at $\mathfrak{q}$ and escaping from it at $\mathfrak{q}$ again. This new cycle could be a 2-cycle. For this to happen the first escape step to $\mathfrak{q}$ has to involve a negation (probability $\frac{1}{2}$), a 2-cycle has to be entered at $\mathfrak{q}$ (probabilities in first row, but see below), the escape point of this 2-cycle has to be $\mathfrak{q}$ (probability $\frac{1}{2}$), and, in the case of $\mathfrak{f}''_i$, the partition that $\mathfrak{q}$ belongs to has to be the same as the one $\mathfrak{p}$ belongs to (probability $\frac{1}{r}$). In the case of 4-cycle reduction the probability to enter a 2-cycle at $\mathfrak{q}$ is slightly lower since we do not have the information that $\ell(\sim(\mathfrak{q} + \mathfrak{f}_{\ell(\mathfrak{q})})) \neq \ell(\mathfrak{q})$; a calculation analogous to the one done at the end of Section 3.1 produces the values listed in the table. □

6-cycles. With proper $\mathfrak{f}_i$ and no sub-cycle, a common 6-cycle is of the form

$$\mathfrak{p} \xrightarrow{(i,+)} \mathfrak{p}+\mathfrak{f}_i \xrightarrow{(j,-)} -\mathfrak{p}-\mathfrak{f}_i-\mathfrak{f}_j \xrightarrow{(k,+)} -\mathfrak{p}-\mathfrak{f}_i-\mathfrak{f}_j+\mathfrak{f}_k \xrightarrow{(i,+)} -\mathfrak{p}-\mathfrak{f}_j+\mathfrak{f}_k \xrightarrow{(j,-)} \mathfrak{p}-\mathfrak{f}_k \xrightarrow{(k,+)} \mathfrak{p}$$

($i \neq j \neq k \neq i$) where with appropriate sign changes steps four and five may be swapped. It may be entered at any of its six points and occurs, when using 4-cycle reduction, with probability $\frac{1}{4r^3} + O(\frac{1}{r^4})$. A lower bound to recur to it follows by multiplying this probability with the recurring probabilities from Table 2.

Table 2. Summary of effect of cycle reduction, detection, and escape methods. With the exception of the two bold entries, all figures are lower bounds.

	Cycle reduction method:	none	2-cycle	4-cycle
Probability to enter	2-cycle	$\mathbf{\frac{1}{2r}}$	$\frac{1}{2r^3}$	$\frac{2(r-2)^2}{(r-1)r^4}$
	4-cycle	$\mathbf{\frac{r-1}{4r^3}}$	$\frac{r-1}{4r^3}$	$\frac{4(r-2)^4(r-1)}{r^{11}}$
Probability to recur to escape point using	$\mathfrak{f}_{\ell(\mathfrak{p})+c}$	$\frac{1}{2r}$	$\frac{1}{2r^2}$	$\frac{(r-2)^2}{r^4}$
	$\mathfrak{f}'$	$\frac{1}{8r}$	$\frac{1}{8r^3}$	$\frac{(r-2)^2}{2r^5}$
	$\mathfrak{f}''_{\ell(\mathfrak{p})}$	$\frac{1}{8r^2}$	$\frac{1}{8r^4}$	$\frac{(r-2)^2}{2r^6}$
Slowdown factor of iteration function		n/a	$\frac{r+1}{r}$	$\frac{r+4}{r}$

3.3 Alternative Approaches

The purpose of using the negation map is to obtain a speedup, hopefully by a factor of $\sqrt{2}$. From Fig. 1 it follows that large r-values cannot be used. From Table 2 it follows that for small r-values and relevant q-values fruitless cycles are likely to occur and recur. Medium r-values look the most promising, but are not compatible with all environments.

Since fruitless cycle occurrence and recurrence cannot be rooted out, alternative methods are needed if we want to make the negation map useful. In this section several possibilities are offered.

Heuristic. *A cycle with at least one doubling is most likely not fruitless.*

Proof. Let $\mathfrak{p} = u\mathfrak{g} + v\mathfrak{h}$ be a point on the cycle. The subsequent points are obtained by adding one of the $\mathfrak{f}_i$ or by doubling, and negating if needed, thus are up to sign linear combinations of the $\mathfrak{f}_i$ and a power-of-two multiple of $\mathfrak{p}$. If $c \geq 1$ is the number of doublings in the cycle, we get a relation of the form

$$\mathfrak{p} = \pm 2^c \mathfrak{p} + \sum_{i=0}^{r-1} c_i \mathfrak{f}_i = \pm 2^c \mathfrak{p} + \sum_{i=0}^{r-1} c_i u_i \mathfrak{g} + \sum_{i=0}^{r-1} c_i v_i \mathfrak{h} \qquad \text{and thus}$$

$$\left((1 \mp 2^c)u - \sum_{i=0}^{r-1} c_i u_i\right)\mathfrak{g} + \left((1 \mp 2^c)v - \sum_{i=0}^{r-1} c_i v_i\right)\mathfrak{h} = 0,$$

where $c_i \in \mathbf{Z}$. Since $1 \mp 2^c \neq 0$, the expression $\left((1 \mp 2^c)u - \sum_{i=0}^{r-1} c_i u_i\right)$ is most likely not divisible by the group order. This also holds if $\{\mathfrak{f}_i : 0 \leq i < r\}$ is enlarged with $\mathfrak{f}'$ or with $\{\mathfrak{f}''_i : 0 \leq i < r\}$. This concludes our heuristic argument.

Cycle reduction by doubling. The regular structure required for cycles is caused by repeated addition and subtraction using the same set of constants. This structure would be broken effectively by using an occasional doubling, i.e., a mixed walk. If such walks are used, the heuristics suggest that cycles occur

only between two doublings. If the doubling frequency is sufficiently high, only short cycles would have to be dealt with.

As borne out by expressions (1) and (2) when using the idealized values $p_i = \frac{1}{r+s}$ for $0 \le i < r$ and $p_D = \frac{s}{r+s}$ for $r > 0$, and as supported by the experiments reported in Table 1, an $r+s$-mixed walk with $s > 1$ always displays noticeably less random behavior than a well-partitioned r'-adding walk for any $r' > r$. Nevertheless, using properly tuned $r+s$-mixed walks may be a way to address the cycle problem while avoiding impractically large r-values.

However, $r+s$-mixed walks have disadvantages caused by the underlying arithmetic. Given the relative speeds of addition and doubling, an $r+s$-mixed walk is $\frac{r+7s/6}{r+s}$ times slower than an r-adding walk. In a SIMD environment where many walks are processed simultaneously, per step a fraction of about $\frac{r}{r+s}$ of the walks will do an addition, whereas the others do a doubling. If the addition and doubling code differ, as is the case for the affine Weierstrass representation, the two types of steps cannot be executed simultaneously. Thus, in such environments, to avoid a slowdown by a factor of more than 2 one needs to swap walks to make all parallel step-operations identical (at non-negligible overhead), or one has to settle for a suboptimal affine point representation that allows identical code. SIMD-application of the negation map and the possibility of another point representation are subjects for further study.

Doubling based cycle reduction and escape. Taking into account that doubling should not be used too frequently, usage could be limited to cycle reduction or escape. This would not solve the SIMD-issue, but the relative inefficiency and non-randomness would be addressed. If doublings are used to escape from fruitless cycles, they would not recur, as that would contradict the heuristics. Cycle reduction using doubling replaces $f(\mathfrak{p})$ and $g(\mathfrak{p})$ by $\bar{f}(\mathfrak{p})$ and $\bar{g}(\mathfrak{p})$, respectively, where

$$\bar{f}(\mathfrak{p}) = \begin{cases} \sim(\mathfrak{p} + \mathfrak{f}_{\ell(\mathfrak{p})}) & \text{if } \ell(\mathfrak{p}) \neq \ell(\sim(\mathfrak{p} + \mathfrak{f}_{\ell(\mathfrak{p})})), \\ \sim(2\mathfrak{p}) & \text{otherwise,} \end{cases}$$

$$\bar{g}(\mathfrak{p}) = \begin{cases} \mathfrak{q} = \sim(\mathfrak{p} + \mathfrak{f}_{\ell(\mathfrak{p})}) & \text{if } \ell(\mathfrak{q}) \neq \ell(\mathfrak{p}) \neq \ell(\sim(\mathfrak{q} + \mathfrak{f}_{\ell(\mathfrak{q})})) \neq \ell(\mathfrak{q}), \\ \sim(2\mathfrak{p}) & \text{otherwise.} \end{cases}$$

It follows from the heuristics that these functions avoid recurring fruitless cycles.

Alternative cycle detection. Because shorter cycles are more frequent, a potentially interesting modification of the cycle detection method from [9] (described at the end of Section 2.3) would be to occasionally compare a point to its kth successor, where k is the least common multiple of all even short cycle lengths that one wants to catch. Detecting, for instance, cycles up to length 12 requires only $\frac{1}{120}$th comparison per step. This can be done in several steps, recording every 12th point to catch 4- and 6-cycles, recording every 10th of these recorded points to catch 8- and 10-cycles, etc. It can be combined with the regular method with large α and β to catch longer cycles infrequently.

However, if a cycle has been detected the k points need to be recorded as before, so an escape point can be chosen deterministically. This argues against

using large k. It also suggests that an improvement can be expected only if cycles occur with low probability, and therefore that the improvement will be marginal at best (cf. α and β choices in Section 4). For this reason we did not conduct extensive experiments with this method.

4 Comparison

We implemented and compared on a traditional non-SIMD platform all previously published and newly proposed methods to deal with fruitless cycles when using the negation map. Here we report on our findings. It quickly turned out that the cycle detection methods from [9] when combined with doubling based cycle reduction and escape, are considerably more efficient than $r+s$-mixed walks with their on average slower steps and less random behavior. Mixed walks are therefore not further discussed. Experiments with the alternative cycle detection method were quickly abandoned as well.

For each combination of iteration function, escape method, and r-value a search was conducted to determine the α and β to be used for the cycle detection method from [9]. Using a heuristic argument that for $\beta = 2k$ with k much smaller than r, cycles of length $\geq \beta$ occur with probability on the order of $\frac{(k-1)!}{(2r)^k}$, values for k that make this probability low enough resulted in good initial values for the search for close to optimal α and β. To give some examples, for "f, e," as explained in Table 3 we used $\alpha = 31$ and $\beta = 20$ for $r = 16$, $\alpha = 3264$ and $\beta = 12$ for $r = 128$, and $\alpha = 52\,418$ and $\beta = 10$ for $r = 256$. For "$\bar{f}$, $\bar{\text{e}}$" and the same r-values we used the same β-values but replaced the α-values by 1 618, 838 848, and 53 687 081, respectively.

Each of the benchmarks presented in Table 3 was run on a single core of an AMD Phenom 2.2GHz 4-core processor, with each of the four cores processing a different combination. A 10-bit distinguishing property was used to get a significant amount of data in a reasonable amount of time. This somewhat affects the performance, but not the cycle behavior as walks continue after hitting a distinguished point. The figures in millions as given in the table are thus an underestimate for the actual per-core yield in units when a more realistic 30-bit distinguishing property would be used (since $2^{30}/2^{10} = 2^{20} \approx 10^6$).

In order to be able to compare the long term yield figures, the expected number of steps must be taken into account using expressions 1 and 2. As a result, the yields are corrected by a factor of $(\frac{r-1}{r})^{\frac{1}{2}}$ for the iteration functions that do not use the negation map, and by a factor of $(\frac{2r-1}{r})^{\frac{1}{2}}$ for the others, with an extra factor of $(\frac{r}{r-1})^{\frac{1}{2}}$ for g and $\bar{g}$. After this correction, the best iteration function without the negation map is the one with $r = 64$. Comparing that one with each iteration function that uses the negation map, thus boosting the latter's yield ratio by a factor of $C = ((\frac{2r-1}{r})/(\frac{63}{64}))^{\frac{1}{2}}$ or $C = ((\frac{2r-1}{r-1})/(\frac{63}{64}))^{\frac{1}{2}}$ for g and $\bar{g}$, leads to the long term speedup figure given in Table 3. Note that the correction factor C depends on the iteration function, and is close to and for some r larger than $\sqrt{2}$.

Table 3. For the (iteration function, escape method, r-value) combinations specified, the non-italics entries list the long term yield (millions of distinguished points, found during the second half hour) and the long term speedup over the best r-value ($r = 64$) without the negation map, taking into account the correction factor C as explained in the text. Cycle detection and subsequent escape by adding $\mathfrak{f}_{\ell(\mathfrak{p})+4}$, $\mathfrak{f}'$, $\mathfrak{f}''_{\ell(\mathfrak{p})}$ and by doubling is indicated by "e," "e′," "e″" and by "ē," respectively. The iteration functions f (2-cycle reduction), g (4-cycle and 2-cycle reduction), $\bar{f}$ (2-cycle reduction using doubling), and $\bar{g}$ (4-cycle and 2-cycle reduction using doubling) are as in sections 2.3, 3.1 and 3.3. The yields are for 256 parallel walks (sharing the inversion) for a 131-bit ECDLP with a 131-bit prime order group. The yields during the first half hour are almost consistently higher, considerably so for poorly performing combinations. They are not meaningful and are thus not listed. The italics entries are A above D, followed by the maximal achievable speedup factor of $\frac{C(10^9-A)}{10^9+D/6}$, as explained in the text.
†: This applies to "no reduction, no escape," "just f," "just $\bar{f}$," "just e," and "just e′."

	$r = 16$	$r = 32$	$r = 64$	$r = 128$	$r = 256$	$r = 512$
Without negation map						
	7.29: 0.98	7.28: 0.99	**7.27**: 1.00	7.19: 0.99	6.97: 0.96	6.78: 0.94
With negation map						
†	0.00: 0.00	0.00: 0.00	0.00: 0.00	0.00: 0.00	0.00: 0.00	0.00: 0.00
just g	0.00: 0.00	0.00: 0.00	0.00: 0.00	0.00: 0.00	0.04: 0.01	3.59: 0.70
just $\bar{g}$	0.00: 0.00	0.00: 0.00	0.00: 0.00	0.75: 0.15	4.90: 0.96	5.90: 1.16
just e″	0.00: 0.00	0.00: 0.00	0.00: 0.00	0.61: 0.12	4.94: 0.97	5.73: 1.12
just ē	3.34: 0.64	4.89: 0.95	5.85: 1.14	6.10: 1.19	6.28: 1.23	6.18: 1.21
f, e	0.00: 0.00 $^{9.4e8}_{0.0e0}\}0.08$	0.00: 0.00 $^{6.6e8}_{0.0e0}\}0.48$	1.52: 0.30 $^{1.0e8}_{0.0e0}\}1.28$	5.93: 1.16 $^{3.6e7}_{0.0e0}\}1.37$	6.47: 1.27 $^{2.9e7}_{0.0e0}\}1.38$	6.36: 1.25 $^{2.5e7}_{0.0e0}\}1.39$
f, e′	0.00: 0.00 $^{3.9e8}_{0.0e0}\}0.86$	3.24: 0.63 $^{8.0e7}_{0.0e0}\}1.30$	6.04: 1.18 $^{4.6e7}_{0.0e0}\}1.35$	6.41: 1.25 $^{3.3e7}_{0.0e0}\}1.38$	6.29: 1.23 $^{2.9e7}_{0.0e0}\}1.38$	6.21: 1.22 $^{2.6e7}_{0.0e0}\}1.39$
f, e″	0.00: 0.00 $^{1.3e8}_{0.0e0}\}1.22$	5.34: 1.04 $^{6.0e7}_{0.0e0}\}1.33$	6.21: 1.21 $^{4.2e7}_{0.0e0}\}1.36$	6.30: 1.23 $^{3.3e7}_{0.0e0}\}1.38$	6.20: 1.21 $^{2.9e7}_{0.0e0}\}1.38$	5.99: 1.17 $^{2.7e7}_{0.0e0}\}1.39$
f, ē	3.71: 0.72 $^{9.2e7}_{9.9e5}\}1.27$	6.36: 1.24 $^{6.8e7}_{2.8e5}\}1.32$	6.50: 1.27 $^{4.2e7}_{6.5e4}\}1.36$	6.57: **1.29** $^{3.3e7}_{1.5e4}\}1.38$	6.47: 1.27 $^{2.9e7}_{3.8e3}\}1.38$	6.30: 1.25 $^{2.7e7}_{9.7e2}\}1.39$
g, e	0.00: 0.00 $^{8.7e8}_{0.0e0}\}0.19$	0.01: 0.00 $^{3.7e8}_{0.0e0}\}0.91$	4.89: 0.96 $^{6.6e7}_{0.0e0}\}1.34$	6.22: 1.22 $^{4.2e7}_{0.0e0}\}1.37$	6.23: 1.22 $^{3.3e7}_{0.0e0}\}1.38$	6.05: 1.19 $^{1.3e7}_{0.0e0}\}1.41$
g, e′	0.00: 0.00 $^{7.8e8}_{0.0e0}\}0.32$	0.01: 0.00 $^{3.0e8}_{0.0e0}\}1.00$	5.32: 1.05 $^{6.0e7}_{0.0e0}\}1.35$	6.26: 1.23 $^{4.1e7}_{0.0e0}\}1.37$	6.25: 1.23 $^{3.0e7}_{0.0e0}\}1.38$	6.11: 1.20 $^{5.5e7}_{0.0e0}\}1.35$
g, e″	0.00: 0.00 $^{7.6e8}_{0.0e0}\}0.34$	1.09: 0.21 $^{1.2e8}_{0.0e0}\}1.27$	5.37: 1.13 $^{6.0e7}_{0.0e0}\}1.35$	6.08: 1.20 $^{4.2e7}_{0.0e0}\}1.37$	6.06: 1.19 $^{3.5e7}_{0.0e0}\}1.38$	5.86: 1.15 $^{4.3e7}_{0.0e0}\}1.37$
g, ē	0.76: 0.15 $^{3.3e8}_{1.6e5}\}0.97$	5.91: 1.17 $^{1.7e8}_{6.0e4}\}1.19$	6.02: 1.18 $^{8.1e7}_{8.1e3}\}1.32$	6.25: 1.23 $^{5.4e7}_{1.0e3}\}1.35$	6.13: 1.20 $^{4.0e7}_{1.2e2}\}1.37$	6.00: 1.18 $^{2.7e7}_{9.0e0}\}1.39$
$\bar{f}$, e	0.00: 0.00 $^{8.7e8}_{2.4e6}\}0.18$	0.00: 0.00 $^{4.3e8}_{1.7e7}\}0.80$	2.70: 0.53 $^{5.4e7}_{1.5e7}\}1.34$	5.96: 1.16 $^{1.1e7}_{7.7e6}\}1.41$	6.34: 1.24 $^{1.0e7}_{3.9e6}\}1.41$	6.20: 1.21 $^{1.4e7}_{1.9e6}\}1.40$
$\bar{f}$, e′	0.01: 0.0 $^{2.6e8}_{4.3e7}\}1.03$	4.24: 0.82 $^{6.8e7}_{2.9e7}\}1.31$	6.32: 1.23 $^{3.9e7}_{1.5e7}\}1.36$	6.43: 1.26 $^{3.2e7}_{7.6e6}\}1.38$	6.33: 1.24 $^{2.8e7}_{3.8e6}\}1.38$	6.20: 1.22 $^{2.7e7}_{1.9e6}\}1.39$
$\bar{f}$, e″	1.34: 0.26 $^{8.9e7}_{5.2e7}\}1.27$	5.80: 1.13 $^{5.3e7}_{2.9e7}\}1.33$	6.23: 1.22 $^{3.9e7}_{1.5e7}\}1.36$	6.21: 1.22 $^{3.6e7}_{7.5e6}\}1.37$	6.15: 1.20 $^{2.8e7}_{3.8e6}\}1.38$	6.00: 1.18 $^{2.6e7}_{1.9e6}\}1.39$
$\bar{f}$, ē	5.58: 1.06 $^{6.1e7}_{4.2e7}\}1.31$	6.14: 1.18 $^{3.7e7}_{3.0e7}\}1.36$	6.34: 1.23 $^{1.8e7}_{1.5e7}\}1.39$	6.42: 1.25 $^{1.1e7}_{7.7e6}\}1.41$	6.27: 1.23 $^{1.0e7}_{3.9e6}\}1.41$	6.07: 1.19 $^{1.4e7}_{1.9e6}\}1.40$
$\bar{g}$, e	2.56: 0.51 $^{1.4e8}_{9.9e7}\}1.23$	5.80: 1.15 $^{7.9e7}_{5.6e7}\}1.31$	6.02: 1.18 $^{5.1e7}_{2.9e7}\}1.35$	6.09: 1.20 $^{4.1e7}_{1.5e7}\}1.37$	6.19: 1.21 $^{2.6e7}_{7.6e6}\}1.39$	5.74: 1.13 $^{7.7e6}_{3.9e6}\}1.41$
$\bar{g}$, e′	4.74: 0.94 $^{1.2e8}_{1.0e8}\}1.25$	5.88: 1.16 $^{7.8e7}_{5.6e7}\}1.31$	6.14: 1.21 $^{5.3e7}_{2.9e7}\}1.35$	6.28: 1.23 $^{3.9e7}_{1.5e7}\}1.37$	6.05: 1.19 $^{2.6e7}_{7.6e6}\}1.39$	5.80: 1.14 $^{7.7e6}_{3.9e6}\}1.41$
$\bar{g}$, e″	4.72: 0.94 $^{1.2e8}_{1.0e8}\}1.25$	5.80: 1.15 $^{7.7e7}_{5.6e7}\}1.31$	6.08: 1.20 $^{5.3e7}_{2.9e7}\}1.35$	6.05: 1.19 $^{3.8e7}_{1.5e7}\}1.37$	5.91: 1.16 $^{1.8e7}_{7.6e6}\}1.40$	5.67: 1.11 $^{7.7e6}_{3.9e6}\}1.41$
$\bar{g}$, ē	4.83: 0.96 $^{1.2e8}_{1.0e8}\}1.25$	5.87: 1.16 $^{7.9e7}_{5.6e7}\}1.31$	6.09: 1.20 $^{5.2e7}_{2.9e7}\}1.35$	6.16: 1.21 $^{4.0e7}_{1.5e7}\}1.37$	6.09: 1.20 $^{2.6e7}_{7.6e6}\}1.39$	5.70: 1.12 $^{7.7e6}_{3.9e6}\}1.41$

Non-doubling 2-cycle reduction (f) with doubling-based cycle escape ($\bar{\text{e}}$) and $r = 128$ performed best, with an overall speedup by a factor of 1.29: although fewer distinguished points are found than for the best case without the negation map ($r = 64$), there is a considerable overall gain because fewer distinguished points (by a factor of C, for the relevant C) should suffice. For $r = 16$ most iteration functions with the negation map perform poorly.

We measured to what extent our failure to achieve a speedup by a factor of $\sqrt{2}$ can be blamed on cycle detection and escape and other overheads, and which part is due to the higher average cost of the iteration function. For most combinations in Table 3 we counted the number S of *useful* steps performed when doing 10^9 group operations, while keeping track of the number D of doublings among them. Here a step is useful if it is not taken as part of a fruitless cycle, so all D doublings are useful. Without the negation map, S would be 10^9 and $D = 0$; this is the basis for the comparison. With the negation map, $A = 10^9 - S$ is counted as the number of *additional* additions due to cycle reductions or fruitless cycles. The inherent slowdown of that iteration function is then $1 + \frac{A+D/6}{S}$, so that it can achieve a speedup by a factor of at most $\frac{CS}{S+A+D/6} = \frac{C(10^9-A)}{10^9+D/6}$, with C as defined above.

Based on Table 3 and Fig. 1, we conclude that our failure to better approach the optimal speedup by a factor of $\sqrt{2}$ is due to an onset of cache effects combined with various overheads. The italics figures from Table 3 make us believe that improvements may be obtained when using better implementations.

Previous results. The only publication that we know that presents practical data about Pollard's rho method used with the negation map is [8]. Only relatively small ECDLPs were solved (42- and 43-bit prime fields) and small r-values were avoided. The adverse cycle behavior that we witnessed can therefore not be expected and we doubt if the results reported are significant for the sizes that we consider. Only mixed walks were used, and an overall speedup by a factor of about 1.35 was reported. Cycle escaping was done by jumping to the sum of all points in a cycle, which cannot be expected to work in general because the sum may depend just on the addition constants.

5 Conclusion

With judicious application of doubling, usage of the negation map to solve ECDLPs over prime fields using Pollard's rho method can indeed be recommended. In the best of circumstances that we have been able to create, however, the speedup falls short of the hoped for $\sqrt{2}$, but is with 1.29 still considerable.

This conclusion does not apply to SIMD-environments where occasional doublings cause considerable delays. Alternative point representations need to be considered to assess the usefulness of the negation map for SIMD platforms, in particular because such platforms are becoming popular again.

Acknowledgements. This work was supported by the Swiss National Science Foundation under grant numbers 200021-119776 and 206021-117409 and

by EPFL DIT. We gratefully acknowledge useful suggestions by Marcelo E. Kaihara and very insightful comments by the ANTS reviewers.

References

1. Avanzi, R.M., Cohen, H., Doche, C., Frey, G., Lange, T., Nguyen, K., Vercauteren, F.: Handbook of Elliptic and Hyperelliptic Curve Cryptography. Chapman & Hall/CRC (2006)
2. Bailey, D.V., et al.: Breaking ECC2K-130. In: Cryptology ePrint Archive, Report 2009/541 (2009), http://eprint.iacr.org/
3. Bos, J.W., Kaihara, M.E., Montgomery, P.L.: Pollard rho on the PlayStation 3. In: Workshop record of SHARCS 2009, pp. 35–50 (2009), http://www.hyperelliptic.org/tanja/SHARCS/record2.pdf
4. Brent, R.P., Pollard, J.M.: Factorization of the eighth Fermat number. Math. Comp. 36(154), 627–630 (1981)
5. Certicom. Certicom ECC Challenge (1997), http://www.certicom.com/images/pdfs/cert_ecc_challenge.pdf
6. Certicom. Press release: Certicom announces elliptic curve cryptosystem (ECC) challenge winner (2002), http://www.certicom.com/index.php/2002-press-releases/38-2002-press-releases/340-notre-dame-mathematician-solves-eccp-109-encryption-key-problem-issued-in-1997
7. Duursma, I.M., Gaudry, P., Morain, F.: Speeding up the discrete log computation on curves with automorphisms. In: Lam, K.-Y., Okamoto, E., Xing, C. (eds.) ASIACRYPT 1999. LNCS, vol. 1716, pp. 103–121. Springer, Heidelberg (1999)
8. Escott, A.E., Sager, J.C., Selkirk, A.P.L., Tsapakidis, D.: Attacking elliptic curve cryptosystems using the parallel Pollard rho method. CryptoBytes Technical Newsletter 4(2), 15–19 (1999), ftp.rsasecurity.com/pub/cryptobytes/crypto4n2.pdf
9. Gallant, R.P., Lambert, R.J., Vanstone, S.A.: Improving the parallelized Pollard lambda search on anomalous binary curves. Math. Comp. 69(232), 1699–1705 (2000)
10. Harley, R.: Elliptic curve discrete logarithms project, http://pauillac.inria.fr/~harley/
11. Koblitz, N.: Elliptic curve cryptosystems. Math. Comp. 48, 203–209 (1987)
12. Koblitz, N.: CM-curves with good cryptographic properties. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 279–287. Springer, Heidelberg (1992)
13. Miller, V.S.: Use of elliptic curves in cryptography. In: Williams, H.C. (ed.) CRYPTO 1985. LNCS, vol. 218, pp. 417–426. Springer, Heidelberg (1986)
14. Montgomery, P.L.: Speeding the Pollard and elliptic curve methods of factorization. Math. Comp. 48, 243–264 (1987)
15. Pollard, J.M.: Monte Carlo methods for index computation (mod p). Math. Comp. 32, 918–924 (1978)
16. Teske, E.: On random walks for Pollard's rho method. Math. Comp. 70(234), 809–825 (2001)
17. van Oorschot, P.C., Wiener, M.J.: Parallel collision search with cryptanalytic applications. Journal of Cryptology 12(1), 1–28 (1999)
18. Wiener, M.J., Zuccherato, R.J.: Faster attacks on elliptic curve cryptosystems. In: Tavares, S., Meijer, H. (eds.) SAC 1998. LNCS, vol. 1556, pp. 190–200. Springer, Heidelberg (1999)

An $O(M(n)\log n)$ Algorithm for the Jacobi Symbol

Richard P. Brent[1] and Paul Zimmermann[2]

[1] Australian National University, Canberra, Australia
[2] INRIA Nancy - Grand Est, Villers-lès-Nancy, France

Abstract. The best known algorithm to compute the Jacobi symbol of two n-bit integers runs in time $O(M(n)\log n)$, using Schönhage's fast continued fraction algorithm combined with an identity due to Gauss. We give a different $O(M(n)\log n)$ algorithm based on the binary recursive gcd algorithm of Stehlé and Zimmermann. Our implementation — which to our knowledge is the first to run in time $O(M(n)\log n)$ — is faster than GMP's quadratic implementation for inputs larger than about 10000 decimal digits.

1 Introduction

We want to compute the Jacobi symbol[1] $(b|a)$ for n-bit integers a and b, where a is odd positive. We give three algorithms based on the 2-adic gcd from Stehlé and Zimmermann [13]. First we give an algorithm whose worst-case time bound is $O(M(n)n^2) = \widetilde{O}(n^3)$; we call this the *cubic* algorithm although this is pessimistic since the algorithm is quadratic on average as shown in [5], and probably also in the worst case. We then show how to reduce the worst-case to $O(M(n)n) = \widetilde{O}(n^2)$ by combining sequences of "ugly" iterations (defined in Section 1.1) into one "harmless" iteration. Finally, we obtain an algorithm with worst-case time $O(M(n)\log n)$. This is, up to a constant factor, the same as the time bound for the best known algorithm, apparently never published in full, but sketched in Bach [1] and in more detail in Bach and Shallit [2] (with credit to Bachmann [3]).

The latter algorithm makes use of the Knuth-Schönhage fast continued fraction algorithm [9] and an identity of Gauss [6]. Although this algorithm has been attributed to Schönhage, Schönhage himself gives a different $O(M(n)\log n)$ algorithm [10,15] which does not depend on the identity of Gauss. The algorithm is mentioned in Schönhage's book [11, §7.2.3], but no details are given there.

With our algorithm it is not necessary to compute the full continued fraction or to use the identity of Gauss for the Jacobi symbol. Thus, it provides an alternative that may be easier to implement.

[1] Notation: we write the Jacobi symbol as $(b|a)$, since this is easier to typeset and less ambiguous than the more usual $\left(\frac{b}{a}\right)$. Also, $M(n)$ is the time to multiply n-bit numbers, and $\widetilde{O}(f(n))$ means $O(f(n)(\log f(n))^c)$ for some constant $c \geq 0$.

G. Hanrot, F. Morain, and E. Thomé (Eds.): ANTS-IX 2010, LNCS 6197, pp. 83–95, 2010.

It is possible to modify some of the other fast GCD algorithms considered by Möller [8] to compute the Jacobi symbol, but we do not consider such possibilities here. At best they give a small constant factor speedup over our algorithm.

We recall the main identities satisfied by the Jacobi symbol: $(bc|a) = (b|a)(c|a)$; $(2|a) = (-1)^{(a^2-1)/8}$; $(b|a) = (-1)^{(a-1)(b-1)/4}(a|b)$ for a, b odd; and $(b|a) = 0$ if $(a, b) \neq 1$.

Note that all our algorithms compute $(b|a)$ with b even positive and a odd positive. For the more general case where b is any integer, we can reduce to b even and positive using $(b|a) = (-1)^{(a-1)/2}(-b|a)$ if b is negative, and $(b|a) = (b+a|a)$ if b is odd.

We first describe a cubic algorithm to compute the Jacobi symbol. The quadratic algorithm in Section 2 is based on this cubic algorithm, and the subquadratic algorithm in Section 3 uses the same ideas as the quadratic algorithm but with an asymptotically fast recursive implementation.

For $a \in \mathbb{Z}$, the notation $\nu(a)$ denotes the 2-adic valuation $\nu_2(a)$ of a, that is the maximum k such that $2^k|a$, or $+\infty$ if $a = 0$.

1.1 Binary Division with Positive Quotient

Throughout the paper we use the binary division with positive quotient defined by Algorithm 1.1. Compared to the "centered division" of [13], it returns a quotient in $[1, 2^{j+1} - 1]$ instead of in $[1 - 2^j, 2^j - 1]$. Note that the quotient q is always odd.

Algorithm 1.1. BinaryDividePos

Input: $a, b \in \mathbb{N}$ with $\nu(a) = 0 < \nu(b) = j$
Output: q and $r = a + qb/2^j$ such that $0 < q < 2^{j+1}$, $\nu(b) < \nu(r)$
1: $q \leftarrow -a/(b/2^j) \bmod 2^{j+1}$ ▷ q is odd and positive
2: **return** $q, r = a + qb/2^j$.

With this binary division, we define Algorithm CubicBinaryJacobi, where the fact that the quotient q is positive ensures that all a, b terms computed remain positive, and a remains odd, thus $(b|a)$ remains well-defined.[2]

Theorem 1. *Algorithm CubicBinaryJacobi is correct (assuming it terminates).*

Proof. We prove that the following invariant holds during the algorithm, if a_0, b_0 are the initial values of a, b:

$$(b_0|a_0) = (-1)^s(b|a).$$

This is true before we enter the while-loop, since $s = 0$, $a = a_0$, and $b = b_0$. For each step in the while loop, we divide b by 2^j, swap a and $b' = b/2^j$, replace a

[2] Möller says in [8]: "*if one tries to use positive quotients* $0 < q < 2^{k+1}$*, the* [binary gcd] *algorithm no longer terminates*". However, with a modified stopping criterion as in Algorithm CubicBinaryJacobi, the algorithm terminates (we prove this below).

Algorithm 1.2. CubicBinaryJacobi

Input: $a, b \in \mathbb{N}$ with $\nu(a) = 0 < \nu(b)$
Output: Jacobi symbol $(b|a)$

```
1: s ← 0,    j ← ν(b)
2: while 2^j a ≠ b do
3:     b' ← b/2^j
4:     (q, r) ← BinaryDividePos(a, b)
5:     s ← (s + j(a^2 − 1)/8 + (a − 1)(b' − 1)/4 + j(b'^2 − 1)/8) mod 2
6:     (a, b) ← (b', r/2^j),    j ← ν(b)
7: if a = 1 then return (−1)^s else return 0
```

by $r = a + qb'$, and divide r by 2^j. The Jacobi symbol is modified by a factor $(-1)^{j(a^2-1)/8}$ for the division of b by 2^j, by a factor $(-1)^{(a-1)(b'-1)/4}$ for the interchange of a and b', and by a factor $(-1)^{j(b'^2-1)/8}$ for the division of r by 2^j. At the end of the loop, we have $\gcd(a_0, b_0) = a$; if $a = 1$, since $(b|1) = 1$, we have $(b_0|a_0) = (-1)^s$, otherwise $(b_0|a_0) = 0$. □

Lemma 1. *The quantity $a + 2b$ is non-increasing in Algorithm CubicBinaryJacobi.*

Proof. At each iteration of the "while" loop, a becomes $b/2^j$, and b becomes $(a + qb/2^j)/2^j$. In matrix notation

$$\begin{pmatrix} a \\ b \end{pmatrix} \leftarrow \begin{pmatrix} 0 & 1/2^j \\ 1/2^j & q/2^{2j} \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix}. \tag{1}$$

Therefore $a + 2b$ becomes

$$\frac{b}{2^j} + 2\left(\frac{a + qb/2^j}{2^j}\right) = \frac{2a}{2^j} + (1 + 2q/2^j)\frac{b}{2^j}. \tag{2}$$

Since $j \geq 1$, the first term is bounded by a. In the second term, $q \leq 2^{j+1} - 1$, thus the second term is bounded by $(5/2^j - 2/2^{2j})b$, which is bounded by $9b/8$ for $j \geq 2$, and equals $2b$ for $j = 1$. □

If $j \geq 2$, then $a + 2b$ is multiplied by a factor at most $9/16$. If $j = q = 1$ then $a + 2b$ decreases, but by a factor which could be arbitrarily close to 1. The only case where $a + 2b$ does not decrease is when $j = 1$ and $q = 3$; in this case $a + 2b$ is unchanged.

This motivates us to define three classes of iterations: *good*, *bad*, and *ugly*. Let us say that we have a *good* iteration when $j \geq 2$, a *bad* iteration when $j = q = 1$, and an *ugly* iteration when $j = 1$ and $q = 3$. Since q is odd and $1 \leq q \leq 2^{j+1} - 1$, this covers all possibilities. For a bad iteration, (a, b) becomes $(b/2, a/2 + b/4)$, and for an ugly iteration, (a, b) becomes $(b/2, a/2 + 3b/4)$. We denote the matrices corresponding to good, bad and ugly iterations by G, B and U respectively. Thus

$$G = G_{j,q} = \begin{pmatrix} 0 & 1/2^j \\ 1/2^j & q/4^j \end{pmatrix},\ B = \begin{pmatrix} 0 & 1/2 \\ 1/2 & 1/4 \end{pmatrix},\ U = \begin{pmatrix} 0 & 1/2 \\ 1/2 & 3/4 \end{pmatrix}.$$

The effect of m successive ugly iterations is easily seen to be given by the matrix

$$U^m = \frac{1}{5}\begin{pmatrix} 1+4(-1/4)^m & 2-2(-1/4)^m \\ 2-2(-1/4)^m & 4+(-1/4)^m \end{pmatrix}. \tag{3}$$

Assume we start from $(a, b) = (a_0, b_0)$, and after $m > 0$ successive ugly iterations we get values (a_m, b_m). Then, from Equation (3),

$$5a_m = (a + 2b) + 2(2a - b)(-1/4)^m, \tag{4}$$
$$5b_m = 2(a + 2b) - (2a - b)(-1/4)^m. \tag{5}$$

We can not have $2a_0 = b_0$ or the algorithm would have terminated. However, a_m must be an integer. This gives an upper bound on m. For a_0, b_0 of n bits, the number of successive ugly iterations is bounded by $n/2 + O(1)$ (a precise statement is made in Lemma 2).

If there were no bad iterations, this would prove that for n-bit inputs the number of iterations is $O(n^2)$, since each sequence of ugly iterations would be followed by at least one good iteration. Bad iterations can be handled by a more complicated argument which we omit, since they will be considered in detail in §2 when we discuss the complexity of the quadratic algorithm (see the proof of Theorem 2).

Since the number of iterations is $O(n^2)$ from Theorem 2, and each iteration costs time $O(M(n))$, the overall time for Algorithm CubicBinaryJacobi is $O(n^2 M(n)) = \widetilde{O}(n^3)$. Note that this worst-case bound is almost certainly too pessimistic (see §4).

2 A Provably Quadratic Algorithm

Suppose we have a sequence of $m > 0$ ugly iterations. It is possible to combine the m ugly iterations into one *harmless* iteration which is not much more expensive than a normal (good or bad) iteration. Also, it is possible to predict the maximal such m in advance. Using this trick, we reduce the number of iterations (good, bad and harmless) to $O(n)$ and their cost to $O(M(n)n) = \widetilde{O}(n^2)$. Without loss of generality, suppose that we start from $(a_0, b_0) = (a, b)$.

Lemma 2. *If $\mu = \nu(a - b/2)$, then we have exactly $\lfloor \mu/2 \rfloor$ ugly iterations starting from (a, b), followed by a good iteration if μ is even, and by a bad iteration if μ is odd.*

Proof. We prove the lemma by induction on μ. If $\mu = 0$, $a - b/2$ is odd, but a is odd, so $b/2$ is even, which yields $j \geq 2$ in BinaryDividePos, thus a, b yield a good iteration. If $\mu = 1$, $a - b/2$ is even, which implies that $b/2$ is odd, thus we have $j = 1$. If we had $q = 3$ in BinaryDividePos, this would mean that

$a + 3(b/2) = 0 \bmod 4$, or equivalently $a - b/2 = 0 \bmod 4$, which is incompatible with $\mu = 1$. Thus we have $q = 1$, and a bad iteration.

Now assume $\mu \geq 2$. The first iteration is ugly since 4 divides $a - b/2$, which implies that $b/2$ is odd. Thus $j = 1$, and $a - b/2 = 0 \bmod 4$ implies that $q = 3$. After one ugly iteration (a, b) becomes $(b/2, a/2 + 3b/4)$, thus $a - b/2$ becomes $-(a - b/2)/4$, and the 2-valuation of $a - b/2$ decreases by 2. □

From the above, we see that, for a sequence of m ugly iterations, $a_0, a_1, \ldots, a_m$ satisfy the three-term recurrence

$$4a_{i+1} - 3a_i - a_{i-1} = 0 \text{ for } 0 < i < m,$$

and similarly for $b_0, b_1, \ldots, b_m$. It follows that $a_i = a \bmod 4$, and similarly $b_i = b \bmod 4$, for $1 \leq i < m$.

We can modify Algorithm CubicBinaryJacobi to consolidate m consecutive ugly iterations into one harmless iteration, using the expressions (4)–(5) for a_m and b_m (we give an optimised evaluation below). It remains to modify step 5 of CubicBinaryJacobi to take account of the m updates to s. Since $j = 1$ for each ugly iteration, we have to increment s by an amount

$$\delta = \sum_{0 \leq i < m} \left(\frac{a_i^2 - 1}{8} + \frac{{b'_i}^2 - 1}{8} + \frac{a_i - 1}{2}\frac{b'_i - 1}{2} \right) \bmod 2,$$

where we write b'_i for $b_i/2$. However, $a_{i+1} = b'_i$ for $0 \leq i < m$, so the terms involving division by 8 "collapse" mod 2, leaving just the first and last terms. The terms involving two divisions by 2 are all equal to $(a - 1)/2 \cdot (b' - 1)/2 \bmod 2$, using the observation that $a_i \bmod 4$ is constant for $0 \leq i < m$. Thus

$$\delta = \left(\frac{a_0^2 - 1}{8} + \frac{a_m^2 - 1}{8} + m\frac{a_0 - 1}{2}\frac{a_1 - 1}{2} \right) \bmod 2.$$

One further simplification is possible. Since $a_0 = a_1 \bmod 4$, and a_0 is odd, we can replace a_1 by a_0 in the last term, and use the fact that $x^2 = x \bmod 2$ to obtain

$$\delta = \left(\frac{a_0^2 - 1}{8} + \frac{a_m^2 - 1}{8} + m\frac{a_0 - 1}{2} \right) \bmod 2. \tag{6}$$

We can economise the computation of a_m and b_m from (4)–(5) by first computing

$$d = a - b', \ m = \nu(d) \text{ div } 2, \ c = (d - (-1)^m(d/4^m))/5,$$

where the divisions by 4^m and by 5 are exact; then $a_m = a - 4c$, $b_m = b + 2c$.

From these observations, it is easy to modify Algorithm CubicBinaryJacobi to obtain Algorithm QuadraticBinaryJacobi. In this algorithm, steps 7–11 implement a harmless iteration equivalent to $m > 0$ consecutive ugly iterations; steps 13–14 implement bad and good iterations, and the remaining steps are common to both. Step 5 of Algorithm CubicBinaryJacobi is split into three steps 4, 13 and 15. In the case of a harmless iteration, the computation of δ satisfying (6) is implicit in steps 4, 10 and 15.

Algorithm 2.1. QuadraticBinaryJacobi

Input: $a, b \in \mathbb{N}$ with $\nu(a) = 0 < \nu(b)$
Output: Jacobi symbol $(b|a)$

1: $s \leftarrow 0, \quad j \leftarrow \nu(b)$
2: **while** $2^j a \neq b$ **do**
3: $\quad b' \leftarrow b/2^j$
4: $\quad s \leftarrow (s + j(a^2 - 1)/8) \bmod 2$
5: $\quad (q, r) \leftarrow \text{BinaryDividePos}(a, b)$
6: $\quad$ **if** $(j, q) = (1, 3)$ **then**
7: $\quad\quad d \leftarrow a - b'$
8: $\quad\quad m \leftarrow \nu(d) \text{ div } 2$
9: $\quad\quad c \leftarrow (d - (-1)^m d/4^m)/5$
10: $\quad\quad s \leftarrow (s + m(a - 1)/2) \bmod 2$
11: $\quad\quad (a, b) \leftarrow (a - 4c, b + 2c)$ ▷ harmless iteration
12: $\quad$ **else**
13: $\quad\quad s \leftarrow (s + (a - 1)(b' - 1)/4) \bmod 2$
14: $\quad\quad (a, b) \leftarrow (b', r/2^j)$ ▷ good or bad iteration
15: $\quad s \leftarrow (s + j(a^2 - 1)/8) \bmod 2, \quad j \leftarrow \nu(b)$
16: **if** $a = 1$ **then** return $(-1)^s$ **else** return 0

Theorem 2. *Algorithm QuadraticBinaryJacobi is correct and terminates after $O(n)$ iterations of the "while" loop (steps 2–15) if the inputs are positive integers of at most n bits, with $0 = \nu(a) < \nu(b)$.*

Proof. Correctness follows from the equivalence to Algorithm CubicBinaryJacobi. To prove that convergence takes $O(n)$ iterations, we show that $a + 2b$ is multiplied by a factor at most $5/8$ in each block of three iterations. This is true if the block includes at least one good iteration, so we need only consider harmless and bad iterations. Two harmless iterations do not occur in succession, so the block must include either (harmless, bad) or (bad, bad). In the first case, the corresponding matrix is $BU^m = BU \cdot U^{m-1}$ for some $m > 0$. We saw in §1.1 that the matrix U leaves $a + 2b$ unchanged, so U^{m-1} also leaves $a + 2b$ unchanged, and we need only consider the effect of BU. Suppose that (a, b) is transformed into $(\widetilde{a}, \widetilde{b})$ by BU. Thus

$$\begin{pmatrix} \widetilde{a} \\ \widetilde{b} \end{pmatrix} = BU \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 1/4 & 3/8 \\ 1/8 & 7/16 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} .$$

We see that

$$\widetilde{a} + 2\widetilde{b} = \frac{a}{2} + \frac{5b}{4} \leq \frac{5}{8}(a + 2b).$$

The case of two successive bad iterations is similar – just replace BU by B^2 in the above, and deduce that $\widetilde{a} + 2\widetilde{b} \leq (a + 2b)/2$.

We conclude that the number of iterations of the while loop is at most $cn + O(1)$, where $c = 3/\log_2(8/5) \approx 4.4243$. □

Remarks

1. A more complicated argument along similar lines can reduce the constant c to $2/\log_2(1/\rho(BU)) = 2/\log_2((11-\sqrt{57})/2) \approx 2.5424$. Here ρ denotes the spectral radius: $\rho(A) = \lim_{k\to\infty} ||A^k||^{1/k}$.
2. In practice QuadraticBinaryJacobi is not much (if any) faster than CubicBinaryJacobi. Its advantage is simply the better worst-case time bound. A heuristic argument suggests that on average only 1/4 of the iterations of CubicBinaryJacobi are ugly.
3. Our implementations of CubicBinaryJacobi and QuadraticBinaryJacobi are slower than GMP's $O(n^2)$ algorithm (which is based on Stein's binary gcd, as in Shallit and Sorenson [12]). However, in the next section we use the ideas of our QuadraticBinaryJacobi algorithm to get an $O(M(n)\log n)$ algorithm. We do not see how to modify the algorithm of Shallit and Sorenson to do this.[3]

3 An $O(M(n)\log n)$ Algorithm

Algorithm HalfBinaryJacobi below is a modification of Algorithm Half-GB-gcd from [13]. (Algorithm Half-GB-gcd is a subquadratic right-to-left gcd algorithm; for more on the general structure of subquadratic gcd algorithms, we refer the reader to Möller [8].) The main differences between Half-GB-gcd and our algorithm are the following:

1. binary division with positive (not centered) quotient is used;
2. the algorithm returns an integer s such that if a, b are the inputs, c, d the output values defined by Theorem 3, then
$$(b|a) = (-1)^s(d|c);$$
3. at steps 4 and 27, we reduce mod 2^{2k_1+2} (resp. 2^{2k_2+2}) instead of mod 2^{2k_1+1} (resp. 2^{2k_2+1}), so that we have enough information to correctly update s_0 at steps 10, 17, 21 and 25;
4. we have to "cut" some harmless iterations in two (step 15).

Remarks. The matrix Q occurring at step 19 is just $2^{2m}U^m$, where U^m is given by Equation (3). Similarly, the matrix Q occurring at step 23 is $2^{2j_0}G_{j_0,q}$. In practice, steps 13–20 can be omitted (so the algorithm becomes a fast version of CubicBinaryJacobi) – this variant is simpler and slightly faster on average.

We now state our main theorem. Its proof is based on comparing the GB sequence of a, b and that of a_1, b_1, where $a_1 = a \bmod 2^{2k_1+2}$ and $b_1 = b \bmod 2^{2k_1+2}$. The GB — which stands for Generalized Binary division, see [13] — sequence of a, b is the sequence of remainders we obtain by applying the binary division iteratively. Two GB sequences *match* if they produce the same binary quotients q_i.

[3] In Algorithm Binary Jacobi of [12], it is necessary to know the sign of $a-n$ ($b-a$ in our notation) to decide whether to perform an interchange. This makes it difficult to construct an recursive $O(M(n)\log n)$ algorithm along the lines of Algorithm HalfBinaryJacobi.

Algorithm 3.1. HalfBinaryJacobi

Input: $a \in \mathbb{N}, b \in \mathbb{N} \cup \{0\}$ with $0 = \nu(a) < \nu(b)$, and $k \in \mathbb{N}$
Output: two integers s, j and a 2×2 matrix R

if $\nu(b) > k$ **then** ▷ $b = 0$ is possible
  Return $0, 0, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
$k_1 \leftarrow \lfloor k/2 \rfloor$
$a_1 \leftarrow a \bmod 2^{2k_1+2}, \quad b_1 \leftarrow b \bmod 2^{2k_1+2}$
$s_1, j_1, R \leftarrow \text{HalfBinaryJacobi}(a_1, b_1, k_1)$
$a' \leftarrow 2^{-2j_1}(R_{1,1}a + R_{1,2}b), \quad b' \leftarrow 2^{-2j_1}(R_{2,1}a + R_{2,2}b)$
$j_0 \leftarrow \nu(b')$
if $j_0 + j_1 > k$ **then**
  Return s_1, j_1, R
$s_0 \leftarrow j_0(a'^2 - 1)/8 \bmod 2$
$q, r \leftarrow \text{BinaryDividePos}(a', b')$
$b'' \leftarrow b'/2^{j_0}$
if $(j_0, q) = (1, 3)$ **then**
  $d \leftarrow a' - b''$
  $m \leftarrow \min(\nu(d) \text{ div } 2, k - j_1)$
  $c \leftarrow (d - (-1)^m d/4^m)/5$
  $s_0 \leftarrow s_0 + m(a' - 1)/2 \bmod 2$
  $(a_2, b_2) \leftarrow (a' - 4c, 2(b'' + c))$ ▷ harmless iteration
  $Q \leftarrow \begin{pmatrix} (4^m + 4(-1)^m)/5 & 2(4^m - (-1)^m)/5 \\ 2(4^m - (-1)^m)/5 & (4^{m+1} + (-1)^m)/5 \end{pmatrix}$
else
  $s_0 \leftarrow s_0 + (a' - 1)(b'' - 1)/4 \bmod 2$
  $(a_2, b_2) \leftarrow (b'', r/2^{j_0})$ ▷ good or bad iteration
  $Q \leftarrow \begin{pmatrix} 0 & 2^{j_0} \\ 2^{j_0} & q \end{pmatrix}$
  $m \leftarrow j_0$
$s_0 \leftarrow s_0 + j_0(a_2^2 - 1)/8 \bmod 2$
$k_2 \leftarrow k - (m + j_1)$
$s_2, j_2, S \leftarrow \text{HalfBinaryJacobi}(a_2 \bmod 2^{2k_2+2}, b_2 \bmod 2^{2k_2+2}, k_2)$
Return $(s_0 + s_1 + s_2) \bmod 2,\ j_1 + j_2 + m,\ S \times Q \times R$

Theorem 3. *Let a, b, k be the inputs of Algorithm HalfBinaryJacobi, and s, j, R the corresponding outputs. If $\left(\begin{smallmatrix} c \\ d \end{smallmatrix}\right) = 2^{-2j} R \left(\begin{smallmatrix} a \\ b \end{smallmatrix}\right)$, then:*

$$(b|a) = (-1)^s (d|c) \quad \textit{and} \quad \nu(2^j c) \le k < \nu(2^j d).$$

Proof (outline). We prove the theorem by induction on the parameter k. The key ingredient is that if we reduce $a, b \bmod 2^{2k_1+1}$ in step 4, then the GB sequence of a_1, b_1 matches that of a, b, for the terms computed by the recursive call at step 5. This is a consequence of [13, Lemma 7] (which also holds for binary division with positive quotient). It follows that in all the binary divisions with inputs a_i, b_i in that recursive call, a_i and $b_i/2^{j_i}$ match modulo 2^{j_i+1} the corresponding values that would be obtained from the full inputs a, b (otherwise the corresponding binary quotient q_i would be wrong). Since here we reduce $a, b \bmod 2^{2k_1+2}$ instead of mod 2^{2k_1+1}, a_i and $b_i/2^{j_i}$ now match modulo 2^{j_i+2} — instead of modulo 2^{j_i+1} — the values that would be obtained from the full inputs a, b, where $2^{j_i+2} \ge 8$ since $j_i \ge 1$.

At step 10, s_0 depends only on $j_0 \bmod 2$ and $a' \bmod 8$, at step 17 it depends on $m \bmod 2$ and $a' \bmod 4$, and at step 21 on $a' \bmod 4$ and $b'' \bmod 4$. Since a' and b'' at step 21 correspond to some a_i and $b_i/2^{j_i}$, it follows that a' and b'' agree mod 8 with the values that would be computed from the full inputs, and thus the correction s_0 is correct. This proves by induction that $(b|a) = (-1)^s (d|c)$.

Now we prove that $\nu(2^j c) \le k < \nu(2^j d)$. If there is no harmless iteration, it is a consequence of the proof of Theorem 1 in [13]. In case there is a harmless iteration, first assume that $m = \nu(d) \operatorname{div} 2$ at step 15. The new values a_2, b_2 at step 18 correspond to m successive ugly iterations, which yield $j = j_1 + m \le k$. Thus $\nu(2^j a_2) \le k$: we did not go too far, and since we are computing the same sequence of quotients as Algorithm QuadraticBinaryJacobi, the result follows. Now if $k - j_1 < \nu(d) \operatorname{div} 2$, we would go too far if we performed $\nu(d) \operatorname{div} 2$ ugly iterations, since it would give $j_0 := \nu(d) \operatorname{div} 2 > k - j_1$, thus $j := j_1 + j_0 > k$, and $\nu(2^j a_2)$ would exceed k. This is the reason why we "cut" the harmless iteration at $m = k - j_1$ (step 15). The other invariants are unchanged. □

Finally we can present our $O(M(n)\log n)$ Algorithm FastBinaryJacobi, which computes the Jacobi symbol by calling Algorithm HalfBinaryJacobi. The general structure is similar to that described in [8] for several asymptotically fast GCD algorithms.

Daireaux, Maume-Deschamps and Vallée [5] prove that, for the positive binary division, the average increase of the most significant bits is 0.65 bits/iteration (which partly cancels an average decrease of two least significant bits per iteration); compare this with only 0.05 bits/iteration on average for the centered division.[4]

[4] We have computed more accurate values of these constants: 0.651993 and 0.048857 respectively.

Algorithm 3.2. FastBinaryJacobi

Input: $a, b \in \mathbb{N}$ with $0 = \nu(a) < \nu(b)$
Output: Jacobi symbol $(b|a)$

$s \leftarrow 0, \quad j \leftarrow \nu(b)$
while $2^j a \neq b$ **do**
  $k \leftarrow \max(\nu(b),\ \ell(b) \text{ div } 3)$ ▷ $\ell(b)$ is length of b in bits
  $s', j, R \leftarrow \text{HalfBinaryJacobi}(a, b, k)$
  $s \leftarrow (s + s') \bmod 2$
  $(a, b) \leftarrow 2^{-2j}(R_{1,1}a + R_{1,2}b, R_{2,1}a + R_{2,2}b), \quad j \leftarrow \nu(b)$
if $a = 1$ **then** return $(-1)^s$ **else** return 0

4 Experimental Results

We have implemented the different algorithms in C (using 64-bit integers) and in GMP (using multiple-precision integers), as well as in Maple/Magma (for testing purposes).

For $\max(a, b) < 2^{26}$ the maximum number of iterations of Algorithm CubicBinaryJacobi is 64, with $a = 15548029$ and $b = 66067306$. The number of iterations seems to be $O(n)$ for $a, b < 2^n$: see Table 1. This is plausible because, from heuristic probabilistic arguments, we expect about half of the iterations to be good, and experiments confirm this. For example, if we consider all admissible $a, b < 2^{20}$, the cumulated number of iterations is 3.585×10^{12} for 2^{38} calls, i.e., an average of 13.04 iterations per call (max 48); the cumulated number of good, bad and ugly iterations is 51.78%, 25.47%, and 22.75% respectively. For $a, b < 2^{60}$, a random sample of 10^8 pairs (a, b) gave 42.72 iterations per call (max 89), with 50.54%, 25.14%, and 24.31% for good, bad and ugly respectively. These ratios seem to be converging to the heuristically expected $1/2 = 50\%$, $1/4 = 25\%$, and $1/4 = 25\%$.

When we consider all admissible $a, b < 2^{20}$, the maximum number of iterations of QuadraticBinaryJacobi is 37 when $a = 933531$, $b = 869894$, the cumulated number of iterations is 3.405×10^{12} (12.39 per call), the cumulated number of good, bad and harmless iterations is 54.51%, 26.82%, and 18.67% respectively. For $a, b < 2^{60}$, a random sample of 10^8 pairs (a, b) gave 40.21 iterations per call (max 76), with 53.70%, 26.71%, and 19.59% for good, bad and harmless respectively. These ratios seem to be converging to the heuristically expected $8/15 = 53.33\%$, $4/15 = 26.67\%$, and $1/5 = 20\%$.

We have also compared the time and average number of iterations for huge numbers, using the fast gcd algorithm in GMP, say `gcd` — which implements the algorithm from [8] — and an implementation of the algorithm from [13], say `bgcd`. For inputs of one million 64-bit words, `gcd` takes about 45.8s on a 2.83Ghz Core 2, while `bgcd` takes about 48.3s and 32,800,000 iterations: this is in accordance with the fact proven in [5] that each step of the binary gcd discards on average two least significant bits, and adds on average about 0.05 most significant bits. Our algorithm `bjacobi` (based on Algorithms 3.1–3.2) takes about 83.1s

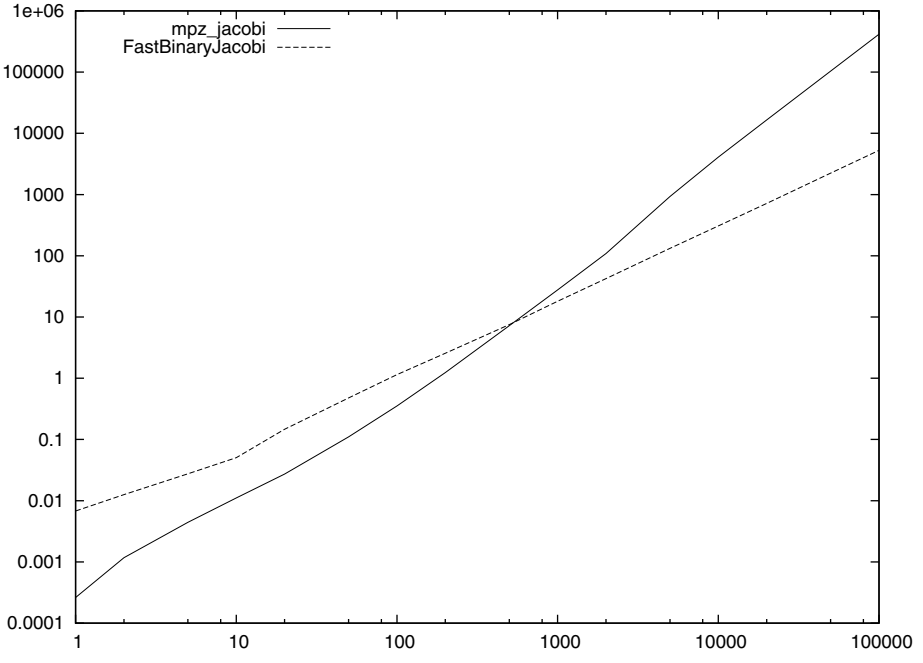


Fig. 1. Comparison of GMP 4.3.1 `mpz_jacobi` routine with our `FastBinaryJacobi` implementation in log-log scale. The x-axis is in 64-bit words, the y-axis in milliseconds on a 2.83Ghz Core 2.

Table 1. Worst cases for CubicBinaryJacobi$(b|a)$, $\max(a,b) < 2^n$

n	iterations	example (a,b)	n	iterations	example (a,b)
5	6	$(7,30)$	22	53	$(2214985, 2781506)$
10	19	$(549,802)$	23	55	$(1383497, 8292658)$
15	34	$(23449, 19250)$	24	58	$(2236963, 12862534)$
20	48	$(656227, 352966)$	25	62	$(28662247, 30847950)$
21	51	$(1596811, 1493782)$	26	64	$(15548029, 66067306)$

and 47,500,000 iterations (for a version with steps 13–20 of Algorithm 3.1 omitted in the basecase routine), which agrees with the theoretical drift of 0.651993 bits per iteration. The break-even point between the $O(n^2)$ implementation of the Jacobi symbol in GMP 4.3.1 and our $O(M(n)\log n)$ implementation is about 535 words, that is about $34,240$ bits or about $10,300$ decimal digits (see Fig. 1).

5 Concluding Remarks

Weilert [15] says: "*We are not able to use a GCD calculation in* $\mathbb{Z}[i]$ *similar to the binary GCD algorithm* $\cdots$ *because we do not get a corresponding quotient*

sequence in an obvious manner". In a sense we filled that gap for the computation of the Jacobi symbol, because we showed how it can be computed using a binary GCD algorithm without the need for a quotient sequence.

We showed how to compute the Jacobi symbol with an asymptotically fast time bound, using such a binary GCD algorithm. Our implementation is faster than a good $O(n^2)$ implementation for numbers with bitsize $n > 35000$. Our subquadratic implementation is available from `http://www.loria.fr/~zimmerma/software/#jacobi`.

Binary division with a centered quotient does not seem to give a subquadratic algorithm; however we can use it with the "cubic" algorithm (which then becomes provably quadratic) since then we control the sign of a, b. For a better quadratic algorithm, we can choose the quotient q so that $abq < 0$, by replacing q by $q - 2^{j+1}$ if necessary: experimentally, this gains on average 2.194231 bits per iteration, compared to 1.951143 for the centered quotient, and 1.348008 for the positive quotient. In comparison, Stein's "binary" algorithm gains on average 1.416488 bits per iteration [4, §7] [7, §4.5.2].

Acknowledgement. We thank Steven Galbraith who asked us about the existence of an $O(M(n) \log n)$ algorithm for the Jacobi symbol, Arnold Schönhage for his comments and a pointer to the work of his former student André Weilert, Damien Stehlé who suggested adapting the binary gcd algorithm, and Marco Bodrato and Niels Möller for testing our implementation. We also thank the two anonymous reviewers, especially the one who actually implemented our new algorithm in Magma! We thank INRIA for its support of the ANC "équipe associée". The first author acknowledges the support of the Australian Research Council.

References

1. Bach, E.: A note on square roots in finite fields. IEEE Trans. on Information Theory 36(6), 1494–1498 (1990)
2. Bach, E., Shallit, J.O.: Algorithmic Number Theory: Efficient Algorithms, vol. 1. MIT Press, Cambridge (1996) (Solution to problem 5.52)
3. Bachmann, P.: Niedere Zahlentheorie, Teubner, Leipzig, vol. 1 (1902); Reprinted by Chelsea, New York (1968)
4. Brent, R.P.: Twenty years' analysis of the binary Euclidean algorithm. In: Davies, J., Roscoe, A.W., Woodcock, J. (eds.) Millennial Perspectives in Computer Science: Proceedings of the 1999 Oxford - Microsoft Symposium in honour of Professor Sir Antony Hoare, Palgrave, New York, pp. 41–53 (2000), `http://wwwmaths.anu.edu.au/~brent/pub/pub183.html`
5. Daireaux, B., Maume-Deschamps, V., Vallée, B.: The Lyapunov tortoise and the dyadic hare. In: Proceedings of the 2005 International Conference on Analysis of Algorithms, DMTCS Proc. AD, pp. 71–94 (2005), `http://www.dmtcs.org/dmtcs-ojs/index.php/proceedings/issue/view/81`
6. Gauss, C.F.: Theorematis fundamentalis in doctrina de residuis quadraticis, demonstrationes et ampliatones novæ. Comm. Soc. Reg. Sci. Gottingensis Rec. 4 (presented February 10, 1817) (1818); Reprinted in Carl Friedrich Gauss Werke, Bd. 2: Höhere Arithmetik, Göttingen, pp. 47–64 (1876)

7. Knuth, D.E.: The Art of Computer Programming. In: Seminumerical Algorithms, 3rd edn., vol. 2, Addison-Wesley, Reading (1997)
8. Möller, N.: On Schönhage's algorithm and subquadratic integer GCD computation. Mathematics of Computation 77(261), 589–607 (2008)
9. Schönhage, A.: Schnelle Berechnung von Kettenbruchentwicklungen. Acta Informatica 1, 139–144 (1971)
10. Schönhage, A.: Personal communication by email (December 2009)
11. Schönhage, A., Grotefeld, A.F.W., Vetter, E.: Fast Algorithms: A Multitape Turing Machine Implementation. BI-Wissenschaftsverlag, Mannheim (1994)
12. Shallit, J., Sorenson, J.: A binary algorithm for the Jacobi symbol. ACM SIGSAM Bulletin 27(1), 4–11 (1993), `http://euclid.butler.edu/~sorenson/papers/binjac.ps`
13. Stehlé, D., Zimmermann, P.: A binary recursive gcd algorithm. In: Buell, D.A. (ed.) ANTS 2004. LNCS, vol. 3076, pp. 411–425. Springer, Heidelberg (2004)
14. Vallée, B.: A unifying framework for the analysis of a class of Euclidean algorithms. In: Gonnet, G.H., Viola, A. (eds.) LATIN 2000. LNCS, vol. 1776, pp. 343–354. Springer, Heidelberg (2000)
15. Weilert, A.: Fast Computation of the Biquadratic Residue Symbol. Journal of Number Theory 96, 133–151 (2002)

New Families of ECM Curves for Cunningham Numbers

Éric Brier[1] and Christophe Clavier[2,3]

[1] Ingenico S.A.
1, rue Claude Chappe, B.P. 346,
07530 Guilherand-Granges, France
eric.brier@ingenico.com
[2] Institut d'Ingénierie Informatique de Limoges (3iL)
43, rue Sainte Anne
F-87000 Limoges
christophe.clavier@3il.fr
[3] Université de Limoges – XLIM
Département de Mathématiques et Informatique
83, rue d'Isle
F-87000 Limoges
christophe.clavier@unilim.fr

Abstract. In this paper we study structures related to torsion of elliptic curves defined over number fields. The aim is to build families of elliptic curves more efficient to help factoring numbers of special form, including numbers from the Cunningham Project. We exhibit a family of curves with rational $\mathbb{Z}/4\mathbb{Z} \times \mathbb{Z}/4\mathbb{Z}$ torsion and positive rank over the field $\mathbb{Q}(\zeta_8)$ and a family of elliptic curves with rational $\mathbb{Z}/6\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z}$ torsion and positive rank over the field $\mathbb{Q}(\zeta_3)$. These families have been used in finding new prime factors for the numbers $2^{972}+1$ and $2^{1048}+1$. Along the way, we classify and give a parameterization of modular curves for some torsion subgroups.

1 Introduction

The Elliptic Curve Method (ECM in short) is a factoring algorithm, whose complexity depends on the size of the smallest prime factor instead of the size of the number to be factored. It can be seen as a variation of the $p-1$ method. The idea is to build an elliptic curve over the ring $\mathbb{Z}/\mathcal{N}\mathbb{Z}$ with a point P on it and to compute the scalar multiplication $\mathcal{M} \cdot P$. Since $\mathcal{N}$ is not a prime, the elliptic curve is not defined over a field. However, computations are done as if we were working on a field and if something fails, this means that a non-trivial factor of $\mathcal{N}$ has been found. The number $\mathcal{M}$ is chosen to be the product of powers of small primes and thus, a prime factor p is found as soon as the order of the elliptic curve reduced modulo p is smooth.

G. Hanrot, F. Morain, and E. Thomé (Eds.): ANTS-IX 2010, LNCS 6197, pp. 96–109, 2010.

Many improvements of the ECM are described in the literature. We will focus on an improvement consisting in choosing the elliptic curve as the reduction modulo $\mathcal{N}$ of an elliptic curve defined over the field $\mathbb{Q}$ with a non-trivial torsion group and positive rank. The torsion group of an elliptic curve is the group of elements of finite order and the rank is the number of generators of the torsion-free part of the group. As soon as small prime factors have been removed from $\mathcal{N}$, the torsion group is preserved in most cases by the modulo $\mathcal{N}$ reduction of the curve, which helps to make the order of the curve smooth. The positive rank is needed to set the starting point P of the algorithm. Possible torsion groups for elliptic curves defined over $\mathbb{Q}$ are in finite number, with maximal order 16. For each possible torsion group, at least a family of elliptic curves with positive rank has been found.

The idea we follow in this paper is to use a number field K for which reduction modulo $\mathcal{N}$ can be made explicit and to build over K an elliptic curve with positive rank and a torsion subgroup as large as possible. Let us give an example : if the number to be factored is of the form $\mathcal{N} = u^2 + 1$, we can make use of the field $K = \mathbb{Q}(i)$ with mapping $i \mapsto u$. The numbers of the Cunningham Project (i.e. numbers of the form $a^m \pm 1$) allow to use m-th roots of unity. It will be interesting to focus on cyclotomic fields or on their subfields. It is important to note that all quadratic extensions of $\mathbb{Q}$ lie in cyclotomic fields.

The paper is organized as follows. Section 2 introduces the necessary notions about modular curves and classifies torsion subgroups that can be of any interest for ECM integer factoring. Section 3 is devoted to construction of parameterized elliptic curves with given torsion subgroup over some cyclotomic extensions of the field of rationals. Section 4 focuses on the search for infinite subfamilies of elliptic curves having nonzero rank, which is mandatory to ECM usage. Section 5 rephrases previous sections results in the context of ECM and gives some instances of new prime factors of Cunningham Project numbers discovered thanks to the work presented here. Finally, section 6 concludes and suggests some research areas to go further.

2 Elliptic Curve Torsion and Modular Curves

An elliptic curve E defined over a number field K turns out to be a commutative group. The Mordell-Weil theorem states that this group is finitely generated and can be written as:

$$E(K) \cong \mathcal{T} \otimes \mathbb{Z}^r$$

where the integer r is called rank and $\mathcal{T}$ is the so called torsion group, which consists in elements of finite order. Furthermore, $\mathcal{T}$ is isomorphic to $\mathbb{Z}/m_1\mathbb{Z} \times \mathbb{Z}/m_2\mathbb{Z}$ with the constraints that m_2 divides m_1 and the m_2-th roots of unity all lie in the field K.

Whereas it is conjectured that the rank is not constrained, the torsion group can take only finitely many different shapes over the field of rationals:

Theorem 1 (Mazur). *The torsion group $\mathcal{T}$ of an elliptic curve defined over the field $\mathbb{Q}$ is isomorphic to one of the following groups:*

$$\mathbb{Z}/m\mathbb{Z} \text{ with } 1 \leq m \leq 10 \text{ or } m = 12$$

$$\mathbb{Z}/2m\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z} \text{ with } 1 \leq m \leq 4$$

This theorem is effective in the sense that for each of these cases, it is possible to give equations of elliptic curves. These parameterizations come from modular curves.

Over the field $\mathbb{C}$ of complex numbers, there is a one-to-one correspondance between isomorphism classes of elliptic curves and the Riemann Surface $X(1)$, which is the quotient $\mathcal{H}^*/SL_2(\mathbb{Z})$, where $\mathcal{H}^*$ is the compactified Poincaré half-plane. For any subgroup Γ of $SL_2(\mathbb{Z})$, the quotient surface $\mathcal{H}^*/\Gamma$ is called a modular curve. Extending notations of [3], we define the following subgroups of $SL_2(\mathbb{Z})$:

$$\Gamma(m) = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in SL_2(\mathbb{Z}), a \equiv d \equiv 1 \bmod m, b \equiv c \equiv 0 \bmod m \right\}$$

$$\Gamma_1(m) = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in SL_2(\mathbb{Z}), a \equiv d \equiv 1 \bmod m, c \equiv 0 \bmod m \right\}$$

$$\Gamma_0(m) = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in SL_2(\mathbb{Z}), c \equiv 0 \bmod m \right\}$$

and the quotients:

$$\begin{aligned} X(m) &= \mathcal{H}^*/\Gamma(m) \\ X_1(m) &= \mathcal{H}^*/\Gamma_1(m) \\ X_0(m) &= \mathcal{H}^*/\Gamma_0(m) \\ X_1(m_1, m_2) &= \mathcal{H}^*/(\Gamma_1(m_1) \cap \Gamma(m_2)) \text{ when } m_2 | m_1 \end{aligned}$$

A point on the surface $X(m)$ corresponds to an elliptic curve together with a basis for its $[m]$-torsion subgroup, up to isomorphism. A point on the surface $X_1(m)$ corresponds, up to isomorphism, to an elliptic curve together with a $[m]$-torsion point. A point on the surface $X_0(m)$ corresponds, up to isomorphism, to an elliptic curve together with a cyclic torsion subgroup of order m. A point on the surface $X(m_1, m_2)$ corresponds to an elliptic curve with a $[m_1]$-torsion point and an independent $[m_2]$-torsion point. Though these notions make use of complex number and analytical tools, the modular curves can also be represented as algebraic curves. An algebraic model of $X_1(m)$ can be found over $\mathbb{Q}$ and the correspondance with an elliptic curve and a $[m]$-torsion point on it is algebraic and defined over $\mathbb{Q}$. The curve $X(m)$ involves the full $[m]$-torsion subgroup and, due to existence of Weil pairing, m-th roots of unity are involved. The rational models and correspondance for $X(m)$ (resp. $X_1(m_1, m_2)$) are defined over the cyclotomic field $\mathbb{Q}(\zeta_m)$ (resp. $\mathbb{Q}(\zeta_{m_2})$). The modular curves associated to torsion

subgroups in Mazur's theorem are genus 0 algebraic curves. This explains why it is possible to give parametric Weierstrass equations.

The Elliptic Curve Method needs a non-bounded number of elliptic curves to compute with. Since algebraic curves of genus greater than 2 have only finitely many rational points over a given number field, we will focus only on torsion structures for which the associated modular curve has genus 0 or 1. Computing the genus of an algebraic curve is not an easy task in the general case but the task is easy with a computer for modular curves of rather small level.

A theorem from Shimura states that the genus of the modular curve $X_1(p)$ for a prime $p \geq 5$ is given by :

$$g = \frac{(p-5)(p-7)}{24}.$$

This implies that the only primes for which the genus of $X_1(p)$ is 0 or 1 are $\{2, 3, 5, 7, 11\}$. When $n|m$, there is a surjective mapping $X_1(m) \to X_1(n)$, and thus the genus of $X_1(m)$ is at least the genus of $X_1(n)$. Computing the genus of $X_1(m)$ for rather small values of m being easy, we can increase the power of these primes until the genus is strictly greater than 1 and we get that the only prime powers for which the genus of $X_1(p^e)$ is 0 or 1 are $\{2, 4, 8, 3, 9, 5, 7, 11\}$. Now, combining this finite set, it is possible to check the following proposition with a finite amount of work:

Proposition 1. *The integers m such that $X_1(m)$ is of genus 0 or 1 are*

$$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 15\}$$

When $m_2|m_1$, there is a surjective mapping $X_1(m_1, m_2) \to X_1(m_1)$, and thus the genus of $X_1(m_1, m_2)$ is at least the genus of $X_1(m_1)$. This implies that if the genus of the modular curve $X_1(m_1, m_2)$ is 0 or 1, the number m_1 is in the list given in proposition 1. Building on this, for any m_1 in this list, we can check if the genus of $X_1(m_1, m_2)$ is 0 or 1 for all divisors m_2 of m_1. The result is given in next proposition.

Proposition 2. *The torsion groups for which the associated modular curve is of genus 0 are:*

$\mathbb{Z}/2\mathbb{Z}$,	$\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$	
$\mathbb{Z}/3\mathbb{Z}$,	$\mathbb{Z}/3\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z}$	
$\mathbb{Z}/4\mathbb{Z}$,	$\mathbb{Z}/4\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$,	$\mathbb{Z}/4\mathbb{Z} \times \mathbb{Z}/4\mathbb{Z}$
$\mathbb{Z}/5\mathbb{Z}$,	$\mathbb{Z}/5\mathbb{Z} \times \mathbb{Z}/5\mathbb{Z}$	
$\mathbb{Z}/6\mathbb{Z}$,	$\mathbb{Z}/6\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$,	$\mathbb{Z}/6\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z}$
$\mathbb{Z}/7\mathbb{Z}$		
$\mathbb{Z}/8\mathbb{Z}$,	$\mathbb{Z}/8\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$	
$\mathbb{Z}/9\mathbb{Z}$		
$\mathbb{Z}/10\mathbb{Z}$		
$\mathbb{Z}/12\mathbb{Z}$		

The torsion groups for which the associated modular curve is of genus 1 are:

$$\begin{array}{l}\mathbb{Z}/6\mathbb{Z} \times \mathbb{Z}/6\mathbb{Z}\\ \mathbb{Z}/8\mathbb{Z} \times \mathbb{Z}/4\mathbb{Z}\\ \mathbb{Z}/9\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z}\\ \mathbb{Z}/10\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}\\ \mathbb{Z}/11\mathbb{Z}\\ \mathbb{Z}/12\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}\\ \mathbb{Z}/14\mathbb{Z}\\ \mathbb{Z}/15\mathbb{Z}\end{array}$$

3 Parameterization of Elliptic Curves with Given Torsion Structure

When the base field is $\mathbb{Q}$, several papers (e.g. [8] and [4]) describe the construction of elliptic curves with prescribed torsion groups. We will study cases that need to work over extensions.

3.1 Construction of $\mathbb{Z}/3\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z}$

To study torsion points, one can use the division polynomials, whose roots are the abscises of torsion points. Since we wish all 3-torsion points to be rational, we start by imposing two rational roots x_1 and x_2 to the polynomial

$$\varphi_3(x) = 3x^4 + 6ax^2 + 12bx - a^2$$

The system $\varphi_3(x_1) = \varphi_3(x_2) = 0$ considered as equations in the variables a and b has roots if and only if $-3x_1x_2$ is a square. A convenient parameterization is

$$\begin{cases} x_1 = 6\xi \\ x_2 = -2\rho^2\xi \end{cases}$$

and the corresponding parameters are

$$\begin{cases} a = -12\xi^2\rho(\rho^2 - 3\rho + 3) \\ b = 2\xi^3(\rho^2 - 3)(\rho^4 - 6\rho^3 + 18\rho^2 - 18\rho + 9) \end{cases}$$

At this stage, we introduce two linear factors in φ_3. The remaining quadratic factor of φ_3 has discriminant equal to $-3(\rho - 1)^2(\rho - 3)^2$. We need -3 to be a square, which is natural since the Weil pairing introduces cubic roots of unity. We thus have x-coordinates of point of order 3 rational.

We now turn on to y-coordinates. Substitutions of x_1 and x_2 in $x^3 + ax + b$ yield $y_1^2 = 2\xi^3(\rho - 3)^2(\rho^2 + 3)^2$ and $y_1^2 = -6\xi^3(\rho - 1)^2(\rho^2 + 3)^2$. To obtain squares, we set $\xi = 2\lambda^2$ and for convenience $\rho = 1 - \tau$. In conclusion, an elliptic curve in short Weierstrass form has rational 3-torsion over $\mathbb{Q}(\zeta_3)$ if and only if its parameters can be written as:

$$\begin{cases} a = 48\lambda^4\ (\tau^3 - 1) \\ b = 16\lambda^6\ (\tau^6 - 20\tau^3 - 8) \end{cases}$$

3.2 Construction of $\mathbb{Z}/6\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z}$

Given as an input the results of previous section, we now have to ensure that $x^3 + ax + b$ has a linear factor to get a point of order 2. In a first step, we set $x = \xi\lambda^2$ to get rid of the homogeneity parameter λ. We consider then $x^3 + ax + b$ as a equation in ξ and τ^3, which is quadratic relatively to the unknown τ^3. The discriminant of this quadratic equation is $-(\xi - 12)^3$. It is natural to set $\xi = 12 - \nu^2$. We now have:

$$x^3 + ax + b = \lambda^6(6\nu^2 + \nu^3 - 4\tau^3 - 32)(6\nu^2 - \nu^3 - 4\tau^3 - 32)$$

Both factors differ only in a sign change for ν. We will keep the first factor, which is a cubic in ν and τ. Since the underlying modular curve has genus 0, this curve must have a singularity. We easily find that the point $(\nu = -4, \tau = 0)$ is singular and to reduce the degree of the curve, we set $\nu = \mu\tau - 4$. After replacement and factorization, we have a degree one equation in τ. To keep consistency in notations and to avoid denominators, we rename μ as $1/\tau$ and modify the scaling factor λ. In conclusion, an elliptic curve in short Weierstrass form has rational 3-torsion and a point of order 2 over $\mathbb{Q}(\zeta_3)$ if and only if its parameters can be written as:

$$\begin{cases} a = -3\lambda^4\ (\tau^{12} - 8\tau^9 + 240\tau^6 - 464\tau^3 + 16) \\ b = -2\lambda^6\ (\tau^{18} - 12\tau^{15} - 480\tau^{12} + 3080\tau^9 - 12072\tau^6 + 4128\tau^3 + 64) \end{cases}$$

3.3 Modular Curve for $\mathbb{Z}/6\mathbb{Z} \times \mathbb{Z}/6\mathbb{Z}$

We know that the modular curve $X(6)$ has genus 1. In this section, we will give a very simple model for this elliptic curve. Let us start with the equation for $\mathbb{Z}/6\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z}$ torsion subgroup. The polynomial $x^3 + ax + b$ has by construction a linear and a quadratic factor. The discriminant of the quadratic factor is $-9(8\tau^3 - 1)^3$. From this we derive the following model:

$$X(6) : s^2 = t^3 + 1$$

3.4 Modular Curve for $\mathbb{Z}/9\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z}$

We start from parameterization of curves with full 3-torsion. One can note that the parameter is involved only to the third power, we thus note $\sigma = \tau^3$ and will work in a first stage only with σ.

We introduce the polynomial χ_9 whose roots are the sums of x-coordinates of points in cyclic subgroups of order 9 and whose degree is 12:

$$\chi_9(z) = z^{12} + 792az^{10} + 47520bz^9 + ... - 3543478272a^6$$

We can de-homogenize this polynomial by setting $\lambda = 1$ and, since a and b are polynomials in σ, we get a polynomial equation in z and σ having a quadratic factor in σ. This factor has a root iff $z - 48$ is six times a square. We set

$z = 6\zeta^2 + 48$ and get $\sigma = (\zeta^3 + 6\zeta^2 + 12\zeta + 72)/8$. We can now factor the division polynomial $\varphi_9(x)$ and obtain an equation of degree 3 in x and 6 in ζ. The solution $x = 12$ and $\zeta = -2$ being a singularity, we set $x = 12 + \xi(\zeta + 2)^2$ and obtain the relation

$$\zeta = -2\frac{\xi^3 + 3\xi^2 - 6\xi + 1}{\xi^3 - 3\xi^2 + 1}$$

We are guaranteed that a point of order 9 has rational x-coordinate, its happens that the y-coordinate is also rational. It is now time to remember that σ must be a cube. Elliptic curve with torsion group of type $\mathbb{Z}/9\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z}$ have same parameters as for $\mathbb{Z}/3\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z}$, provided that

$$\tau^3 = \frac{8(\xi^2 - \xi + 1)^3(\xi^3 - 6\xi^2 + 3\xi + 1)}{(\xi^3 - 3\xi^2 + 1)^3}$$

Some algebraic manipulations turn the equation $\sigma^3 = \xi^3 - 6\xi^2 + 3\xi + 1$ into the elliptic model:

$$X_1(9, 3) : s^2 = t^3 + 16$$

3.5 Construction of $\mathbb{Z}/4\mathbb{Z} \times \mathbb{Z}/4\mathbb{Z}$

In short Weierstrass form, points of order 2 are points whose y-coordinate is 0. It follows that the general form of curve with $\mathbb{Z}/2\mathbb{Z}$ torsion is:

$$y^2 = (x - u)(x^2 + ux + v)$$

For the same reasons the general form of curve with $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$ torsion is:

$$y^2 = (x - u)(x - v)(x + u + v)$$

On this elliptic curve, a point $P = (x, y)$ can be written $P = 2Q$ iff the numbers $x - u$, $x - v$ and $x + u + v$ are squares, see [2, Theorem 4.2 page 85]. Thus, if we require that all 4-torsion are rational, all 2-torsion points must be doubles and we ask for 0, $\pm(u - v)$, $\pm(2u + v)$ and $\pm(u + 2v)$ being squares. One can note that -1 has to be a square, which is not a surprise: if 4-torsion is rational, the Weil pairing will produce fourth roots of unity, i.e. square roots of -1.

We first impose $2u + v$ and $2v + u$ to be squares. To do so, we invert the system:

$$\begin{cases} 2u + v = r^2 \\ u + 2v = s^2 \end{cases} \iff \begin{cases} u = (2r^2 - s^2)/3 \\ v = (2s^2 - r^2)/3 \end{cases}$$

Then, it remains to ensure that $u - v$ is also a square. The factorization of $u - v$ is $(r - s)(r + s)$. It is convenient to write $r = \mu + \nu$ and $s = \mu - \nu$. We get $u - v = 4\mu\nu$, which must be a square. We can set $\mu = \tau^2\nu$. Last, to get rid of denominators, we set $\nu = 3\lambda$. In conclusion, an elliptic curve in short Weierstrass form has rational 4-torsion over $\mathbb{Q}(\zeta_4)$ if and only if its parameters can be written as:

$$\begin{cases} a = -27\lambda^4\,(\tau^8 + 14\tau^4 + 1) \\ b = \quad 54\lambda^6\,(\tau^{12} - 33\tau^8 - 33\tau^4 + 1) \end{cases}$$

3.6 Modular Curve for $\mathbb{Z}/8\mathbb{Z} \times \mathbb{Z}/4\mathbb{Z}$

To obtain a point of order 8, one of the points of order 4 must be expressed as the doubling of a rational point. We take for instance one of the points with $x = 3\tau^4 - 15$. Differences with x-coordinates of 2-torsion points must be squares, these differences factor as:

$$\begin{aligned} -18\ (\tau^2 + 1) \\ 18\ (\tau^2 - 1) \\ 9\ (\tau^4 - 1) \end{aligned}$$

We can easily impose the second expression to be a square by setting

$$\tau = (\kappa^2 + 2)/(\kappa^2 - 2)$$

Then, the two other expressions are squares iff $\kappa^4 + 4$ is a square. In the equation $\sigma^2 = \kappa^4 + 4$, we apply the change of variables $\sigma = s^2/t^2 - 2t$ and $\kappa = -s/t$ and get the model:

$$X_1(8, 4) : s^2 = t^3 - t$$

3.7 Construction of $\mathbb{Z}/5\mathbb{Z} \times \mathbb{Z}/5\mathbb{Z}$

To reach full rational 5-torsion, we begin with two rational cyclic subgroups of order 5. Let χ_5 denote the polynomial, whose roots are the sums of x-coordinates of points over the 6 cyclic subgroups of order 5:

$$\chi_5(z) = z^6 + 20az^4 + 160bz^3 - 80a^2z^2 - 128abz - 80b^2$$

We note z_1 and z_2 two roots of χ_5 and to take benefit of symmetry use the transformation $z_1 = u + v$ and $z_2 = u - v$. We consider the system $\chi_5(z_1) = \chi_5(z_2) = 0$ as equations in a and b and eliminate the unknown a, obtaining a quartic in b with parameters u and v. It is then convenient to set $b = (u^2 - v^2)\beta$ to reduce degrees in u and v. This quartic presents a strong singularity when $v = 0$ and $\beta = u/4$, which leads us to set $\beta = (u/4 + \gamma v/8)$. The result is still a quartic in γ but the degree in v fell down to 2 and the discriminant of this quadratic equation in v is a square iff $9 - 5\gamma^2$ is five times a square. We use conic parameterization techniques to obtain:

$$\gamma = \frac{6(\mu^2 + \mu - 1)}{5(\mu^2 + 1)}$$

Now v can be expressed as the product of u and a rational function of μ. We unroll substitutions to get the value of b and come back to equations $\chi_5(z_1) = \chi_5(z_2) = 0$. They have a common linear factor in a and we now have values for a and b.

Knowing that χ_5 has two rational roots, we can strengthen our wishes and factor the division polynomial φ_5. No surprise that we get two quadratic factors, whose discriminants are squares if and only if $\mu^2 + 1$ and $5(\mu^2 + 1)$ are squares.

We remember that we are working over the field of fifth roots of unity, in which 5 is a square. We just have to set

$$\mu = \frac{2\tau}{\tau^2 - 1}$$

Now that x-coordinates for 5-torsion points are rational, we choose the value of homogeneity parameter u to have y-coordinates rationals

$$u = -6\lambda^2(\tau^2+1)(\tau^4 - 2\tau^3 - 6\tau^2 + 2\tau + 1)(2\tau^4 + \tau^3 + 3\tau^2 - \tau + 2)$$

In conclusion, an elliptic curve in short Weierstrass form has full rational 5-torsion over $\mathbb{Q}(\zeta_5)$ if and only if its parameters can be written as:

$$\begin{cases} a = -27\lambda^4\ (\tau^{20} + 228\tau^{15} + 494\tau^{10} - 228\tau^5 + 1) \\ b = \quad 54\lambda^6\ (\tau^{30} - 522\tau^{25} - 10005\tau^{20} - 10005\tau^{10} + 522\tau^5 + 1) \end{cases}$$

4 Construction of Elliptic Curve with Large Prescribed Torsion and Positive Rank

4.1 Description of the Method

For an elliptic curve being useful for the Elliptic Curve Method, its rank has to be non-zero. This means that we still have to produce sub-families of curves with an extra rational point. When the modular curve is of genus 1, we did not find any method because we are lacking of freedom on the parameters. This section is devoted to the method we use to produce sub-families with positive rank in the case of a parameterization by $\mathbb{P}_1(K)$.

In this case, the parameters a and b are, up to the scaling factor λ, polynomials in $K(\tau)$ and we can take x to be also a polynomial $x = \lambda^2\xi(\tau)$. Then $x^3 + ax + b$ becomes itself λ^6 times a polynomial. The polynomial ξ being fixed, we can look for values of τ, which turns $x^3 + ax + b$ into a square. This approach is equivalent to looking for rational points on hyperelliptic curves of rather high genus and will yield only finitely many curves. Our method consists in choosing the polynomial ξ in such a way that $x^3 + ax + b$ contains as much as possible of square factors.

We note $a = \lambda^4\alpha(\tau)$, $b = \lambda^6\beta(\tau)$ and $\sigma(\tau) = \xi(\tau)^3 + \alpha(\tau)\xi(\tau) + \beta(\tau)$. For readability, we will omit the parameter τ for polynomial and all derivatives will be taken relatively to τ. We wish to have square factors, i.e. relations of type $\sigma \equiv 0 \bmod (\tau - \tau_0)^2$. In most cases, this relation imposes to define ξ modulo $(\tau - \tau_0)^2$. Since increasing the degree of ξ will in the end increase the degree of σ, we try to obtain this relation with a constraint only on ξ modulo $(\tau - \tau_0)$. Let us compute derivatives:

$$\sigma' = (3\xi^2 + \alpha)\xi' + (\xi\alpha' + \beta')$$

To avoid constraints modulo $(\tau - \tau_0)^2$, we must keep freedom on ξ', which leads to $3\xi^2 + \alpha = 0$. Combining this relation with $\xi^3 + \alpha\xi + \beta$, we get the criterion

$\Delta = 4\alpha^3 + 27\beta^2 = 0$ and the value for $\xi = -3\beta/2\alpha$. Now, we have to check that $\xi\alpha' + \beta' = 0$. Under the previous conditions, this is equivalent to $\Delta' = 0$. The values τ_0 that will be of interest will thus be multiple roots of the discriminant Δ. To have a maximum number of degrees of freedom, for each of these roots we try to impose conditions on ξ modulo $(\tau - \tau_0)^e$ and check whether we get $\sigma \equiv 0 \bmod (\tau - \tau_0)^{2e}$.

The last step is to combine multiple roots using the Chinese Remainder Theorem in $K[\tau]$. For each possible τ_0, we fix ξ modulo some power $(\tau - \tau_0)^e$, the exponent e being less than the maximum "useful" exponent. We obtain candidates for ξ and for each candidate we factor σ. Since we wish to have σ being a square, we write $\sigma = \sigma_1^2\sigma_0$ with σ_0 square-free. If ξ does not correspond to torsion points and if the degree of σ_0 is less than 5, we can parameterize by a curve of genus 0 or 1. If the auxiliary curve is of genus one (i.e. an elliptic curve) and if we can exhibit a point, we build an infinite family of elliptic curves with given torsion and rank at least one.

4.2 Results for $\mathbb{Z}/4\mathbb{Z} \times \mathbb{Z}/4\mathbb{Z}$

Taking the values for a and b given in section 3.5, we first factor the discriminant

$$\Delta = -2^4 3^{12} \tau^4 (\tau - 1)^4 (\tau + 1)^4 (\tau^2 + 1)^4$$

The values of interest for τ are $\{0, 1, -1, \iota, -\iota\}$. We then check that each of them can be used up to the second power. The number of candidates we can generate for ξ is 3^5. To simplify exploration of all these candidates, we compute once for all a polynomial Ξ that satisfies all modular conditions

$$\Xi = 9\tau^8 - 24\tau^4 + 3$$

take its remainder modulo the polynomial

$$\tau^{e_0}(\tau - 1)^{e_1}(\tau + 1)^{e_{-1}}(\tau - \iota)^{e_\iota}(\tau - \iota)^{e_{-\iota}}$$

We get values σ_0 of degree 0 that are of no interest since they correspond to torsion points. We get no values of degree 1, 16 different values of degree 2, 32 of degree 3 and 62 of degree 4. The simplest value of σ_0 is $3^6(\tau^2 - 3)$, which corresponds to $\xi = 9\tau^6 - 15\tau^4 - 9\tau^2 + 3$. To turn σ_0 into a square, one can set

$$\tau = \frac{\nu^2 + 3}{2\nu} \text{ and } \lambda = 8\nu^3$$

Unrolling substitutions, we have

$$\begin{cases} a = -432\nu^4\,(\nu^{16} + 24\nu^{14} + 476\nu^{12} + 4200\nu^{10} + 18022\nu^8 \\ \qquad\qquad +37800\nu^6 + 38556\nu^4 + 17496\nu^2 + 6561) \\ \\ b = \;3456\nu^6\,(\nu^{24} + 36\nu^{22} + 66\nu^{20} - 6732\nu^{18} - 101409\nu^{16} - 707256\nu^{14} \\ \qquad\qquad -2772260\nu^{12} - 6365304\nu^{10} - 8214129\nu^8 - 4907628\nu^6 \\ \qquad\qquad +433026\nu^4 + 2125764\nu^2 + 531441) \end{cases}$$

The point of infinite order is given by

$$\begin{cases} x = \ 3\ (3\nu^{12} + 34\nu^{10} + 117\nu^8 + 316\nu^6 + 1053\nu^4 + 2754\nu^2 + 2187) \\ \\ y = 27\ (\nu^2 - 3)(\nu^2 + 1)(\nu^2 + 9)(\nu^6 + 5\nu^4 + 15\nu^2 + 27)^2 \end{cases}$$

The choice of parameters giving such a torsion group when -1 is a square has also been studied to speed-up factorisation in [9].

4.3 Results for $\mathbb{Z}/6\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z}$

Following the same steps, we start from formulae given in section 3.2 and factor the discriminant:

$$\Delta = -2^8 3^6 \tau^3 (\tau^3 + 1)^6 (\tau^3 - 8)^3$$

The values of interest for τ_0 are $\{-1, -\zeta_3, -\zeta_3^2, 0, 2, 2\zeta_3, 2\zeta_3^2\}$. Only the first 3 values can be used up to the second power, the four last ones being of interest only to the first power. The solution for all modular constraints is

$$\Xi = 2\tau^9 - 9\tau^6 - 42\tau^3 - 4$$

The 432 possible candidates for ξ yield 32 cases where σ_0 is of degree 4. Among them, one of the simplest corresponds to $\sigma_0 = -3\tau(5\tau^3 + 32)$ with $\xi = -13\tau^6 - 44\tau^3 - 4$. The elliptic curve $\rho^2 = -3\tau(5\tau^3 + 32)$ has nonzero rank over $\mathbb{Q}$, a point of infinite order being $(-1, 9)$. The points of this auxiliary elliptic curve parameterize an infinite family of elliptic curve having nonzero rank over $\mathbb{Q}$ and a torsion group containing $\mathbb{Z}/6\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z}$ over $\mathbb{Q}(\zeta_3)$

4.4 Results for $\mathbb{Z}/5\mathbb{Z} \times \mathbb{Z}/5\mathbb{Z}$

Once again, we start by factoring:

$$\Delta = -2^8 3^{12} \tau^5 (\tau^{10} - 11\tau^5 - 1)^5$$

The eleven values of interest for τ_0 can all be used up to the second power and the polynomial compatible with all constraints is

$$\Xi = -\frac{1}{25}(252\tau^{20} - 5508\tau^{15} + 29019\tau^{10} + 7686\tau^5 + 75)$$

Unfortunately, the 3^{11} possible candidates for ξ all give σ_0 polynomials of degree five or more, except for those corresponding to 5-torsion points.

We also noticed that α is of degree 20 and β of degree 30. If we restrict ourselves to polynomials of degree 10 for ξ, the degree of σ will not exceed 30. In the case the leading coefficient of ξ is -3, the degree of σ falls down to 28. One can see this as using the value $\tau_0 = \infty$. This is compatible with 10 modular constraints and we also tried the 24068 candidates built this way, with no success.

Remark: To speed up computations and avoid to compute in a quartic extension of $\mathbb{Q}$, we instead performed this computations in the field $\mathbb{F}_{32621}$, which contains fifth roots of unity. For sure, if a solution had been found, we would have needed to perform actual computations in $\mathbb{Q}(\zeta_5)$.

4.5 Half Way to $\mathbb{Z}/8\mathbb{Z} \times \mathbb{Z}/4\mathbb{Z}$

As the modular curve $X_1(8,4)$ is of genus one, we lack freedom on curve parameters to ensure in addition a non zero rank. We will try to cover a part of the path from $X_1(4,4)$ to $X_1(8,4)$. We elaborate on the results of section 4.2 and will use the same parameterization, with a dedicated choice of values for the parameter ν.

As in section 3.5, we use the characterization of points P that can be written $P = [2]Q$ with Q a point with rational coordinates. Among the twelve points of order 4, there is one that needs the quantities $\nu(\nu^2+3)$ and $(\nu^2+1)(\nu^2+9)$ to be squares. Would both be squares, we would get a point of order 8. We limit ourselves to the first condition only.

At this stage, it is quite natural to consider the elliptic curve

$$\mu^2 = \nu(\nu^2+3).$$

The rank of this auxiliary curve over the field $\mathbb{Q}$ is one and an infinite subgroup is generated by the point $P_0 = (1,2)$. Each of the points $[k]P_0$ with $k \in \mathbb{N}$ yields a value of ν to be plugged into the formulæ of section 4.2.

We thus get a infinite family of elliptic curves with nonzero rank, $\mathbb{Z}/4\mathbb{Z}\times\mathbb{Z}/4\mathbb{Z}$ torsion over $\mathbb{Q}(\zeta_4)$, and better chances to get $\mathbb{Z}/8\mathbb{Z}\times\mathbb{Z}/4\mathbb{Z}$ torsion over the same number field.

5 Application to Factoring

One can see an ECM implementation as a black box taking as inputs:

A number $\mathcal{N}$ to be factored
Elliptic curve paramaters a and b
Coordinates of a point P on the curve modulo $\mathcal{N}$

and computing the scalar multiplication $\mathcal{M} \cdot P$ on this curve for a smooth large integer $\mathcal{M}$, expecting the result being at infinity for some prime factor of $\mathcal{N}$. In most implementations, projective coordinates are used and if $\mathcal{M} \cdot P$ is at infinity modulo a prime factor, this factor can be retrieved by a simple GCD between the number to be factored and the third coordinate. For full explanations on implementations and improvements of ECM, see [10], [1] and [7].

For the torsion groups $\mathbb{Z}/4\mathbb{Z}\times\mathbb{Z}/4\mathbb{Z}$ and $\mathbb{Z}/6\mathbb{Z}\times\mathbb{Z}/3\mathbb{Z}$, we found curves having parameters and a point of infinite order defined over $\mathbb{Q}$. These curves can be used for any number to be factored $\mathcal{N}$. However, the benefit of torsion is attained only when one knowns that suitable roots of unity exist in the finite fields defined by prime factors of $\mathcal{N}$. For order 16 torsion groups, numbers of the form $a^{4n}-b^{4n}$ or $a^{2n}+b^{2n}$ satisfy these conditions. The torsion group of order 18 can be used on numbers of the form $a^{3n} \pm b^{3n}$. The suggested extension towards torsion group of order 32 can be used for numbers of the form $a^{8n} - b^{8n}$ or $a^{4n} + b^{4n}$.

To implement results of section 4.2, the parameter ν can be chosen at random or iteratively on integers. To implement results of section 4.3, things are slightly

less simple, since an auxiliary elliptic curve has to be used. In this case one has to select an integer k randomly or in sequence, compute a scalar multiplication on the auxiliary elliptic curve to get the inputs of ECM.

We adapted our ECM implementation in order to use these new families of elliptic curves. Making use of results on $\mathbb{Z}/4\mathbb{Z} \times \mathbb{Z}/4\mathbb{Z}$ torsion we found several factors of Cunningham numbers. Among them, one can mention the larger one:

$$5546025484206613872527377154544456740766039233 | 2^{1048} + 1$$

We won't give here full details of the factorization since they do not correspond to notations of these paper, this factor having been found in an early stage of development of this paper.

We also implemented the variant with $\mathbb{Z}/6\mathbb{Z}\times\mathbb{Z}/3\mathbb{Z}$ torsion. Among the factors we found, the larger to mention is

$$15812147735432893557636948081842050625168 17 | 2^{972} + 1$$

This factor has been discovered using the input parameters:

$$\begin{cases} a &= 29826081614523423723477944537088124780779 \mod p \\ b &= 12998080963266534977610607798174418536 3149 \mod p \\ x &= 47994679345592513140857304243216026498 8537 \mod p \\ y &= 34122396666617422996194223430401896860 5682 \mod p \end{cases}$$

The order of the curve modulo p factors as:

$$\begin{aligned} \#\mathcal{E}(\mathbb{F}_p) = 2 \times 3^2 \times 29 \times 241 \times 691 \times 5279 \times 20353 \times 252589 \\ \times 1489097 \times 2258261 \times 199312079 \end{aligned}$$

6 Conclusion

We exhibited two torsion groups, that can be used for ECM factoring, of orders 16 and 18. Classical implementations make use of the torsion group $\mathbb{Z}/8\mathbb{Z}\times\mathbb{Z}/2\mathbb{Z}$ that can be used for all numbers but of slightly smaller order. It would be really interesting to have a precise analysis of complexity improvements obtained by using torsion groups, as well as partial construction of torsion structure as in section 4.5.

In the case of torsion group of order 25, we did not succeed in constructing elliptic curves having nonzero rank. This by no way means that no such curves exist. Solving this issue would result in specific implementations for numbers of the form $a^{5n} \pm 1$ with the larger available torsion group.

Some torsion groups correspond to a modular curve of genus one. The obstruction in using them for ECM is the lack of freedom to build curve with nonzero rank: to build a curve with this torsion, one only have to select a multiple of a generator on this modular curve. Several approaches could improve the situation: being able to construct a large number of curves with nonzero rank

by using rank computation software or being able to construct a point on the curve modulo $\mathcal{N}$ after the curve has been generated.

Last, while infinite families of curves are needed for ECM factoring of integers, individual curves providing large torsion groups over some number fields could be used during the sieving phase of the special number field sieve (see [5] and [6]). Though further research is needed to hunt for interesting individual curves, we quote one preliminary result: the choice of $\nu = 1$ in section 4.2 ensures a torsion subgroup of order 32 over the fields $\mathbb{Q}(\zeta_{24})$ and $\mathbb{Q}(\zeta_{40})$ and of order 64 over the field $\mathbb{Q}(\zeta_{120})$.

References

1. Cohen, H.: A Course in Computational Algebraic Number Theory. Graduate Texts in Mathematics, vol. 138. Springer, Heidelberg (1991)
2. Knapp, A.W.: Elliptic Curves. Princeton University Press, Princeton (1992)
3. Koblitz, N.: Introduction to Elliptic Curves and Modular Forms. Graduate Texts in Mathematics, vol. 97. Springer, Heidelberg (1993)
4. Kubert, D.S.: Universal bounds on the torsion of elliptic curves. In: Proceedings of the London Mathematical Society, pp. 193–237 (1976)
5. Lenstra, A.K., Lenstra, H.W.: The Development of the Number Field Sieve. LNM, vol. 1554. Springer, Heidelberg (1993)
6. Lenstra, A.K., Lenstra, H.W., Manasse, M.S., Pollard, J.M.: The Factorization of the Ninth Fermat Number. In: Mathematics of Computation, vol. 61. American Mathematical Society, Providence (1993)
7. Lenstra, H.W.: Factoring integers with elliptic curves. Annals of Mathematics 126, 649–673 (1987)
8. Mazur, B.: Rational isogenies of prime degree. Invent. Math., 129–162 (1978)
9. Montgomery, P.L.: Speeding the pollard and elliptic curve methods of factorization. Mathematics of Computation 48, 243–264 (1987)
10. Zimmermann, P., Dodson, B.: Twenty Years of ECM. In: Hess, F., Pauli, S., Pohst, M. (eds.) ANTS 2006. LNCS, vol. 4076, pp. 525–542. Springer, Heidelberg (2006)

Visualizing Elements of Sha[3] in Genus 2 Jacobians

Nils Bruin and Sander R. Dahmen*

Department of Mathematics, Simon Fraser University, Burnaby, BC, Canada
nbruin@sfu.ca, sdahmen@irmacs.sfu.ca

Abstract. Mazur proved that any element ξ of order three in the Shafarevich-Tate group of an elliptic curve E over a number field k can be made visible in an abelian surface A in the sense that ξ lies in the kernel of the natural homomorphism between the cohomology groups $H^1(\mathrm{Gal}(\overline{k}/k), E) \to H^1(\mathrm{Gal}(\overline{k}/k), A)$. However, the abelian surface in Mazur's construction is almost never a jacobian of a genus 2 curve. In this paper we show that any element of order three in the Shafarevich-Tate group of an elliptic curve over a number field can be visualized in the jacobians of a genus 2 curve. Moreover, we describe how to get explicit models of the genus 2 curves involved.

1 Introduction

Let E be an elliptic curve over a field k with separable closure $\overline{k}$. We write $H^1(k, E[3]) := H^1(\mathrm{Gal}(\overline{k}/k), E[3](\overline{k}))$ for the first galois cohomology group taking values in the 3-torsion of E (the notation $H^i(k, A)$ is used similarly for other group schemes A/k later in this paper). We are primarily concerned with the question which $\delta \in H^1(k, E[3])$ are *visible* in the jacobian of a genus 2 curve. Mazur defines *visibility* in the following way. Let $0 \to E \to A \to B \to 0$ be a short exact sequence of abelian varieties over k. By taking galois cohomology, we obtain the exact sequence

$$A(k) \longrightarrow B(k) \longrightarrow H^1(k, E) \xrightarrow{\phi} H^1(k, A)\ . \tag{1.1}$$

Elements of the kernel of ϕ are said to be *visible* in A. Mazur chose this term because a model of the principal homogeneous space corresponding to an element $\xi \in H^1(k, E)$ that is visible in A can be obtained as a fiber of A over a point in $B(k)$ (this can readily be seen from (1.1)). By extension, we say that $\delta \in H^1(k, E[n])$ is visible in A if the image of δ under the natural homomorphism $H^1(k, E[n]) \to H^1(k, E)$ is visible in A.

Let us restrict to the case that k is a number field for the rest of this section. Inspired by some surprising experimental data [4], Mazur [5] proved, that for any element ξ in the Shafarevich-Tate group $Ш(E/k)$ of order three, there exists an abelian variety A over k such that ξ is visible in A. The abelian variety that

* Research of both authors supported by NSERC.

G. Hanrot, F. Morain, and E. Thomé (Eds.): ANTS-IX 2010, LNCS 6197, pp. 110–125, 2010.

Mazur constructs is almost never principally polarizable over $\overline{k}$ and hence is almost never a jacobian of a genus 2 curve. In the present paper, we show that any element from $Ш(E/k)[3]$ is in fact visible in the jacobian of a genus 2 curve. Moreover, we describe how to get an explicit model of such a genus 2 curve.

2 Torsors and Theta Groups

Throughout this section let $n > 1$ be an integer, let k be a perfect field of characteristic not dividing n and let E denote an elliptic curve over k. In [2], many equivalent interpretations are given for the group $H^1(k, E[n])$. For our purposes, we need two classes of objects. The first is most closely related with descent in general and our question in particular. We consider E-torsors under $E[n](\overline{k})$ and, following [2], call them n-coverings.

Definition 1. *An n-covering $\pi : C \to E$ of an elliptic curve E is an unramified covering over k that is galois and irreducible over $\overline{k}$, with $\mathrm{Aut}_{\overline{k}}(C/E) \simeq E[n](\overline{k})$. Two n-coverings $\pi_1 : C_1 \to E$, $\pi_2 : C_2 \to E$ are called isomorphic if there exists a k-morphism $\phi : C_1 \to C_2$ such that $\pi_1 = \pi_2 \circ \phi$.*

Over $\overline{k}$, all n-coverings are isomorphic to the trivial n-covering, the multiplication-by-n map $[n] : E \to E$.

Proposition 1 ([2, Proposition 1.14])**.** *The k-isomorphism classes of n-coverings of E are classified by $H^1(k, E[n])$.*

For $\delta \in H^1(k, E[n])$ we denote by C_δ the curve in the covering $C_\delta \to E$ corresponding to δ. We remark that $\delta \in H^1(k, E[n])$ has trivial image in $H^1(k, E)$ if and only if C_δ has a k-rational point.

We write O for the identity on E. The complete linear system $|n \cdot O|$ determines a morphism $E \to \mathbb{P}^{n-1}$, where the translation action of $E[n]$ extends to a linear action on $\mathbb{P}^{n-1}$. This gives a projective representation $E[n] \to \mathrm{PGL}_n$. The lift of this representation to GL_n gives rise to a group Θ_E, which fits in the following diagram.

$$\begin{array}{ccccccccc} 1 & \longrightarrow & \mathbb{G}_m & \xrightarrow{\alpha_E} & \Theta_E & \xrightarrow{\beta_E} & E[n] & \longrightarrow & 1 \\ & & \| & & \downarrow & & \downarrow & & \\ 1 & \longrightarrow & \mathbb{G}_m & \longrightarrow & \mathrm{GL}_n & \longrightarrow & \mathrm{PGL}_n & \longrightarrow & 1 \end{array} \tag{2.1}$$

The group $E[n](\overline{k})$ carries additional structure. It also has the Weil pairing e_E, which is a non-degenerate alternating galois covariant pairing taking values in the n-th roots of unity

$$e_E : E[n](\overline{k}) \times E[n](\overline{k}) \to \mu_n(\overline{k}).$$

The commutator of Θ_E corresponds to the Weil pairing, meaning that for $x, y \in \Theta_E$ we have

$$xyx^{-1}y^{-1} = \alpha_E(e_E(\beta_E(x), \beta_E(y))).$$

Definition 2. *A theta group for $E[n]$ is a central extension of group schemes*

$$1 \to \mathbb{G}_m \xrightarrow{\alpha} \Theta \xrightarrow{\beta} E[n] \to 1$$

such that the Weil-pairing on $E[n]$ corresponds to the commutator, i.e. for $x, y \in \Theta$ we have

$$xyx^{-1}y^{-1} = \alpha(e_E(\beta(x), \beta(y))).$$

Two theta groups

$$1 \to \mathbb{G}_m \to \Theta_i \to E[n] \to 1, \qquad i = 1, 2$$

are called isomorphic if there exists a group scheme isomorphism $\phi : \Theta_1 \to \Theta_2$ over k making the following diagram commutative.

$$\begin{array}{ccccccccc} 1 & \longrightarrow & \mathbb{G}_m & \longrightarrow & \Theta_1 & \longrightarrow & E[n] & \longrightarrow & 1 \\ & & \| & & \downarrow{\scriptstyle \phi} & & \| & & \\ 1 & \longrightarrow & \mathbb{G}_m & \longrightarrow & \Theta_2 & \longrightarrow & E[n] & \longrightarrow & 1 \end{array}$$

Over $\overline{k}$, all theta-groups are isomorphic to Θ_E as central extensions; see [2, Lemma 1.30].

Proposition 2. ([2, Proposition 1.31])**.** *Let $E[n]$ be the n-torsion subscheme of an elliptic curve E over a field k, equipped with its Weil pairing. The isomorphism classes of theta-groups for $E[n]$ over k are classified by $H^1(k, E[n])$.*

The theta group associated to $\delta \in H^1(k, E[n])$ may allow for a matrix representation $\Theta \to \mathrm{GL}_n$ that fits in a diagram like (2.1). This is measured by the *obstruction map* Ob introduced in [6] and [2]. This map can be obtained by taking non-abelian galois cohomology of the defining sequence of Θ_E:

$$\cdots \longrightarrow H^1(k, \Theta_E) \longrightarrow H^1(k, E[n]) \xrightarrow{\mathrm{Ob}} H^2(k, \mathbb{G}_m) = \mathrm{Br}(k) \longrightarrow \cdots .$$

Note that, except in some trivial cases, Ob is *not* a group homomorphism. The map Ob also has an interpretation in terms of n-coverings. Let $C \to E$ be an n-covering associated to $\delta \in H^1(k, E[n])$. We have that $\mathrm{Ob}(\delta) = 0$ if and only if C admits a model $C \to \mathbb{P}^{n-1}$ with $\mathrm{Aut}_{\overline{k}}(C/E) = E[n](\overline{k})$ acting linearly, in which case C is $\overline{k}$-isomorphic to E as a curve and the covering $C \to E$ is simply a translation composed with multiplication-by-n.

Remark 1. Note that if k is a number field, then any element in $\mathrm{Br}(k)$ that restricts to the trivial element in $\mathrm{Br}(k_v)$ in all completions k_v of k, is trivial itself. It follows that Ob is trivial on the n-Selmer group $S^{(n)}(E/k)$.

3 Visibility in Surfaces

Let E_1 be an elliptic curve over a perfect field k of characteristic distinct from 3. In what follows, we will consider $\delta \in H^1(k, E_1[3])$ with $\mathrm{Ob}(\delta) = 0$. A possible way of constructing an abelian surface A such that δ is visible in A starts by taking a suitable elliptic curve E_2/k together with a k-group scheme isomorphism $\lambda : E_1[3] \to E_2[3]$. Let $\Delta \subset E_1 \times E_2$ be the graph of λ so that

$$\Delta(\overline{k}) = \{(P, \lambda(P)) : P \in E_1[3](\overline{k})\}.$$

Let $A := (E_1 \times E_2)/\Delta$ and write $\phi : E_1 \times E_2 \to A$ for the corresponding isogeny. Since $\Delta \subset E_1[3] \times E_2[3]$, we have another isogeny $\phi' : A \to E_1 \times E_2$ such that $\phi' \circ \phi = 3$. We write p^* for the composition $E_1 \to (E_1 \times E_2) \xrightarrow{\phi} A$ and p_* for the composition $A \xrightarrow{\phi'} (E_1 \times E_2) \to E_1$ and q^*, q_* for the corresponding morphisms concerning E_2. It is straightforward to verify that p^*, q^* are embeddings, that $\phi = p^* - q^*$ (where the projections are understood), and that $\phi' = p_* \times q_*$.

We combine the galois cohomology of the short exact sequences

$$0 \to E_1 \xrightarrow{p^*} A \xrightarrow{q_*} E_2 \to 0,$$

$$0 \to E_2 \xrightarrow{q^*} A \xrightarrow{p_*} E_1 \to 0, \text{ and}$$

$$0 \to E_i[3] \to E_i \xrightarrow{3} E_i \to 0 \text{ for } i = 1, 2$$

to obtain the big (symmetric) commutative diagram with exact rows and columns

$$\begin{array}{ccccccc}
 & & & & E_2(k) & \xrightarrow{q^*} & A(k) \\
 & & & & \downarrow{\scriptstyle 3} & & \downarrow{\scriptstyle q_*} \\
 & & & & E_2(k) & = & E_2(k) \\
 & & & & \downarrow{\scriptstyle \alpha} & & \downarrow \\
E_1(k) & \xrightarrow{3} & E_1(k) & \longrightarrow & H^1(k, \Delta) & \longrightarrow & H^1(k, E_1) \\
\downarrow{\scriptstyle p^*} & & \| & & \downarrow & & \downarrow \\
A(k) & \xrightarrow{p_*} & E_1(k) & \longrightarrow & H^1(k, E_2) & \longrightarrow & H^1(k, A)
\end{array}$$

where we note that $H^1(k, \Delta) \simeq H^1(k, E_1[3]) \simeq H^1(k, E_2[3])$. We see that δ is visible in A precisely if $\delta \in H^1(k, E_1[3]) = H^1(k, \Delta)$ lies in the image of α, i.e., if the curve $C_{\lambda(\delta)}$ corresponding to $\lambda(\delta) \in H^1(k, E_2[3])$ has a rational point. We summarize these observations, which are due to Mazur.

Lemma 1. *Let E_1 be an elliptic curve over a perfect field k of characteristic distinct from 3 and let $\delta \in H^1(k, E[3])$ with $\mathrm{Ob}(\delta) = 0$. Suppose that there exists an elliptic curve E_2/k and a k-group scheme isomorphism $\lambda : E_1[3] \to E_2[3]$ such that the curve $C_{\lambda(\delta)}$ corresponding to $\lambda(\delta)$ has a k-rational point. Then δ is visible in the abelian surface $(E_1 \times E_2)/\Delta$ where Δ denotes the graph of λ.*

Mazur also observed, in the case of a number field k, that if $\delta \in S^{(3)}(E/k)$, then C_δ admits a plane cubic model. Furthermore, there is a pencil of cubics through the 9 flexes of C_δ, and each non-singular member corresponds to a 3-covering $C_t \to E_t$, where $E_t[3] \simeq E[3]$ and $C_t \to E_t$ represents δ. It is therefore easy to find a t such that C_t has a rational point; simply pick a rational point and solve for t. To refine the construction, one can ask

Question 1. Can one make $\delta \in H^1(k, E[3])$ visible in the jacobian of a genus 2 curve?

Note that $E_1 \times E_2$ is principally polarized via the product polarization. This gives rise to a Weil pairing on $(E_1 \times E_2)[3]$, corresponding to the product pairing. If A is a jacobian, then A must be principally polarized over $\overline{k}$. One way this could happen is if the isogeny $\phi : E_1 \times E_2 \to A$ gives rise to a principal polarization. This would be the case if the kernel Δ is a maximal isotropic subgroup of $E_1[3] \times E_2[3]$ with respect to the product pairing. That means that $\lambda : E_1[3] \to E_2[3]$ must be an *anti*-isometry, i.e. for all $P, Q \in E_1[3]$ we must have

$$e_{E_2}(\lambda(P), \lambda(Q)) = e_{E_1}(P, Q)^{-1}.$$

Note that the original cubic C is a member of the pencil that Mazur constructs, so in his construction λ is actually an *isometry*, i.e. it preserves the Weil-pairing. Below we consider a pencil of cubics that leads to an anti-isometry λ.

4 Anti-isometric Pencils

Let k be a perfect field of characteristic distinct from $2, 3$. Following [7], we associate to a ternary cubic form $F \in k[x, y, z]$ three more ternary cubic forms. Namely, the Hessian of F

$$H(F) := -\frac{1}{2} \begin{vmatrix} \frac{\partial F^2}{\partial x \partial x} & \frac{\partial F^2}{\partial x \partial y} & \frac{\partial F^2}{\partial x \partial z} \\ \frac{\partial F^2}{\partial y \partial x} & \frac{\partial F^2}{\partial y \partial y} & \frac{\partial F^2}{\partial y \partial z} \\ \frac{\partial F^2}{\partial z \partial x} & \frac{\partial F^2}{\partial z \partial y} & \frac{\partial F^2}{\partial z \partial z} \end{vmatrix},$$

the Caylean of F

$$P(F) := -\frac{1}{xyz} \begin{vmatrix} \frac{\partial F}{\partial x}(0, z, -y) & \frac{\partial F}{\partial y}(0, z, -y) & \frac{\partial F}{\partial z}(0, z, -y) \\ \frac{\partial F}{\partial x}(-z, 0, x) & \frac{\partial F}{\partial y}(-z, 0, x) & \frac{\partial F}{\partial z}(-z, 0, x) \\ \frac{\partial F}{\partial x}(y, -x, 0) & \frac{\partial F}{\partial y}(y, -x, 0) & \frac{\partial F}{\partial z}(y, -x, 0) \end{vmatrix}$$

and a ternary cubic form denoted $Q(F)$, for which we refer to [7, Section 11.2]. For most cases one can take $Q(F)$ to be $H(P(F))$ or $P(H(F))$, but there are some exceptional cases where $P(F), Q(F)$ span an appropriate pencil and $P(F), H(P(F))$ do not. The left action of GL_3 on k^3 induces a right action of GL_3 on ternary cubic forms (or, more generally, on $k[x, y, z]$). For a ternary cubic form F and an $M \in \mathrm{GL}_3$ we denote this action simply by $F \circ M$. The significance

of the three associated ternary cubic forms lies in the fact that $H(F)$ depends covariantly on F (of weight 2) and $P(F)$ and $Q(F)$ depend contravariantly on F (of weights 4 and 6 respectively). This means that for every ternary cubic form F and every $M \in \mathrm{GL}_3$ we have, with $d := \det M$ that

$$\begin{aligned} H(F \circ M) &= d^2 H(F) \circ M \\ P(F \circ M) &= d^4 P(F) \circ M^{-T} \\ Q(F \circ M) &= d^6 Q(F) \circ M^{-T}, \end{aligned}$$

where M^{-T} denotes the inverse transpose of M.

Now consider a smooth cubic curve C in $\mathbb{P}^2$ given by the zero locus of a ternary cubic form F. Then C has exactly 9 different flex points Φ, which all lie on the (not necessarily smooth) curve given by $H(F) = 0$. The smoothness of C guarantees that F and $H(F)$ will be linearly independent over k. Hence Φ can be described as the intersection $F = H(F) = 0$. We call Φ the *flex scheme* of C. At least one of $P(F)$ and $Q(F)$ turns out to be nonsingular (still assuming that C is nonsingular) and the intersection $P(F) = Q(F) = 0$ equals the flex points Φ^* of the nonsingular cubics among $P(F)$ and $Q(F)$ (if, say, $P(F)$ is nonsingular, then Φ^* can of course also be written as $P(F) = H(P(F)) = 0$).

We can consider the pencil of cubics through Φ, explicitly given by

$$C_{(s:t)} : sF(x, y, z) + tH(F)(x, y, z) = 0. \tag{4.1}$$

Classical invariant theory tells us the following. This pencil has exactly 4 singular members and all other members have flex scheme equal to Φ. Conversely, any nonsingular cubic with flex scheme Φ occurs in this pencil. Furthermore, both $P(sF + tH(F))$ and $Q(sF + tH(F))$ are linear combinations of $P(F)$ and $Q(F)$. This shows that the flex scheme Φ^* is independent of the choice of C through Φ and only depends on Φ. We call Φ^* the dual flex scheme of Φ and we will justify this name below.

As a simple, but important example we take $F := x^3 + y^3 + z^3$. Then we compute

$$H(F) = -108xyz, \quad P(F) = -54xyz, \quad Q(F) = 324(x^3 + y^3 + z^3).$$

Now define Φ_0 to be the flex scheme of $F = 0$, i.e.

$$\Phi_0 := \{[x : y : z] \in \mathbb{P}^2 : x^3 + y^3 + z^3 = xyz = 0\}. \tag{4.2}$$

Then we see that the flex scheme given by $P(F) = Q(F) = 0$ (which is the flex scheme of $Q(F) = 0$) equals Φ_0, i.e.

$$\Phi_0^* = \Phi_0.$$

The pencil of cubics through Φ_0 (note that $108 \neq 0$ in k), which is given by

$$s(x^3 + y^3 + z^3) = txyz,$$

is a model over k for the universal elliptic curve over the (genus zero) modular curve $X(3)$; see [5, p. 225]. Geometrically all flex schemes are linear transformations of each other. In particular, for any flex scheme Φ there exists an $M \in \mathrm{GL}_3(\overline{k})$ such that $\Phi = M\Phi_0$. This shows that the pencil (4.1) associates to a general flex scheme Φ is a twist of the universal elliptic curve over $X(3)$.

The contravariance of P and Q implies that the assignment $\Phi \mapsto \Phi^*$ has the contravariance property that for any flex scheme Φ and $M \in \mathrm{GL}_3$

$$(M\Phi)^* = M^{-T}\Phi^*. \tag{4.3}$$

We also note that this implies that the assignment $\Phi \mapsto \Phi^{**} := (\Phi^*)^*$ is covariant in the sense that for any flex scheme Φ and $M \in \mathrm{GL}_3$ we have $(M\Phi)^{**} = M\Phi^{**}$. Writing $\Phi = M\Phi_0$ and using $(\Phi_0)^{**} = \Phi_0^* = \Phi_0$ we now get

$$\Phi^{**} = (M\Phi_0)^{**} = M\Phi_0^{**} = M\Phi_0 = \Phi.$$

This justifies calling Φ^* the *dual* flex scheme of Φ.

Remark 2. In the discussion above it was convenient to consider just one projective plane $\mathbb{P}^2$. A more canonical way would be to consider a projective plane $\mathbb{P}^2$ with coordinates x, y, z (for a point) and the dual projective plane, denoted $(\mathbb{P}^2)^*$, where the point with coordinates u, v, w describes the line $ux+vy+wz = 0$. Now let C be a smooth cubic curve in $\mathbb{P}^2$ given by the zero locus of the ternary cubic form $F(x, y, z)$ with flex scheme Φ. The 9 tangent lines through Φ determine 9 points in $(\mathbb{P}^2)^*$. Generically, these 9 points in $(\mathbb{P}^2)^*$ will not be the flex points of a smooth cubic curve, hence generically there will a unique cubic curve going through these points. This curve in $(\mathbb{P}^2)^*$ is exactly given by the zero locus of the Caylean, i.e. $P(F)(u, v, w) = 0$; see also [10, pp.151,190–191]. Moreover, if the characteristic of k is zero, then it turns out that this cubic curve is nonsingular if and only if the j-invariant of C is nonzero.

To any flex scheme Φ we associate a group $\Theta(\Phi) \subset \mathrm{GL}_3$ as follows. Choose a nonsingular cubic curve C through Φ and let E be its jacobian. After identifying E and C as curves over $\overline{k}$, we get an action of $E[3]$ on C, which extends to a linear action on $\mathbb{P}^2$. This determines an embedding $\chi : E[3] \to \mathrm{PGL}_3$. Obviously, the image $\chi(E[3])$ only depends on Φ. We define $\Theta(\Phi)$ to be the inverse image of $\chi(E[3])$ in GL_3. Actually $\Theta(\Phi)$ can be defined just in terms of Φ, without choosing C, since it turns out that $\chi(E[3])$ consists exactly of the linear transformations that preserve Φ. (One way of quickly finding these linear transformations explicitly is by using the fact that, for any two distinct points of Φ, the line through these two points intersects Φ in a unique third point.) The construction gives rise to the theta group

$$1 \to \mathbb{G}_m \to \Theta(\Phi) \to E[3] \to 1.$$

Note that the isomorphism class of this theta group may still depend on the choice of identification of C with E. This corresponds to the choice of an isomorphism between $\Theta(\Phi)/\mathbb{G}_m$ and $E[3]$. If Φ is defined over k, then $E[3]$ and

$\Theta(\Phi)$ are also defined over k and the element in $H^1(k, E[3])$ corresponding to this theta group is the same as the element corresponding to the 3-covering $C \to C/E[3] \simeq E$ for any nonsingular cubic curve C through Φ. The construction also shows that for any $M \in \mathrm{GL}_3$ we have

$$\Theta(M\Phi) = M\Theta(\Phi)M^{-1}. \tag{4.4}$$

Proposition 3. *Let $\Phi_1 \subset \mathbb{P}^2$ be a flex scheme and let $\Phi_2 := \Phi_1^*$ be the dual flex scheme. For $i = 1, 2$ let C_i be a smooth plane cubic with flex scheme Φ_i, denote its jacobian by E_i and consider an induced theta group*

$$1 \longrightarrow \mathbb{G}_m \xrightarrow{\alpha_i} \Theta(\Phi_i) \xrightarrow{\beta_i} E_i[3] \longrightarrow 1\ . \tag{4.5}$$

Then the outer automorphism $(-T) : \mathrm{GL}_3 \to \mathrm{GL}_3$ given by $M \mapsto M^{-T}$, yields an isomorphism $\Theta(\Phi_1) \to \Theta(\Phi_2)$. There exists an anti-isometry $\lambda : E_1[3] \to E_2[3]$ making the following diagram commutative.

$$\begin{array}{ccccccccc} 1 & \longrightarrow & \mathbb{G}_m & \xrightarrow{\alpha_1} & \Theta(\Phi_1) & \xrightarrow{\beta_1} & E_1[3] & \longrightarrow & 1 \\ & & \downarrow{\scriptstyle x\mapsto x^{-1}} & & \downarrow{\scriptstyle (-T)} & & \downarrow{\scriptstyle \lambda} & & \\ 1 & \longrightarrow & \mathbb{G}_m & \xrightarrow{\alpha_2} & \Theta(\Phi_2) & \xrightarrow{\beta_2} & E_2[3] & \longrightarrow & 1 \end{array} \tag{4.6}$$

In particular, let $\delta_i \in H^1(k, E_i[3])$ correspond to the theta group (4.5). Then under the isomorphism $H^1(k, E_1[3]) \simeq H^1(k, E_2[3])$ induced by λ, the cocycle δ_1 maps to δ_2.

Proof. Once the isomorphism $\Theta(\Phi_1) \to \Theta(\Phi_2)$ given by $M \mapsto M^{-T}$ is established, the existence of an isomorphism $\lambda : E_1[3] \to E_2[3]$ making the diagram (4.6) commutative, follows immediately. That λ must be an anti-isometry can readily be seen as follows. Let $P, Q \in E_1[3]$ and choose $x, y \in \Theta(\Phi_1)$ such that $P = \beta_1(x)$ and $Q = \beta_1(y)$. Then

$$\begin{aligned} \alpha_2(e_{E_2}(\lambda(P), \lambda(Q))) &= \alpha_2(e_{E_2}(\beta_2(x^{-T}), \beta_2(y^{-T}))) \\ &= x^{-T}y^{-T}x^Ty^T \\ &= (xyx^{-1}y^{-1})^{-T} \\ &= \alpha_1(e_{E_1}(\beta_1(x), \beta_1(y)))^{-T} \\ &= \alpha_1(e_{E_1}(P, Q)^{-1}). \end{aligned}$$

The last statement of the proposition is also immediate, so we are left with establishing $(-T) : \Theta(\Phi_1) \xrightarrow{\sim} \Theta(\Phi_2)$. It suffices to show that for a flex scheme $\Phi \subset \mathbb{P}^2$ we have $\Theta(\Phi)^{-T} = \Theta(\Phi^*)$. Write $\Phi = M\Phi_0$ for some $M \in \mathrm{GL}_3$ with Φ_0 given by (4.2). Then a straightforward calculation shows that $\Theta(\Phi_0)^{-T} = \Theta(\Phi_0)$. We also know that $\Phi_0^* = \Phi_0$, so we get $\Theta(\Phi_0)^{-T} = \Theta(\Phi_0^*)$. Together with (4.3) and (4.4) we finally obtain,

$$\begin{aligned}\Theta(\Phi)^{-T} &= \Theta(M\Phi_0)^{-T}\\ &= M^{-T}\Theta(\Phi_0)^{-T}M^T\\ &= M^{-T}\Theta(\Phi_0^*)(M^{-T})^{-1}\\ &= \Theta(M^{-T}\Phi_0^*)\\ &= \Theta((M\Phi_0)^*)\\ &= \Theta(\Phi^*).\end{aligned}$$

□

Remark 3. The construction above of the dual flex scheme Φ^* of a flex scheme Φ involved choosing a smooth cubic going through Φ. Without using theta groups, it was not obvious from this construction that the degree 9 étale algebra $k(\Phi)$ is isomorphic to $k(\Phi^*)$. However, there exists a nice explicit geometric construction of the dual flex scheme that remedies these shortcomings of the earlier construction. Given a flex scheme Φ, we proceed as follows. We label its 9 points over $\overline{k}$ with $P_1, \ldots, P_9$. There are 4 sets of 3 lines, (corresponding to the 4 singular members of the pencil of cubics through ϕ) containing these points. We label the line that contains P_i, P_j, P_k with $l_{\{i,j,k\}}$. One can label the points such that the subscripts are

$$\begin{array}{llll}\{1,2,3\} & \{1,4,7\} & \{1,5,9\} & \{1,6,8\}\\ \{4,5,6\}, & \{2,5,8\}, & \{2,6,7\}, & \{2,4,9\},\\ \{7,8,9\} & \{3,6,9\} & \{3,4,8\} & \{3,5,7\}\end{array}$$

Naturally, two different lines $l_{\{i_1,j_1,k_1\}}, l_{\{i_2,j_2,k_2\}}$ meet in a unique point. If for example $i_1 = i_2$, then the intersection point is P_{i_1}. If the two sets $\{i_1, j_1, k_1\}$ $\{i_2, j_2, k_2\}$ are disjoint, then the two lines meet in a point outside Φ. We name this point $L_{\{i_3,j_3,k_3\}}$, where $\{i_1, j_1, k_1, i_2, j_2, k_2, i_3, j_3, k_3\} = \{1, \ldots, 9\}$. As it turns out, the four points that have i in their label all lie on a line p_i. It is also straightforward to check that the p_i together with the $L_{\{i,j,k\}}$ form a configuration in $(\mathbb{P}^2)^*$ that is completely dual to the P_i with the $l_{\{i,j,k\}}$. The p_i form the $\overline{k}$ points of a flex scheme in $(\mathbb{P}^2)^*$, which is justifiably a flex scheme Φ^* dual to Φ, and its construction immediately implies the contravariance property $(M\Phi)^* = M^{-T}\Phi^*$.

We can easily verify that the two constructions of Φ^* coincide for one flex scheme, for instance Φ_0. The general result then follows because any flex scheme can be expressed as $M\Phi_0$ for some $M \in \mathrm{GL}_3(\overline{k})$.

Since the action of $\mathrm{Gal}(\overline{k}/k)$ on $\{P_1, \ldots, P_9\}$ must act via collinearity-preserving permutations, we see that if $\sigma(P_i) = P_{\sigma(i)}$, then $\sigma(p_i) = p_{\sigma(i)}$. Hence, we see that the $\overline{k}$-points of Φ and its dual have the same Galois action and hence $k(\Phi)$ is isomorphic as a k-algebra to $k(\Phi^*)$.

5 Recovering the Genus 2 Curve

Let k be a field and let E_1, E_2 be two elliptic curves over k with an anti-isometry $\lambda : E_1[3] \to E_2[3]$ and denote by Δ the graph of λ as before. Recall that $E_1 \times E_2$ is

principally polarized via the product polarization and that the induced polarization on $A := (E_1 \times E_2)/\Delta$ is also principal in this case. It is a classical fact that if A is not geometrically isomorphic to a product of elliptic curves, then A (together with its principal polarization) is isomorphic to the jacobian of a genus 2 curve C. Let us assume from now on that E_1 and E_2 are non-isogenous. In [8] it is shown that in this case A is always isomorphic over k to the jacobian of a genus 2 curve C/k. This is enough to get our main theoretical result.

Theorem 4. *Let E be an elliptic curve over a number field k and let $\xi \in Ш(E/k)[3]$. Then ξ is visible in the jacobian of a genus 2 curve C/k.*

Proof. Let $\delta \in S^{(3)}(E/k)$ be a cocycle representing ξ. By Proposition 2, there is a 3-covering $C_\delta \to E$ corresponding to δ. According to Remark 1, we have that $\mathrm{Ob}(\delta) = 0$ and hence that $C_\delta \subset \mathbb{P}^2$. Let $\Phi \subset \mathbb{P}^2$ be its flex scheme. The construction in Section 4 gives us a pencil of cubics through Φ^*, so we can easily pick a non-singular one with a rational point. It follows from Proposition 3 that such a curve is of the form $C_{\lambda(\delta)}$ for some elliptic curve E_2 and some anti-isometry $\lambda : E[3] \to E_2[3]$.

This places us in the situation of Lemma 1, so δ is visible in an abelian surface $A = (E \times E_2)/\Delta$. We have ensured that λ is an anti-isometry, which implies that the surface is principally polarized. As long as we make sure that E, E_2 are non-isogenous (and this is easy given the freedom we have in choosing $C_\lambda(\delta)$) it follows that A is a jacobian. □

Remark 4. We could of course state a more general result about visibility of elements $\delta \in H^1(k, E[3])$ with $\mathrm{Ob}(\delta) = 0$ for an elliptic curves E over a perfect field k of characteristic distinct from 2 or 3. Note however that if k is too small, there might not be enough non-isogenous elliptic curves available. The exclusion of fields of characteristic 3 is a serious one, the exclusion of non-perfect fields less so. Most of what we are saying could be generalized to the non-perfect case, basically because for an elliptic curve over any field of characteristic distinct from 3, the multiplication by 3 map is separable. The exclusion of fields of characteristic 2 stems from the fact that the necessary invariant theory in this case is not readily available.

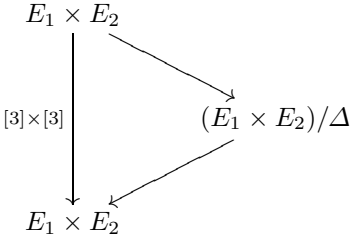
We continue with the construction of the genus 2 curve C. Define the divisor $\Theta := 0_1 \times E_2 + E_1 \times 0_2$ on $E_1 \times E_2$, which gives a principal polarization on $E_1 \times E_2$. Next, consider the set $\mathcal{D}$ of effective divisors on $E_1 \times E_2$ over $\overline{k}$ which are linearly equivalent to 3Θ and invariant under Δ. Also consider the set $\mathcal{C}$ of effective divisors C on A over $\overline{k}$ whose pull-back to $E_1 \times E_2$ are linearly equivalent to 3Θ and which satisfy $(C \cdot C) = 2$. Frey and Kani show that there exist unique curves $D \in \mathcal{D}$ and $C \in \mathcal{C}$ defined over k which are invariant under multiplication by -1. Furthermore, because E_1 and E_2 are not isogenous, D and C are irreducible smooth curves of genus 10 and 2 respectively and the natural map $D \to C$ is unramified of degree 9.

If k is a perfect field of characteristic distinct from 2 or 3, the curves D and C can be explicitly constructed as follows. Embed E_1 in $\mathbb{P}^2$, given by, say

$F(x,y,z)=0$, for a ternary cubic F/k (such an F is readily obtained if E_1 is given by a Weierstrass model). Express E_2 as $G := sP(F)+tQ(F)=0$ for some $s,t \in k$. This way, we obtain an embedding of $E_1 \times E_2$ in $\mathbb{P}^2 \times \mathbb{P}^2$ given by

$$F(x,y,z) = G(u,v,w) = 0.$$

Moreover, by appealing to Proposition 3 we obtain that the curve on this surface given by $xu+yv+zw=0$ must be the curve D. The genus 2 curve C is the image of D in $(E_1 \times E_2)/\Delta$.



The map $[3]\times[3]$ is much more accessible, though. We claim that the subgroup of $E_1[3] \times E_2[3]$ under which D is invariant is equal to Δ. Hence, we can find a (singular) model of C as a curve on $E_1 \times E_2$ by computing $([3]\times[3])(D)$. This can easily be done via interpolation, as explained in the next section by means of an example. As for our claim above, suppose that D is invariant under some $\sigma \in E_1[3] \times E_2[3]$ with $\sigma \notin \Delta$. Without loss of generality we may assume that $\sigma = (P, 0_{E_2}) \in E_1[3] \times E_2[3]$ with $P \neq 0_{E_1}$. Denote by $M \in \mathrm{PGL}_3(\overline{k})$ the linear action corresponding to translation by P. Now for all $([x:y:z],[u:v:w])$ on D we have

$$(x,y,z)(u,v,t)^T = (x,y,z)M^T(u,v,w)^T = 0.$$

This yields $(u,v,t) = (x,y,z) \times (x,y,z)M^T$, where $\times$ denotes the standard cross product. This association actually defines a birational transformation $\phi : \mathbb{P}^2 \to \mathbb{P}^2$ (a Cremona transformation with singular points corresponding to the eigenspaces of M). Note that ϕ is defined on all the $[x:y:z]$ on E_1, so the image of E_1 under ϕ is an irreducible curve birational to E_1. Together with the assumption that E_1 and E_2 are not isogenous, we get that this image intersects E_2 in only finitely many points, so D is not invariant under σ.

6 Examples

Following the first example in [4, Table 1], consider the elliptic curve $681b1$ (in Cremona's notation), given by the minimal Weierstrass equation

$$E_1 : y^2+xy = x^3+x^2-1154x-15345.$$

It turns out that the plane cubic curve

$$C_1 : x^3+5x^2y+5x^2z+2xy^2+xyz+xz^2+y^3-5y^2z+2yz^2+6z^3 = 0$$

defines an element ξ (up to inverse) of order three in $Ш(E_1/\mathbb{Q})$. The contravariants, denoted P_0, Q_0, are given by

$$\begin{aligned} P_0 = &-478x^3 + 2525x^2y + 916x^2z - 1127xy^2 + 29xyz \\ &-160xz^2 + 753y^3 - 1228y^2z + 260yz^2 + 301z^3, \\ Q_0 = &-122314x^3 + 618551x^2y + 191092x^2z - 271157xy^2 - 7825xyz \\ &-28120xz^2 + 184011y^3 - 264916y^2z + 55892yz^2 + 73663z^3. \end{aligned}$$

Now the curve

$$C_2 : 55033P_0 - 235Q_0 = 0$$

has a rational point $[x : y : z] = [10 : 8 : 7]$ and its jacobian is the elliptic curve $681c1$, given by the minimal Weierstrass equation

$$E_2 : y^2 + y = x^3 - x^2 + 2.$$

To construct the corresponding genus two curve C such that ξ becomes visible in its jacobian we could take the curve in $C_1 \times C_2 \subset \mathbb{P}^2 \times \mathbb{P}^2$ with coordinates $([x : y : z], [u : v : w])$ given by the equation $xu + yv + zw = 0$, and take its image under $C_1 \times C_2 \to E_1 \times E_2$, since this is a twist of $[3] \times [3] : E_1 \times E_2 \to E_1 \times E_2$ anyway. We will follow Section 5 more closely. Obviously, E_1 is given by $F = 0$ if we define

$$F := y^2z + xyz - (x^3 + x^2z - 1154xz^2 - 15345z^3).$$

The contravariants of the ternary cubic F are given by

$$\begin{aligned} P = &-2308x^3 + 3462x^2y - 5x^2z - 275056xy^2 + 5xyz \\ &+6xz^2 + 136951y^3 + 13853y^2z - 3yz^2, \\ Q = &-725020x^3 + 1087530x^2y + 27721x^2z - 65861608xy^2 - 27721xyz \\ &-30xz^2 + 32749549y^3 + 3217559y^2z + 15yz^2 + 24z^3. \end{aligned}$$

Write $j(s, t)$ for the j-invariant of the curve given by $sP + tQ = 0$. The j-invariant of E_2 equals $-4096/2043$ and the equation $j(s, t) = -4096/2043$ has exactly one solution in $\mathbb{P}^1(\mathbb{Q})$, namely $[s : t] = [55033 : -235]$ (compare with the definition of C_2). This gives us a new model for E_2, namely

$$E_2 : 55033P - 235Q = 0.$$

We consider the surface $E_1 \times E_2$ embedded in $\mathbb{P}^2 \times \mathbb{P}^2$ as

$$F(x, y, z) = 0, \quad 55033P(u, v, w) - 235Q(u, v, w) = 0.$$

The curve D on this surface is given by

$$xu + yv + zw = 0.$$

The image of D under multiplication by 3 on $E_1 \times E_2$ is the genus two curve C. Using the defining properties of C from Section 5 (such as the invariance under multiplication by -1), we get that as a curve on $E_1 \times E_2$ it must be of the form

$$axu + byv + czw + dxw + ezu = 0$$

for some $a, b, c, d, e \in \mathbb{Q}$. We simply generate 4 points on C (over a number field), compute the image under multiplication by 3 of these points and solve for a, b, c, d, e. If the dimension of the solution space is greater than 1, we must of course add points (or take 4 better ones) so that the solution space becomes 1-dimensional. This gives us our equation for C. By a linear change of the u, v, w coordinates we can change the model for E_2 back to the original minimal Weierstrass model. Thus, the model for $E_1 \times E_2$ embedded in $\mathbb{P}^2 \times \mathbb{P}^2$ is

$$E_1 : y^2z + xyz = x^3 + x^2z - 1154xz^2 - 15345z^3,$$

$$E_2 : v^2w + vw^2 = u^3 - u^2w + 2w^3$$

and C is the curve on this surface given by

$$4xu - 155zu + xv + 2yv - 40xw + yw + 1314zw = 0.$$

Hyperelliptic models for C are

$$\begin{aligned} Y^2 + (X+1)Y &= 3X^5 + 5X^4 + X^3 - 8X^2 - 5X + 2 \text{ or} \\ Y^2 &= (3X-1)(X+1)(4X^3 + 4X^2 - 9). \end{aligned}$$

Next, consider the elliptic curve $2006e1$, given by the minimal Weierstrass equation

$$E_1 : y^2 + xy = x^3 + x^2 - 58293654x - 171333232940.$$

It turns out that the plane cubic curve

$$C_1 : 20x^3+44x^2y+21x^2z-77xy^2+71xyz+44xz^2+31y^3+3y^2z+150yz^2+z^3 = 0$$

defines an element ξ (up to inverse) of order three in $Ш(E_1/\mathbb{Q})$. In the sixth example in [4, Table 1] the elliptic curve E_2 which 'explains' $Ш(E_1/\mathbb{Q})$ is $2006d1$. However, for this choice of E_2, there only exists an isometry between $E_1[3]$ and $E_2[3]$ and not an anti-isometry. The corresponding abelian surface $(E_1 \times E_2)/\Delta$ visualizing ξ will not be the jacobian of a genus 2 curve. If instead we take for E_2 the elliptic curve $6018c1$, then we do have an anti-isometry between $E_1[3]$ and $E_2[3]$. Following the same route as in the first example, we find that ξ is visible in the jacobian of the genus 2 curve C with hyperelliptic models

$$\begin{aligned} Y^2 + (X^2+X)Y = &-9675X^6 - 94041X^5 - 914X^4 + 1301674X^3 - 352310X^2 \\ &- 2071181X - 945269 \quad \text{or} \\ Y^2 = &\, 43(2X+13)(18X^2 - 81X + 89)(25X^3 + 193X^2 + 224X + 76). \end{aligned}$$

7 Applications to 3-Descent

In this section we survey some of the ways in which explicit visibility might aid computations of Mordell-Weil groups and related quantities of elliptic curves. We recall that given an abelian variety A over a number field k, the group $Ш(A/k) \subset H^1(k, A)$ consists of the cocycle classes that are everywhere locally trivial. It measures the difference between the Mordell-Weil group $A(k)$ and the *Selmer group* $S(A/k) \subset \varprojlim_n H^1(k, A[n])$ which is an everywhere local approximation to $A(k)$, in the sense that the following sequence is exact.

$$0 \to A(k) \to S(A/k) \to Ш(A/k) \to 0$$

An n-descent usually means an explicit computational process to compute

$$S^{(n)}(A/k) = S(A/k)/nS(A/k) \subset H^1(k, A[n]).$$

It provides a bound on $\mathrm{rk}A(k)$ and conversely, if $A(k)$ is known, then we can use

$$0 \to A(k)/nA(k) \to S^{(n)}(A/k) \to Ш(A/k)[n] \to 0$$

to compute $\#Ш(A/k)[n]$ and thus obtain information on $\#Ш(A/k)$. In principle, one can use visibility to refine this information. We will argue using an example. Stein and Watkins [12] found the following elliptic curve

$$E : y^2 + xy = x^3 - x^2 + 94x + 9.$$

Using a 2-descent and some point searching (with for instance Magma [1]) it is straightforward to verify that $E(\mathbb{Q}) \simeq \mathbb{Z} \times \mathbb{Z}$ and that $\#Ш(E/\mathbb{Q})[2] = 1$. Using a 3-descent (see [2, 3, 11], implemented in Magma), with unproved S-unit data we find that

$$C_1 : x^3 + 2x^2z + 2xy^2 + xyz - xz^2 - y^3 + 3y^2z - 6yz^2 + z^3 = 0,$$
$$C_2 : x^3 - 2xy^2 + 3xyz + 2y^3 + y^2z + yz^2 + 3z^3 = 0$$

are 3-coverings of E that have points everywhere locally and we can verify by looking at preimages of representatives of $E(\mathbb{Q})/3E(\mathbb{Q})$ that C_1, C_2 have no rational points. The same process allows us to find more than 18 such spaces, verifying unconditionally that $\#Ш(E/\mathbb{Q})[3] \geq 9$. The conditional 3-descent computation suggests that C_1, C_2 represent cocycles generating $S^{(3)}(E/\mathbb{Q})/E(\mathbb{Q})$, so one expects that $\#Ш(E/\mathbb{Q})[3] = 9$ and indeed BSD predicts that $\#Ш(E/\mathbb{Q}) = 9$.

Visibility could help with proving that $\#Ш(E/\mathbb{Q})[3^\infty] = 9$. The construction in this paper yields an abelian surface $A = \mathrm{Jac}(C)$, together with a map

$$\phi_* : Ш(E/\mathbb{Q}) \times Ш(E'/\mathbb{Q}) \to Ш(A/\mathbb{Q})$$

where we know that $\ker(\phi_*)$ is contained in the 3-torsion, because multiplication-by-three factors through ϕ. If we can make sure that $\ker(\phi_*)$ contains the classes

represented by C_1, C_2 (this implies that $E'(\mathbb{Q})$ is of rank at least 2), it may well be that $\#Ш(A/\mathbb{Q})[3] = 1$. If we can compute $S^{(3)}(A/\mathbb{Q})$, we can check this and the result would follow.

Thus, visibility allows us to substitute a 9-descent on an elliptic curve with a 3-descent on the Jacobian of a genus 2 curve. Both are theoretically computable, but in neither case does it seem practical at this point. Since A has a 3-isogeny to $E \times E'$, the 3-torsion algebra (generically of degree 80), splits in two algebras of degrees 72 and 8 respectively. However, doing class group computations for degree 72 algebras over $\mathbb{Q}$ still seems well out of range.

It is conceivable that some appropriate galois-stable set $\mathcal{S}$ of divisors on C exists with $\#\mathcal{S} < 72$. The group $\mathrm{Sp}_4(\mathbb{F}_3)$ has an index 27 subgroup, for instance, predicting a transitive action on 27 objects somewhere. If for some fixed divisor D_0 we have that $A[3] = \langle [D - D_0] : D \in \mathcal{S} \rangle$, it may be possible to adapt ideas about fake Selmer groups [9] for application to A and only require class group information for algebras of degree $\#\mathcal{S}$.

At this point it is unclear if this approach has any advantages to a direct 9-descent on E and whether either method can be made practical for the example given in this section.

Acknowledgments. The authors would like to thank the referees, who provided various helpful comments which found their way into this article.

References

[1] Bosma, W., Cannon, J., Playoust, C.: The Magma computer algebra system is described in the Magma algebra system. I. The user language. J. Symbolic Comput. 24(3-4), 235–265 (1997)
[2] Cremona, J.E., Fisher, T.A., O'Neil, C., Simon, D., Stoll, M.: Explicit n-descent on elliptic curves. I. Algebra. J. Reine Angew. Math. 615, 121–155 (2008)
[3] Cremona, J.E., Fisher, T.A., Stoll, M.: Minimisation and reduction of 2-, 3- and 4-coverings of elliptic curves, arXiv: 0908.1741 (2009), http://arxiv.org/abs/0908.1741
[4] Cremona, J.E., Mazur, B.: Visualizing elements in the Shafarevich-Tate group. Experiment. Math. 9(1), 13–28 (2000)
[5] Mazur, B.: Visualizing elements of order three in the Shafarevich-Tate group. Asian J. Math. 3(1), 221–232 (1999); Sir Michael Atiyah: a great mathematician of the twentieth century
[6] O'Neil, C.: The period-index obstruction for elliptic curves. J. Number Theory 95(2), 329–339 (2002)
[7] Fisher, T.: The Hessian of a genus one curve, arXiv: math/0610403 (2006), http://arxiv.org/abs/math/0610403
[8] Frey, G., Kani, E.: Curves of genus 2 covering elliptic curves and an arithmetical application. In: Arithmetic Algebraic Geometry (Texel, 1989), pp. 153–176 (1991)
[9] Poonen, B., Schaefer, E.F.: Explicit descent for Jacobians of cyclic covers of the projective line. J. Reine Angew. Math. 488, 141–188 (1997)

[10] Salmon, G.: A treatise on the higher plane curves, 3rd edn. Hodges, Foster, and Figgis, Grafton Street, Dublin (1879)
[11] Schaefer, E.F., Stoll, M.: How to do a p-descent on an elliptic curve. Trans. Amer. Math. Soc. 356(3), 1209–1231 (2004)
[12] Stein, W.A., Watkins, M.: A database of elliptic curves—first report. In: Fieker, C., Kohel, D.R. (eds.) ANTS 2002. LNCS, vol. 2369, pp. 267–275. Springer, Heidelberg (2002), http://wstein.org/Tables/ecdb/

On Weil Polynomials of $K3$ Surfaces

Andreas-Stephan Elsenhans[1,⋆] and Jörg Jahnel[2]

[1] Universität Bayreuth, Mathematisches Institut, Universitätsstraße 30, D-95447 Bayreuth, Germany
stephan.elsenhans@uni-bayreuth.de
[2] Fachbereich 6, Mathematik, Universität Siegen, Walter-Flex-Straße 3, D-57072 Siegen, Germany
jahnel@mathematik.uni-siegen.de

Abstract. For $K3$ surfaces, we derive some conditions the characteristic polynomial of the Frobenius on the étale cohomology must satisfy. These conditions may be used to speed up the computation of Picard numbers and the decision of the sign in the functional equation[⋆⋆]. Our investigations are based on the Artin-Tate formula.

1 Introduction

An algebraic integer such that all its conjugates have absolute value $\sqrt{r}$ is called an r-Weil number. Correspondingly, a possibly reducible monic polynomial $\Phi \in \mathbb{Z}[T]$ such that all roots have absolute value $\sqrt{r}$ is called an r-Weil polynomial.

Let q be a prime power and $r = q^k$. Then, for every smooth projective variety V over $\mathbb{F}_q$, the eigenvalues of the Frobenius endomorphism Frob on the étale cohomology $H^k_{\text{ét}}(V_{\overline{\mathbb{F}}_q}, \mathbb{Q}_l)$ are r-Weil numbers [3, Lemme 1.7]. Conversely, every q^k-Weil number is an eigenvalue of Frob on $H^k_{\text{ét}}(V_{\overline{\mathbb{F}}_q}, \mathbb{Q}_l)$ for a suitable smooth projective variety V over $\mathbb{F}_q$. Actually, this fact is a direct consequence of the results of T. Honda [9].

In this note, we will study the Weil numbers of $K3$ surfaces. As the second Betti number of a $K3$ surface is $b_2(V) = 22$ and q is always a root of the characteristic polynomial, the possible Weil numbers are of degree at most 20.

We will show that *not* all q^2-Weil polynomials $\Phi \in \mathbb{Z}[T]$ satisfying $\deg \Phi = 22$ and $\Phi(q) = 0$ occur as characteristic polynomials of Frob on the étale cohomology of $K3$ surfaces. Concerning $K3$ surfaces of fixed degree, even more restrictions result. Our investigations are based on the Artin-Tate formula which we will recall in section 3.

⋆ The first author was partially supported by the Deutsche Forschungsgemeinschaft (DFG) through a funded research project.

⋆⋆ The computer part of this work was executed on the Sun Fire V20z Servers of the Gauß Laboratory for Scientific Computing at the Göttingen Mathematisches Institut. Both authors are grateful to Prof. Y. Tschinkel for the permission to use these machines as well as to the system administrators for their support.

G. Hanrot, F. Morain, and E. Thomé (Eds.): ANTS-IX 2010, LNCS 6197, pp. 126–141, 2010.

An application. The characteristic polynomial of Frob may be computed by counting points over extensions of the ground field. Indeed, for V a $K3$ surface over $\mathbb{F}_q$, the Lefschetz trace formula [13, Ch. VI, §12] yields $\mathrm{tr}(\mathrm{Frob}^e) = \#V(\mathbb{F}_{q^e}) - q^{2e} - 1$.

When we denote the eigenvalues of Frob by $r_1, \ldots, r_{22}$, we have $\mathrm{tr}(\mathrm{Frob}^e) = r_1^e + \cdots + r_{22}^e =: \sigma_e(r_1, \ldots, r_{22})$. Newton's identity [20]

$$s_k(r_1, \ldots, r_{22}) = \frac{1}{k} \sum_{j=0}^{k-1} (-1)^{k+j+1} \sigma_{k-j}(r_1, \ldots, r_{22}) s_j(r_1, \ldots, r_{22})$$

shows that the knowledge of $\sigma_e(r_1, \ldots, r_{22})$, for $e = 1, \ldots, k$, is sufficient in order to determine the coefficient $(-1)^k s_k$ of T^{22-k} of the characteristic polynomial Φ of Frob. Further, there is the functional equation

$$q^{\deg \Phi} \Phi(T) = \pm T^{\deg \Phi} \Phi(q^2/T) \tag{1}$$

which, as $\deg \Phi = 22$, relates the coefficient of T^k with that of T^{22-k}.

Nevertheless, this method is time-consuming. The size of the fields to be considered grows exponentially. One would like to avoid point counting over large fields and, nevertheless, determine Φ sufficiently well in order to decide things such as the sign in (1). Algorithms of this type were presented in [6]. For example, Algorithm 22 of [6] verifies that the geometric Picard rank is 2, having counted points over $\mathbb{F}_p, \ldots, \mathbb{F}_{p^9}$ for p a prime number.

The main result of the present article leads to a more substantial approach to this problem. In fact, we will show that certain hypothetical characteristic polynomials are impossible, in general. This leads to an improvement of [6, Algorithm 22]. Sections 7 and 8 will be devoted to examples showing how this improvement works in practice.

Remark 1. A continuation of this application, which we have in mind, is the computation of the geometric Picard rank for $K3$ surfaces over $\mathbb{Q}$. Here, the general strategy is to use reduction modulo p. One applies the inequality

$$\mathrm{rk}\,\mathrm{Pic}(V_{\overline{\mathbb{Q}}}) \leq \mathrm{rk}\,\mathrm{Pic}(V_{\overline{\mathbb{F}}_p})$$

which is true for every smooth variety V over $\mathbb{Q}$ and every prime p of good reduction. Then, the number of eigenvalues of Frob which are roots of unity is an upper bound for the Picard number. More details are given in [6] and [7].

2 The Galois Group of a Weil Polynomial

For a randomly chosen irreducible polynomial over $\mathbb{Q}$, one expects the Galois group to be the full symmetric group. In this sense, the irreducible factors of a Weil polynomial are not very random.

When we consider the operation of Frob on a cohomology group of even degree, cyclotomic factors do arise. They correspond to the algebraic part of the cohomology, i.e., to the image of the Picard group and its analogues in higher codimension. The corresponding Galois group is always abelian.

Concerning the remaining factors, still, there are restrictions on the Galois group. Note that, for each root of an irreducible r-Weil polynomial not of degree 1, the complex conjugate is a root, too. This means, the roots come in pairs. The product of each pair is equal to r. The Galois group therefore acts on the pairs. For a suitable integer n, it is a subgroup of the semi-direct product $(\mathbb{Z}/2\mathbb{Z})^n \rtimes S_n \subset S_{2n}$. Here, each factor $(\mathbb{Z}/2\mathbb{Z})$ acts on one pair by complex conjugation. The complex conjugation itself belongs to the center of the group.

An experimental result. One could ask for further restrictions on the Galois group. For that, we computed the characteristic polynomial of Frob for a few thousand randomly chosen $K3$ surfaces. In each case, the factorization of that polynomial had precisely one irreducible factor which was not cyclotomic. This coincides with Zarhin's results [18] for ordinary $K3$ surfaces.

Furthermore, in the vast majority of the examples, the Galois group of the last factor was actually equal to the semi-direct product $(\mathbb{Z}/2\mathbb{Z})^n \rtimes S_n \subset S_{2n}$. For example, this was true for 875 out of 1 000 $K3$ surfaces of degree 2 over $\mathbb{F}_3$ and 923 out of 1 000 $K3$ surfaces of degree 2 over $\mathbb{F}_7$.

The resolvent algebra. Let $\Phi \in \mathbb{Q}[T]$ be a polynomial such that its set of roots is of the particular form $\{r_1, r_1', \ldots, r_n, r_n'\}$ such that $r_1 r_1' = \ldots = r_n r_n' =: r \in \mathbb{Q}$. Then, the sums $r_1 + r_1', \ldots, r_n + r_n'$ are the roots of a polynomial $R \in \mathbb{Q}[T]$ of half the degree. We will call R the *resolvent polynomial* and $A := \mathbb{Q}[T]/R$ the *resolvent algebra* of Φ.

Remarks 2. a) When Φ is an r-Weil polynomial of even degree, the assumption is satisfied if and only if $\sqrt{r}$ is a root of even multiplicity (or no root) of Φ. In this case, $(-\sqrt{r})$ has even multiplicity, too.

In fact, this means exactly that Φ fulfills the functional equation (1) with the plus sign.

b) On the other hand, when one wants to verify that a given polynomial satisfying the functional equation is, in fact, a Weil polynomial, the resolvent is helpful. Observe that the roots of the initial polynomial are all of absolute value $\sqrt{r}$ if and only if the roots of the resolvent are all real and in the interval $[-2\sqrt{r}, 2\sqrt{r}]$. That property may easily be checked using Sturm's chain theorem.

This is a fast and exact replacement of [6, Algorithm 23].

3 The Artin-Tate Formula

Let us recall the Artin-Tate conjecture in the special case of a $K3$ surface.

Conjecture 3 (Artin-Tate)**.** *Let V be a $K3$ surface over a finite field $\mathbb{F}_q$. Denote by ρ the rank and by Δ the discriminant of the Picard group of V, defined over $\mathbb{F}_q$. Then,*

$$|\Delta| = \frac{\lim_{T \to q} \frac{\Phi(T)}{(T-q)^\rho}}{q^{21-\rho} \# \mathrm{Br}(V)} .$$

Here, Φ denotes the characteristic polynomial of Frob *on* $H^2_{\text{ét}}(V_{\overline{\mathbb{F}}_q}, \mathbb{Q}_l)$. *Finally,* $\mathrm{Br}(V)$ *is the Brauer group of* V.

Remarks 4. i) The characteristic polynomial Φ is independent of the choice of the auxiliary prime l as long as $l \neq p$ for $q = p^e$ [3, Théorème 1.6].

ii) For a general non-singular, projective surface, the exponent of q in the numerator is $b_2(V) - h_{02}(V) - \rho$. Here, $h_{02}(V)$ denotes the Hodge number.

iii) The Artin-Tate conjecture is proven for most $K3$ surfaces. Most notably, the Tate conjecture implies the Artin-Tate conjecture [11, Theorem 6.1].

iv) The Tate conjecture claims that all zeroes of Φ of the form $q\zeta$ for ζ a root of unity belong to the algebraic part of $H^2_{\text{ét}}(V_{\overline{\mathbb{F}}_q}, \mathbb{Q}_l)$. I.e., it asserts that the transcendental part never generates a zero of this form.

The evidence for this is overwhelming as far as $K3$ surfaces are concerned. The Tate conjecture is proven for elliptic $K3$ surfaces [1] and ordinary $K3$ surfaces [15]. In characteristic different from 2 and 3, even more particular cases were successfully treated [16].

v) It is expected that $\mathrm{Br}(V)$ is always a finite group. This is actually equivalent to the Tate conjecture. In this case, $\#\mathrm{Br}(V)$ is automatically a perfect square. We may therefore compute the square class of Δ making use of the Artin-Tate conjecture.

An unconditional version of the Artin-Tate formula

Notation 5. i) For n a positive integer, we will denote by μ_n the sheaf of n-th roots of unity with respect to the fppf topology. When l is a prime number, we put $H^d_{\text{fppf}}(V_{\overline{\mathbb{F}}_q}, T_l\mu) := \varprojlim_e H^d_{\text{fppf}}(V_{\overline{\mathbb{F}}_q}, \mu_{l^e})$.

ii) For l a prime number and M an abelian group, the notation $M_{l\text{-pow}}$ shall be used for the l-power torsion subgroup of M. Similarly, we will write $M_{l\text{-div}} \subseteq M_{l\text{-pow}}$ for the subgroup of infinitely l-divisible elements.

iii) We will denote by M^{Frob} and M_{Frob} the invariants, respectively coinvariants, under the operation of Frob on the abelian group M. The coinvariants may have torsion even when M is torsion-free. Write M'_{Frob} for the torsion-free quotient.

Proposition 6. *Let V be a $K3$ surface over a finite field $\mathbb{F}_q$ and l be any prime. Write Φ for the characteristic polynomial of* Frob *on the étale cohomology of $V_{\overline{\mathbb{F}}_q}$ and ρ for the multiplicity of q as a zero of Φ.*

i) *Then, the Brauer group* $\mathrm{Br}(V)$ *is a torsion group. The quotient*

$$\mathrm{Br}_0(V, l) := \mathrm{Br}(V)_{l\text{-pow}} / \mathrm{Br}(V)_{l\text{-div}}$$

is a finite group of square order.

ii) *Further, $H^2_{\text{fppf}}(V_{\overline{\mathbb{F}}_q}, T_l\mu)^{\text{Frob}}$ is a free $\mathbb{Z}_l$-module of rank ρ.*

iii) *Denote by Δ_l the discriminant of the bilinear form*

$$H^2_{\text{fppf}}(V_{\overline{\mathbb{F}}_q}, T_l\mu)^{\text{Frob}} \times H^2_{\text{fppf}}(V_{\overline{\mathbb{F}}_q}, T_l\mu)^{\text{Frob}} \longrightarrow \mathbb{Z}_l$$

defined by Poincaré duality. Then,

$$\nu_l(\Delta_l) = \nu_l\left(\frac{\lim_{T\to q}\frac{\Phi(T)}{(T-q)^\rho}}{q^{21-\rho}\#\mathrm{Br}_0(V,l)}\right).$$

Proof. i) Finiteness of $\mathrm{Br}_0(V,l)$ follows immediately from [8, (8.9)]. Further, there is a non-degenerate alternating pairing $\mathrm{Br}_0(V,l)\times\mathrm{Br}_0(V,l)\to\mathbb{Q}_l/\mathbb{Z}_l$ constructed in [19, Lemma 3.4.1]. This ensures that the group order is a perfect square.

ii) and iii) We denote the zeroes of Φ by $r_1,\ldots,r_{22}$.

First case. $l\neq p$. Here, $H:=H^2_{\mathrm{fppf}}(V_{\overline{\mathbb{F}}_q},T_l\mu)=H^2_{\text{ét}}(V_{\overline{\mathbb{F}}_q},\mathbb{Z}_l(1))$ is the same as l-adic étale cohomology. It is a free $\mathbb{Z}_l$-module of rank 22. In the present case, the operation of Frob on H is known to be semi-simple [4, Corollary 1.10]. The eigenvalues are $r_1/q,\ldots,r_{22}/q$. Assertion ii) follows immediately from this.

Further, we have $\nu_l(\Delta_l)=\nu_l(\#\mathrm{coker}(H^{\mathrm{Frob}}\to\mathrm{Hom}(H^{\mathrm{Frob}},\mathbb{Z}_l)))$, the map being induced by Poincaré duality. Identifying $\mathrm{Hom}(H,\mathbb{Z}_l)$ with H, the module $\mathrm{Hom}(H^{\mathrm{Frob}},\mathbb{Z}_l)$ goes over into H'_{Frob}. Here, as shown in [19, Proposition 1.4.2], $(H_{\mathrm{Frob}})_{\mathrm{tors}}\cong\mathrm{Br}_0(V,l)$. Further, the order of the cokernel of the canonical homomorphism $H^{\mathrm{Frob}}\to H_{\mathrm{Frob}}$ is equal to the l-primary part of $\prod_{r_j\neq q}(1-r_j/q)$. Altogether, this implies the claim.

Second case. $l=p$. Here, some modifications are necessary which are described in [11]. More concretely, the short exact sequence

$$0\to\mathrm{Pic}(V_{\overline{\mathbb{F}}_q})\otimes_{\mathbb{Z}}\mathbb{Z}_p\to H^2_{\mathrm{fppf}}(V_{\overline{\mathbb{F}}_q},T_p\mu)\to\varprojlim\mathrm{Br}(V_{\overline{\mathbb{F}}_q})_{p^n}\to 0$$

immediately shows that $H:=H^2_{\mathrm{fppf}}(V_{\overline{\mathbb{F}}_q},T_p\mu)$ is a torsion-free $\mathbb{Z}_p$-module. Otherwise, its structure is rather different from the previous case. The rank of H is, in general, less than 22. Eigenvalues of Frob are only those r_j/q which are units in $\overline{\mathbb{Q}}_p$ [11, 1.4]. But this is enough to show ii).

Generally, there are unipotent connected quasi-algebraic groups U^d and étale group schemes D^d_n for $d=2,3$ and $n\gg 0$ which provide short exact sequences $0\to U^d(\overline{\mathbb{F}}_q)\to H^d_{\mathrm{fppf}}(V_{\overline{\mathbb{F}}_q},\mu_{p^n})\to D^d_n(\overline{\mathbb{F}}_q)\to 0$. For varying n, the vector groups $U^3(\overline{\mathbb{F}}_q)$ are connected by identities. Further, $D^3_n=0$. Hence, if $\dim U^3=s$ then $\#H^3_{\mathrm{fppf}}(V_{\overline{\mathbb{F}}_q},T_p\mu)^{\mathrm{Frob}}=q^s$ the operation of Frob being semi-simple. Actually, one has $s=0$ except when V is supersingular.

Poincaré duality is available [12, Theorem 5.2 and Corollary 2.7.c)] only at the level of torsion coefficients. Thereby, $U^2(\overline{\mathbb{F}}_q)$ and $U^3(\overline{\mathbb{F}}_q)$ are dual to each other. One has $\varprojlim U^2(\overline{\mathbb{F}}_q)=0$ and $R^1\varprojlim U^2(\overline{\mathbb{F}}_q)=0$ as the connecting homomorphisms are zero. Hence, $H^2_{\mathrm{fppf}}(V_{\overline{\mathbb{F}}_q},T_p\mu)\cong\varprojlim D^2_n(\overline{\mathbb{F}}_q)$. Further, it turns out that the homomorphism $H_{\mathrm{Frob}}\to\mathrm{Hom}(H^{\mathrm{Frob}},\mathbb{Z}_p)$ does not need to be bijective. It has a cokernel exactly of order q^s (cf. [11, Lemma 5.2]).

Summarizing, we find that Δ_p has the same p-adic valuation as

$$q^s\cdot\prod_{\substack{\nu_p(r_j/q)=0\\ r_j\neq q}}(1-r_j/q)\,.$$

For iii), it remains to show the following. Up to p-adic units, the product of the remaining factors, i.e. $\prod_{\nu_p(r_j/q)\neq 0}(1-r_j/q)$, equals q^{s-1}. This is worked out in [11, sec. 7]. □

Remark 7. The Tate conjecture implies $H^2_{\text{fppf}}(V_{\bar{\mathbb{F}}_q}, T_l\mu)^{\text{Frob}} \cong \text{Pic}(V_{\bar{\mathbb{F}}_q})\otimes_{\mathbb{Z}}\mathbb{Z}_l$. Further, it is equivalent to $\text{Br}(V)_{l\text{-div}} = 0$. Thus, Proposition 6 goes over into the Artin-Tate formula in its usual form. However, the Tate conjecture is unknown in general, even for $K3$ surfaces. For this reason, we prefer to apply the version of the Artin-Tate formula which holds unconditionally.

4 The Rank-1 Condition

Let V be a $K3$ surface of degree d over a finite field $\mathbb{F}_q$. Assume that q is a simple zero of the characteristic polynomial of Frob. Then, the Tate conjecture is true for V and the arithmetic Picard rank is equal to 1. The discriminant of $\text{Pic}(V)$ is equal to d. A comparison with the analytic discriminant computed via the Artin-Tate formula leads to a non-trivial condition for hypothetical Weil polynomials.

Remarks 8. a) This is a condition for rank-1 surfaces of a given degree d. It is not a condition for $K3$ surfaces, in general.

b) The degree of a $K3$ surface may be any even integer greater than zero. On the other hand, when the arithmetic Picard rank is 1, the number $(-q)$ is necessarily among the Frobenius eigenvalues. Hence, the Artin-Tate formula can generate only even numbers.

c) The Artin-Tate conjecture implies the inequality $\#\text{Br}(V)|\Delta| \leq 2^{22-\rho}q$. Thus, the left hand side is $\text{O}(q)$. Observe the following striking consequence. Over the field $\mathbb{F}_q$, there is no $K3$ surface of a square-free degree $d > 2^{21}q$ and arithmetic Picard rank 1.

Remark 9. The rank-1 condition may be extended to other situations where a subgroup of the Picard group is known. For this, one has to compare the predicted ranks and discriminants with the known ones.

5 The Field Extension Condition

Notation 10. For q a positive integer, let Φ be a q^2-Weil polynomial. Then, we will write

$$E^{(c)}_{\Phi} := \prod_{r_j\neq q}\frac{q^c - r_j^c}{q - r_j}\Big/ q^{(c-1)(21-\rho)}\,.$$

Here, r_j runs over all the zeroes of Φ. Further, ρ is the multiplicity of the zero q.

Observation 11 (Field extension for the characteristic polynomial). Let V be any smooth, projective variety over $\mathbb{F}_q$ and $\prod_j(T - r_j)$ the characteristic polynomial of Frob on $H^2_{\text{ét}}(V_{\bar{\mathbb{F}}_q}, \mathbb{Q}_l)$. Then, the corresponding polynomial for $V_{\mathbb{F}_{q^d}}$ is $\prod_j(T - r_j^d)$.

Theorem 12. *Let V be a $K3$ surface over $\mathbb{F}_q$. Further, let c be a positive integer. Then, for Φ the characteristic polynomial of* Frob, *the expression $E_\Phi^{(c)}$ is a perfect square in $\mathbb{Q}$.*

Proof. If there is an $r_j \neq q$ such that $r_j^c = q^c$ then $E_\Phi^{(c)} = 0$. Otherwise, for every prime l, $H^2_{\text{fppf}}(V_{\overline{\mathbb{F}}_q}, T_l\mu)^{\text{Frob}_{q^c}}$ is a sublattice of finite index in $H^2_{\text{fppf}}(V_{\overline{\mathbb{F}}_q}, T_l\mu)^{\text{Frob}_q}$. In particular, the discriminants differ by a factor being a perfect square. Dividing the Artin-Tate formulas for $V_{\mathbb{F}_{q^c}}$ and $V_{\mathbb{F}_q}$ through each other yields that $\nu_l(E_\Phi^{(c)})$ is even for every l. Finally, it is easy to see that $E_\Phi^{(c)} > 0$. □

Remark 13. Assume the Tate conjecture. Then, $E_\Phi^{(c)}$ is non-zero if and only if $\operatorname{rk}\operatorname{Pic}(V_{\mathbb{F}_q}) = \operatorname{rk}\operatorname{Pic}(V_{\mathbb{F}_{q^c}})$.

Definition 14. We will call the condition on $E_\Phi^{(c)}$ to be a perfect square, the *field extension condition* for the field extension $\mathbb{F}_{q^c}/\mathbb{F}_q$.

Explicit computation of the expression $E_\Phi^{(c)}$. Our goal is now to describe the square class of $E_\Phi^{(c)}$ more explicitly. It will turn out that, for an arbitrary Weil polynomial, $E_\Phi^{(c)}$ may be a non-square. In other words, Theorem 12 provides a non-trivial condition.

Remark 15. A priori, there are infinitely many conditions, one for each value of c. The main result of this section is that there is in fact only one condition. Further, this condition may be checked easily.

Lemma 16. *Let $f \in \mathbb{Q}[T]$ be a q^2-Weil polynomial. Suppose $f(q) \neq 0$ and $f(-q) \neq 0$. Then, for $r_1, \ldots, r_{2l}$ the zeroes of f,*

$$\prod_{j=1}^{2l} \frac{q^c - r_j^c}{q - r_j} \in \begin{cases} (\mathbb{Q}^*)^2 \cup \{0\} & \text{for } c \text{ odd,} \\ f(-q)(\mathbb{Q}^*)^2 \cup \{0\} & \text{for } c \text{ even.} \end{cases}$$

Further, the left hand side is actually in $f(-q)(\mathbb{Q}^*)^2$ for $c = 2$.

Proof. First observe that, for $c = 2$, the numerators $q^2 - r_j^2$ are all non-zero according to the assumption. Hence, the additional assertion is clear once we showed the main one.

For that, let us start with the contribution of one pair of complex conjugate roots. Put $r_j = q(u + iv)$. Then, the corresponding factor is

$$\begin{aligned} \frac{(q^c - r_j^c)(q^c - \overline{r}_j^c)}{(q - r_j)(q - \overline{r}_j)} &= \frac{(q^c - q^c(u+iv)^c)(q^c - q^c(u-iv)^c)}{(q - q(u+iv))(q - q(u-iv)} \\ &= q^{2(c-1)} \prod_{k=1}^{c-1} (1 - \zeta_c^k(u+iv))(1 - \zeta_c^k(u-iv)) \,. \end{aligned}$$

Using $(u + iv)(u - iv) = 1$, we get

$$q^{2(c-1)} \prod_{k=1}^{c-1} (1 - 2\zeta_c^k u + \zeta_c^{2k}) \,.$$

Next, for $k \neq c/2$, let us multiply the factors for k and $c-k$. This yields

$$(1 - 2\zeta_c^k u + \zeta_c^{2k})(1 - 2\zeta_c^{c-k} u + \zeta_c^{2c-2k}) = 2 + 4u^2 - 8u \operatorname{Re}(\zeta_c^k) + 2\operatorname{Re}(\zeta_c^{2k}) .$$

As $\operatorname{Re}(\zeta_c^{2k}) = 2\operatorname{Re}(\zeta_c^k)^2 - 1$, the latter term is the same as

$$4u^2 - 8u \operatorname{Re}(\zeta_c^k) + 4\operatorname{Re}(\zeta_c^k)^2 = (2u - 2\operatorname{Re}(\zeta_c^k))^2 .$$

Multiplying over all k such that $1 \leq k < c/2$, we find a square in $\mathbb{Q}(u)$. Consequently, up to the factor for $k = c/2$, if present, the contribution of the pair $\{r_j, \overline{r}_j\}$ is a square in the resolvent algebra A of f.

Multiplying over all l pairs means to form a norm for the extension $A/\mathbb{Q}$. As the norm of a square is a square, the result is a perfect square in $\mathbb{Q}$. For c odd, this completes the argument.

For c even, the factors for $k = c/2$ are still missing. These are the ones for $\zeta_c^k = -1$. We find the product

$$\prod_{j=1}^{l} (1 + r_j/q)(1 + \overline{r}_j/q) = q^{-2l} f(-q) .$$

The assertion follows. □

Proposition 17. *Let Φ be a q^2-Weil polynomial of even degree. Then,*

$$E_\Phi^{(c)} \in \begin{cases} (\mathbb{Q}^*)^2 \cup \{0\} & \text{for } c \text{ odd}, \\ q\Phi(-q)(\mathbb{Q}^*)^2 \cup \{0\} & \text{for } c \text{ even}. \end{cases}$$

For $c = 2$, we actually have $E_\Phi^{(c)} \in q\Phi(-q)(\mathbb{Q}^*)^2$.

Proof. *First case: c* is odd.

Then, the denominator $q^{(c-1)(21-\rho)}$ is a perfect square. The zeroes $(-q)$ contribute factors q^{c-1} which are squares, too. Finally, the contribution to $E_\Phi^{(c)}$ of the zeroes not being real is a perfect square according to Lemma 16.

Second case: c is even.

If $(-q)$ is a zero of Φ then $E_\Phi^{(c)} = 0$. This coincides with the claim as $\Phi(-q) = 0$. Otherwise, write $\Phi(T) = (T-q)^\rho f(T)$ where $f(q) \neq 0$ and $f(-q) \neq 0$. By assumption, ρ is even. Hence, $q^{(c-1)(21-\rho)}$ is in the square class of q. Further, the zeroes of Φ differing from q are exactly the zeroes of f. Their contribution is in $f(-q)(\mathbb{Q}^*)^2$ for $c = 2$ and in $f(-q)(\mathbb{Q}^*)^2 \cup \{0\}$, in general. As ρ is even, $f(-q)(\mathbb{Q}^*)^2$ is the same class as $\Phi(-q)(\mathbb{Q}^*)^2$. The assertion follows. □

Corollary 18. *Let $f \in \mathbb{Z}[T]$ be a q^2-Weil polynomial.*

i) *Then, all field extension conditions for $\mathbb{F}_{q^c}/\mathbb{F}_q$ are satisfied if only if the condition for the quadratic extension $\mathbb{F}_{q^2}/\mathbb{F}_q$ does hold.*

ii) *For extensions of odd degree, the field extension condition is always satisfied.*

iii) *If $\mathbb{F}_q$ and $\mathbb{F}_{q^2}$ lead to different Picard ranks then all the field extension conditions are satisfied.*

Remark 19. One might want to study the field extension conditions for $\mathbb{F}_{q^{ac}}/\mathbb{F}_{q^a}$, i.e., for an extended ground field. Our calculations show that this does not lead to new conditions.

Simplification of the field extension test. Denote by ϕ_n the n-th cyclotomic polynomial. Correspondingly, there is the monic polynomial ψ_n given by $\psi_n(T) := q^{\varphi(n)}\phi_n(T/q)$. This is a q^2-Weil polynomial.

Lemma 20. *Let $n > 1$ be an integer. Then,*

$$\psi_n(-q) \in \begin{cases} (\mathbb{Q}^*)^2 & \text{if } n \text{ is not a power of } 2\,, \\ 2(\mathbb{Q}^*)^2 & \text{for } n = 2^m, m \geq 2\,, \\ \{0\} & \text{for } n = 2\,. \end{cases}$$

Proof. It is well known (see, e.g., [14, sec. 3]) that $\phi_n(-1) = 1$ unless n is a power of 2. Further, the formula $\phi_{2^e}(t) = t^{2^{e-1}} + 1$ shows $\phi_2(-1) = 0$ and $\phi_{2^e}(-1) = 2$ for $e > 1$. Observe, finally, that $\varphi(n)$ is always even for $n > 2$. □

Remark 21. The result used here is a very special case of the value of a cyclotomic polynomial at a root of unity.

Theorem 22. *Let $\Phi \in \mathbb{Z}[T]$ be a q^2-Weil polynomial of even degree. Factorize Φ as*

$$\Phi(T) = (T-q)^r(T+q)^s\psi_{n_1}(T)\cdot\ldots\cdot\psi_{n_k}(T)\Phi_1(T)$$

such that Φ_1 has no root being a root of unity multiplied by q. Denote by M the number of the powers of 2 among the $n_1, \ldots, n_k$. Then,

i) *if c is odd then $E_\Phi^{(c)} \in (\mathbb{Q}^*)^2 \cup \{0\}$.*

ii) *If c is even and $s > 0$ then $E_\Phi^{(c)} = 0$ for every c.*

iii) *Finally, if c is even and $s = 0$ then $E_\Phi^{(c)} \in 2^M q\Phi_1(-q)(\mathbb{Q}^*)^2 \cup \{0\}$. Furthermore, for $c = 2$, one actually has*

$$E_\Phi^{(2)} \in 2^M q\Phi_1(-q)(\mathbb{Q}^*)^2\,.$$

Proof. i) and ii) are immediate consequences from Proposition 17. For iii), observe the assumption implies that r is even. In particular, $(-2q)^r$ is a perfect square. The assertion now follows from Proposition 17 together with Corollary 20. □

Remark 23. Suppose $\Phi \in \mathbb{Z}[T]$ is a q^2-Weil polynomial of degree 22. In order to show that Φ may not be the characteristic polynomial of the Frobenius for a $K3$ surface over $\mathbb{F}_q$, it suffices to verify that $s = 0$ and $2^M q\Phi_1(-q)$ is a non-square.

Example 24. As an example, we look at $K3$ surfaces of Picard rank 18 such that the Picard group is defined over an extension of odd degree. Then, $(-q)$ is not an eigenvalue of the Frobenius. The transcendental part of the characteristic polynomial is given by $(T^4 + aT^3 + bT^2 + aq^2T + q^4)$. Hence, the field extension condition usually requires that $(2q^2 - 2aq + b)q$ is a perfect square. If, however, the cyclotomic factors contain an odd number of type ψ_{2^n} then $2(2q^2 - 2aq + b)q$ is required to be a square.

6 The Special Case of a Degree-2 Surface – Twisting

When a $K3$ surface has a non-trivial automorphism, one can hope to get more conditions by inspecting the corresponding twist. This is the case for degree-2 surfaces.

The Twist. Let the $K3$ surface V be given by the equation

$$w^2 = f_6(x, y, z)\,.$$

Then, for n a non-square in $\mathbb{F}_q$, consider the twist $\widetilde{V}$ of V given by

$$nw^2 = f_6(x, y, z)\,.$$

Fact 25. *Assume that $q, r_2, \ldots, r_{22}$ are the eigenvalues of* Frob *for V. Then, the eigenvalues for $\widetilde{V}$ are $q, -r_2, \ldots, -r_{22}$.*

Proof. For e even, $V_{\mathbb{F}_{q^e}}$ and $\widetilde{V}_{\mathbb{F}_{q^e}}$ are isomorphic. When e is odd, we have

$$\#V(\mathbb{F}_{q^e}) + \#\widetilde{V}(\mathbb{F}_{q^e}) = 2\cdot\#\mathbf{P}^2(\mathbb{F}_{q^e}) = 2q^{2e} + 2q^e + 2\,.$$

It is easy to check that the Lefschetz trace formula, applied to the eigenvalues $q, -r_2, \ldots, -r_{22}$, implies exactly this relation. □

Proposition 26. *Let V be a $K3$ surface of degree 2 over $\mathbb{F}_q$. Denote by Φ the characteristic polynomial of* Frob *for V and by $\widetilde{\Phi}$ the corresponding polynomial for the twist $\widetilde{V}$.*

i) *Then, Φ has a simple zero at q if and only if $\widetilde{\Phi}$ does not have a zero at $(-q)$. I.e., the rank-1 condition can be applied to the one precisely when the field extension condition is non-empty for the other one.*

ii) *The two conditions are equivalent to each other.*

Proof. i) immediately follows from Fact 25.

ii) By assumption, we can write $\Phi(T) = (T - q)(T + q)^{2n-1} f(T)$. Here both, $f(q)$ and $f(-q)$ are non-zero. Fact 25 shows, the corresponding polynomial for the twist is $\widetilde{\Phi}(T) = (T - q)^{2n} f(-T)$. Using these two formulas, one can make the conditions explicit. The rank-1 condition for Φ simply means $(2q)^{2n-1} f(q) = 2$ in $\mathbb{Q}^*/(\mathbb{Q}^*)^2$ which is equivalent to saying that $qf(q)$ is a perfect square. This is precisely the field extension condition for $\widetilde{\Phi}$. □

7 Examples

Let us show in detail the data for a few examples. Our goal is to illustrate how the Artin-Tate conditions work in practice.

Example 27 (A $K3$ surface of degree 2 over $\mathbb{F}_7$). Consider the surface V over $\mathbb{F}_7$, given by

$$w^2 = y^6 + 3z^6 + 5xz^5 + 5x^2y^4 + x^2z^4 + 3x^3y^3 + x^3z^3 + 5x^4y^2 + x^4z^2 + 5x^5y + 2x^6\,.$$

Over $\mathbb{F}_7, \ldots, \mathbb{F}_{7^9}$, there are exactly 66, 2 378, 118 113, 5 768 710, 282 535 041,

13 841 275 877, 678 223 852 225, 33 232 944 372 654, and 1 628 413 551 007 224 points. We claim that $\operatorname{rk}\operatorname{Pic}(V_{\overline{\mathbb{F}}_7}) = 2$.

Assuming the characteristic polynomial of the Frobenius has more than two zeroes of the form 7 times a root of unity, [6, Algorithm 22] leaves us with three candidates Φ_1, Φ_2, Φ_3.

$$\begin{aligned}\Phi_i(t) \;=\; & t^{22} - 16\,t^{21} + 140\,t^{20} - 1\,029\,t^{19} + 5\,831\,t^{18} - 36\,015\,t^{17} + 268\,912\,t^{16} \\ & - 1\,882\,384\,t^{15} + 11\,529\,602\,t^{14} - 46\,118\,408\,t^{13} + a_i t^{12} + b_i t^{11} + c_i t^{10} \\ & + (-1)^{j_i}[-110\,730\,297\,608\,t^9 + 1\,356\,446\,145\,698\,t^8 - 10\,851\,569\,165\,584\,t^7 \\ & + 75\,960\,984\,159\,088\,t^6 - 498\,493\,958\,544\,015\,t^5 + 3\,954\,718\,737\,782\,519\,t^4 \\ & - 34\,196\,685\,556\,119\,429\,t^3 + 227\,977\,903\,707\,462\,860\,t^2 \\ & - 1\,276\,676\,260\,761\,792\,016\,t + 3\,909\,821\,048\,582\,988\,049]\end{aligned}$$

for

$$\begin{aligned} j_1 = 0, \quad (a_1, b_1, c_1) &= (161\,414\,428, -1\,129\,900\,996, \quad 7\,909\,306\,972)\,, \\ j_2 = 1, \quad (a_2, b_2, c_2) &= (\ 80\,707\,214, \qquad\qquad 0, -3\,954\,653\,486)\,, \\ j_3 = 1, \quad (a_3, b_3, c_3) &= (121\,060\,821, \qquad\qquad 0, -5\,931\,980\,229)\,. \end{aligned}$$

Each of the three polynomials leads to an upper bound of 4 for the rank of the geometric Picard group. All three have roots of absolute value 7, only. Applying the Artin-Tate formula, we find the following.

Table 1. Hypothetical ranks and discriminants

polynomial	field	arithmetic Picard rank	$\#\mathrm{Br}(V)\lvert\Delta\rvert$
Φ_1	$\mathbb{F}_7$	2	58
	$\mathbb{F}_{49}$	2	4524
Φ_2	$\mathbb{F}_7$	1	4
	$\mathbb{F}_{49}$	2	1996
Φ_3	$\mathbb{F}_7$	1	6
	$\mathbb{F}_{49}$	2	2997

The polynomial Φ_1 is excluded by the field extension condition as the two values in the rightmost column define different square classes. On the other hand, the rank-1 condition excludes Φ_2 and Φ_3 since we have a degree-2 example. Thus, relative to the Tate conjecture, geometric Picard rank 2 is proven.

Example 28 (continuation). On the same surface, point counting over $\mathbb{F}_{7^{10}}$ leads to a number of 79 792 267 067 823 523. For the characteristic polynomial of the Frobenius, we find the two candidates Φ_4, Φ_5,

$$\begin{aligned}\Phi_i(t) \;=\; & t^{22} - 16\,t^{21} + 140\,t^{20} - 1\,029\,t^{19} + 5\,831\,t^{18} - 36\,015\,t^{17} + 268\,912\,t^{16} \\ & - 1\,882\,384\,t^{15} + 11\,529\,602\,t^{14} - 46\,118\,408\,t^{13} + 40\,353\,607\,t^{12} + a_i t^{11} \\ & + (-1)^{j_i}[\,-1\,977\,326\,743\,t^{10} + 110\,730\,297\,608\,t^9 - 1\,356\,446\,145\,698\,t^8 \\ & + 10\,851\,569\,165\,584\,t^7 - 75\,960\,984\,159\,088\,t^6 + 498\,493\,958\,544\,015\,t^5 \\ & - 3\,954\,718\,737\,782\,51\,9t^4 + 34\,196\,685\,556\,119\,429\,t^3 \\ & - 227\,977\,903\,707\,462\,860\,t^2 + 1\,276\,676\,260\,761\,792\,016\,t \\ & - 3\,909\,821\,048\,582\,988\,049]\end{aligned}$$

for $j_4 = 0$, $a_4 = 0$, $j_5 = 1$, and $a_5 = 564\,950\,498$. Φ_4 corresponds to the minus sign in the functional equation, Φ_5 to the case of the plus sign. Both candidates, according to the Tate conjecture, imply geometric Picard rank 2.

To decide which sign is the right one, one would first check the absolute values of the roots. Unfortunately, both polynomials only have roots of absolute value 7. The Artin-Tate formula provides the picture given in the table below.

Table 2. Hypothetical ranks and discriminants

polynomial	field	arithmetic Picard rank	$\#\mathrm{Br}(V)\lvert\Delta\rvert$
Φ_4	$\mathbb{F}_7$	1	2
	$\mathbb{F}_{49}$	2	997
Φ_5	$\mathbb{F}_7$	2	55
	$\mathbb{F}_{49}$	2	4125

Thus, Φ_5 is excluded by the field extension condition. The minus sign in the functional equation is correct.

Example 29 (A $K3$ surface of degree 8 over $\mathbb{F}_3$). Consider the complete intersection V of the three quadrics in $\mathbf{P}^5_{\mathbb{F}_3}$, given by q_1, q_2, and q_3,

$$\begin{aligned} q_1 &:= -xy + xz + xu + xv + xw - y^2 - yz - yv + yw \\ &\qquad + z^2 + zu + zw - u^2 - uw + v^2 + w^2\,, \\ q_2 &:= -x^2 + xy + xz - xv + xw - y^2 + yz - yu - yv \\ &\qquad + yw - zu - zw + uw - v^2 + vw\,, \\ q_3 &:= xu - yz\,. \end{aligned}$$

V is smooth and, therefore, a $K3$ surface. As q_3 is of rank 4, V carries an elliptic fibration. There are precisely 14, 98, 794, 6 710, 59 129, 532 460, 4 784 990, 43 049 510, and 387 374 024 points over $\mathbb{F}_3, \ldots, \mathbb{F}_{3^9}$. From these data, let us check whether one can prove $\mathrm{rk}\,\mathrm{Pic}(V_{\overline{\mathbb{F}}_3}) = 2$.

Assume that the characteristic polynomial of the Frobenius has more than two zeroes of the form 3 times a root of unity. Then, [6, Algorithm 22] leaves us with five polynomials $\Psi_1, \ldots, \Psi_5$,

$$\begin{aligned} \Psi_i(t) \;=\; & t^{22} - 4\,t^{21} + 27\,t^{18} + 81\,t^{17} - 243\,t^{16} + 6\,561\,t^{13} + a_1 t^{12} + b_1 t^{11} + c_1 t^{10} \\ & + (-1)^{j_i}[531\,441\,t^9 - 14\,348\,907\,t^6 + 43\,046\,721\,t^5 + 129\,140\,163\,t^4 \\ & \qquad - 13\,947\,137\,604\,t + 31\,381\,059\,609] \end{aligned}$$

for

$$\begin{aligned} j_1 = 0, \quad (a_i, b_i, c_i) &= (-59\,049,\ 236\,196, -531\,441)\,, \\ j_2 = 0, \quad (a_2, b_2, c_2) &= (\qquad 0, -118\,098, \qquad 0)\,, \\ j_3 = 0, \quad (a_3, b_3, c_3) &= (\ 19\,683, -236\,196,\ 177\,147)\,, \\ j_4 = 1, \quad (a_4, b_4, c_4) &= (-59\,049, \qquad 0,\ 531\,441)\,, \\ j_5 = 1, \quad (a_5, b_5, c_5) &= (-39\,366, \qquad 0,\ 354\,294)\,. \end{aligned}$$

Applying the Artin-Tate formula to these polynomials, we obtain the following data.

Table 3. Hypothetical ranks and discriminants

polynomial	field	arithmetic Picard rank	$\#\mathrm{Br}(V)\lvert\Delta\rvert$
Ψ_1	$\mathbb{F}_3$	2	24
	$\mathbb{F}_9$	4	1116
Ψ_2	$\mathbb{F}_3$	2	27
	$\mathbb{F}_9$	2	81
Ψ_3	$\mathbb{F}_3$	2	28
	$\mathbb{F}_9$	2	112
Ψ_4	$\mathbb{F}_3$	3	144
	$\mathbb{F}_9$	4	1152
Ψ_5	$\mathbb{F}_3$	1	2
	$\mathbb{F}_9$	2	65

Observe that an elliptic surface of Picard rank 2 automatically has a discriminant of the form $(-n^2)$ for n an integer. We may therefore exclude everything except for Ψ_4. Note that Ψ_2 is, in addition, incompatible with the field extension condition.

Thus, using the numbers of points over the fields up to $\mathbb{F}_{3^9}$, we only obtain that, either the geometric Picard rank is equal to 2, or Ψ_4 is the characteristic polynomial of the Frobenius in which case it is 4.

Example 30 (continuation). The number of points over $\mathbb{F}_{3^{10}}$ is 34 871 648 631. This additional information reproduces Ψ_1 and Ψ_4 as possible characteristic polynomials of Frob. Consequently, the minus sign holds in the functional equation and the geometric Picard rank of V is equal to 4.

8 Statistics

We tested the Artin-Tate conditions on samples of $K3$ surfaces of degrees 2, 4, 6, and 8. The possibilities of computing are limited by the fact that point counting over large finite fields is slow. In degree 2, decoupling [6, Algorithm 17] (see also [5]) leads to a substantial speed-up. In higher degrees, one may focus on elliptic $K3$ surfaces and exploit the fact that point counting on the elliptic fibers is fast. The numbers and particularities of the examples treated are listed in Table 4.

Table 4. Numbers of examples computed

	$p=2$	$p=3$	$p=5$	$p=7$
$d=2$	1000 rand	1000 rand	1000 dec	1000 dec
$d=4$	1000 rand	1000 ell		
$d=6$	1000 rand	1000 ell		
$d=8$	1000 rand	1000 ell		

dec = decoupled, ell = elliptic, rand = random

The remaining parameters of the surfaces were chosen by a random number generator. We stored the equations and the numbers of points over $\mathbb{F}_p, \ldots, \mathbb{F}_{p^{10}}$ in a file.

Results I. Point counting until $\mathbb{F}_{p^9}$. First, we tried to show that the geometric Picard-rank was equal to 2 only using the numbers of rational points over $\mathbb{F}_p, \ldots, \mathbb{F}_{p^9}$. I.e., we applied [6, Algorithm 22]. This algorithm produces a list of hypothetical Weil polynomials for each surface. If one is able to exclude all of them then, relative to the Tate conjecture, rank 2 is proven. To exclude a particular polynomial, we first checked whether the roots are of absolute value p. When a surface was known to be elliptic over $\mathbb{F}_p$, we checked in addition that the predicted Picard rank over $\mathbb{F}_p$ was at least equal to 2.

Then, we applied the Artin-Tate conditions to the polynomials. We checked the field extension condition and the rank-1 condition. For surfaces known to be elliptic over $\mathbb{F}_p$, we observed the fact that arithmetic Picard rank 2 forces the discriminant to be minus a perfect square. The results are summarized in Table 5.

Table 5. Distribution of the remaining hypothetical characteristic polynomials

	Number of polynomials	0	1	2	3	4	5	6
$d = 2$, $p = 2$	without	84	479	312	89	21	12	3
	with A-T conditions	149	598	218	28	7	0	0
$d = 2$, $p = 3$	without	116	480	285	88	24	4	3
	with A-T conditions	214	573	193	20	0	0	0
$d = 2$, $p = 5$	without	85	581	209	96	25	4	0
	with A-T conditions	158	651	169	20	2	0	0
$d = 2$, $p = 7$	without	92	534	232	98	37	7	0
	with A-T conditions	214	611	154	21	0	0	0
$d = 4$, $p = 2$	without	40	532	303	87	29	8	1
	with A-T conditions	81	638	249	27	5	0	0
$d = 4$, $p = 3$	without	22	669	242	57	9	1	0
	with A-T conditions	53	785	161	1	0	0	0
$d = 6$, $p = 2$	without	39	549	312	70	22	6	2
	with A-T conditions	83	645	257	14	1	0	0
$d = 6$, $p = 3$	without	16	713	217	47	7	0	0
	with A-T conditions	50	797	148	5	0	0	0
$d = 8$, $p = 2$	without	25	657	268	38	8	4	0
	with A-T conditions	29	723	239	5	4	0	0
$d = 8$, $p = 3$	without	12	720	236	27	4	1	0
	with A-T conditions	20	803	175	2	0	0	0

Results II. Point counting until $\mathbb{F}_{p^{10}}$. Using data up to $\mathbb{F}_{p^{10}}$, one obtains two hypothetical Weil polynomials for each of the surfaces. The two polynomials correspond to the possible signs in the functional equation (1). One has to exclude one of them. For this, we first checked the absolute values of the roots. For surfaces known to be elliptic over $\mathbb{F}_p$, we then tested whether the predicted arithmetic Picard rank is at least 2. Then, we applied the Artin-Tate conditions. We checked the field extensions and the rank-1 condition. For elliptic surfaces, supposed to be of arithmetic Picard rank 2, we tested, in addition, whether the predicted discriminant was minus a square.

Table 6 shows the number of surfaces with known signs. In the case that the sign is not known, we computed the numbers of points predicted over further extensions of $\mathbb{F}_p$. Comparing these numbers for both hypothetical polynomials indicates whether further point counting would lead to a decision of the sign. We count how often which fields had to be considered in order to decide the sign.

Table 6. Sign decision in the functional equation

p	2	3	5	7	2	3	2	3	2	3
d	2	2	2	2	4	4	6	6	8	8
Known signs without A-T	768	843	864	869	761	876	790	888	822	897
Known signs using A-T	863	940	940	961	863	943	868	933	867	944
Remaining unknown signs	137	60	60	39	137	57	132	67	133	56
Data up to $\mathbb{F}_{p^{11}}$ insufficient	84	23	15	12	69	19	77	25	72	21
Data up to $\mathbb{F}_{p^{12}}$ insufficient	41	11	2	1	39	3	42	11	47	7
Data up to $\mathbb{F}_{p^{13}}$ insufficient	22	5	1	0	24	2	20	2	24	2
Data up to $\mathbb{F}_{p^{14}}$ insufficient	13	2	0	0	12	0	13	1	8	0
Data up to $\mathbb{F}_{p^{15}}$ insufficient	7	0	0	0	8	0	7	0	5	0
Data up to $\mathbb{F}_{p^{16}}$ insufficient	4	0	0	0	3	0	2	0	4	0
Data up to $\mathbb{F}_{p^{17}}$ insufficient	4	0	0	0	2	0	2	0	0	0
Data up to $\mathbb{F}_{p^{18}}$ insufficient	4	0	0	0	0	0	1	0	0	0
Data up to $\mathbb{F}_{p^{19}}$ insufficient	2	0	0	0	0	0	1	0	0	0
Data up to $\mathbb{F}_{p^{20}}$ insufficient	0	0	0	0	0	0	0	0	0	0

Using these data, we repeated our attempt to prove that the geometric Picard rank is equal to 2. More precisely, we checked whether only two roots of the characteristic polynomial are of the form p times a root of unity. The numbers of surfaces for which we succeeded are listed in Table 7.

Table 7. Numbers of rank-2 cases using $\mathbb{F}_{p^{10}}$-data

		rank 2 proven	rank 2 possible
$p = 2$, $d = 2$	without	271	330
	with A-T conditions	278	301
$p = 3$, $d = 2$	without	397	460
	with A-T conditions	409	428
$p = 5$, $d = 2$	without	353	425
	with A-T conditions	360	382
$p = 7$, $d = 2$	without	460	511
	with A-T conditions	464	476
$p = 2$, $d = 4$	without	132	197
	with A-T conditions	138	163
$p = 3$, $d = 4$	without	79	114
	with A-T conditions	79	81
$p = 2$, $d = 6$	without	145	183
	with A-T conditions	152	163
$p = 3$, $d = 6$	without	74	101
	with A-T conditions	74	81
$p = 2$, $d = 8$	without	65	93
	with A-T conditions	65	74
$p = 3$, $d = 8$	without	23	47
	with A-T conditions	23	25

Conclusion. The Artin-Tate conditions usually halve the number of cases with unknown signs. Furthermore, they double the number of cases where geometric Picard rank 2 may be proven only using data up to $\mathbb{F}_{p^9}$. Comparing Table 5 with Table 7, we see, however, that still only about one half of the cases with Picard rank 2 may be detected when counting until $\mathbb{F}_{p^9}$.

Remark 31. Let us finally mention that the Artin-Tate conditions came to us as a big surprise. It is astonishing that the Artin-Tate formula may be incompatible with itself under field extensions. Thus, it seems not entirely unlikely that there are even more constraints and one can still do better.

References

1. Artin, M., Swinnerton-Dyer, S.P.: The Shafarevich-Tate conjecture for pencils of elliptic curves on $K3$ surfaces. Invent. Math. 20, 249–266 (1973)
2. Beauville, A.: Surfaces algébriques complexes, Astérisque 54, Société Mathématique de France, Paris (1978)
3. Deligne, P.: La conjecture de Weil I. Publ. Math. IHES 43, 273–307 (1974)
4. Deligne, P.: Relèvement des surfaces $K3$ en caractéristique nulle. In: Prepared for publication by Luc Illusie, Algebraic surfaces (Orsay 1976–78). LNM, vol. 868, pp. 58–79. Springer, Berlin (1981)
5. Elsenhans, A.-S., Jahnel, J.: The Asymptotics of Points of Bounded Height on Diagonal Cubic and Quartic Threefolds. In: Algorithmic Number Theory (ANTS 7), pp. 317–332. Springer, Berlin (2006)
6. Elsenhans, A.S., Jahnel, J.: $K3$ surfaces of Picard rank one and degree two. In: Algorithmic Number Theory (ANTS 8), pp. 212–225. Springer, Berlin (2008)
7. Elsenhans, A.S., Jahnel, J.: On the computation of the Picard group for $K3$ surfaces (2009) (preprint)
8. Grothendieck, A.: Le groupe de Brauer, III: Exemples et compléments. In: Grothendieck, A. (ed.) Dix exposés sur la Cohomologie des schémas, pp. 88–188. North-Holland, Amsterdam (1968)
9. Honda, T.: Isogeny classes of abelian varieties over finite fields. J. Math. Soc. Japan 20, 83–95 (1968)
10. van Luijk, R.: $K3$ surfaces with Picard number one and infinitely many rational points. Algebra & Number Theory 1, 1–15 (2007)
11. Milne, J.S.: On a conjecture of Artin and Tate. Ann. of Math. 102, 517–533 (1975)
12. Milne, J.S.: Duality in the flat cohomology of a surface. Ann. Sci. École Norm. Sup., 4^e série 9, 171–201 (1976)
13. Milne, J.S.: Étale Cohomology. Princeton University Press, Princeton (1980)
14. Motose, K.: On values of cyclotomic polynomials. VIII. Bull. Fac. Sci. Technol. Hirosaki Univ. 9, 15–27 (2006)
15. Nygaard, N.O.: The Tate conjecture for ordinary $K3$ surfaces over finite fields. Invent. Math. 74, 213–237 (1983)
16. Nygaard, N.O., Ogus, A.: Tate's conjecture for $K3$ surfaces of finite height. Ann. of Math. 122, 461–507 (1985)
17. Tate, J.: Conjectures on algebraic cycles in l-adic cohomology. In: Motives, Proc. Sympos. Pure Math., vol. 55(1), pp. 71–83. Amer. Math. Soc., Providence (1994)
18. Zarhin, Y.I.: Transcendental cycles on ordinary $K3$ surfaces over finite fields. Duke Math. J. 72, 65–83 (1993)
19. Zarhin, Y.I.: The Brauer group of an abelian variety over a finite field. Izv. Akad. Nauk SSSR Ser. Mat. 46, 211–243 (1982) (Russian)
20. Zeilberger, D.: A combinatorial proof of Newtons's identities. Discrete Math. 49, 319 (1984)

Class Invariants by the CRT Method

Andreas Enge[1] and Andrew V. Sutherland[2]

[1] INRIA Bordeaux–Sud-Ouest, France
[2] Massachusetts Institute of Technology, Cambridge, MA 02139, USA

Abstract. We adapt the CRT approach for computing Hilbert class polynomials to handle a wide range of class invariants. For suitable discriminants D, this improves its performance by a large constant factor, more than 200 in the most favourable circumstances. This has enabled record-breaking constructions of elliptic curves via the CM method, including examples with $|D| > 10^{15}$.

1 Introduction

Every ordinary elliptic curve E over a finite field $\mathbb{F}_q$ has *complex multiplication* by an imaginary quadratic order $\mathcal{O}$, by which we mean that the endomorphism ring $\mathrm{End}(E)$ is isomorphic to $\mathcal{O}$. The Deuring lifting theorem implies that E is the reduction of an elliptic curve $\hat{E}/\mathbb{C}$ that also has complex multiplication by $\mathcal{O}$. Let K denote the fraction field of $\mathcal{O}$. The j-invariant of $\hat{E}$ is an algebraic integer whose minimal polynomial over K is the *Hilbert class polynomial* H_D, where D is the discriminant of $\mathcal{O}$. Notably, the polynomial H_D actually lies in $\mathbb{Z}[X]$, and its splitting field is the *ring class field* $K_{\mathcal{O}}$ for the order $\mathcal{O}$.

Conversely, an elliptic curve $E/\mathbb{F}_q$ with complex multiplication by $\mathcal{O}$ exists whenever q satisfies the norm equation $4q = t^2 - v^2 D$, with $t, v \in \mathbb{Z}$ and $t \not\equiv 0$ modulo the characteristic of $\mathbb{F}_q$. In this case H_D splits completely over $\mathbb{F}_q$, and its roots are precisely the j-invariants of the elliptic curves $E/\mathbb{F}_q$ that have complex multiplication by $\mathcal{O}$. Such a curve has $q + 1 \pm t$ points, where t is determined, up to a sign, by the norm equation. With a judicious selection of D and q one may obtain a curve with prescribed order. This is known as the *CM method*.

The main challenge for the CM method is to obtain the polynomial H_D, which has degree equal to the class number $h(D)$, and total size $O(|D|^{1+\epsilon})$. There are three approaches to computing H_D, all of which, under reasonable assumptions, can achieve a running time of $O(|D|^{1+\epsilon})$. These include the complex analytic method [12], a p-adic algorithm [9, 7], and an approach based on the Chinese Remainder Theorem (CRT) [2]. The first is the most widely used, and it is quite efficient; the range of discriminants to which it may be applied is limited not by its running time, but by the space required. The polynomial H_D is already likely to exceed available memory when $|D| > 10^9$, hence one seeks to apply the CM method to alternative class polynomials that have smaller coefficients than H_D. This makes computations with $|D| > 10^{10}$ feasible.

Recently, a modified version of the CRT approach was proposed that greatly reduces the space required for the CM method [30]. Under the Generalised Riemann Hypothesis (GRH), this algorithm is able to compute $H_D \bmod P$ using

G. Hanrot, F. Morain, and E. Thomé (Eds.): ANTS-IX 2010, LNCS 6197, pp. 142–156, 2010.

$O(|D|^{1/2+\epsilon} \log P)$ space and $O(|D|^{1+\epsilon})$ time. (Here and in the following, all complexity estimates refer to bit operations.) The reduced space complexity allows it to handle much larger discriminants, including examples with $|D| > 10^{13}$.

An apparent limitation of the CRT approach is that it depends on some specific features of the j-function. As noted in [2], this potentially precludes it from computing class polynomials other than H_D. The purpose of the present article is to show how these obstructions may be overcome, allowing us to apply the CRT method to many functions other than j, including two infinite families.

Subject to suitable constraints on D, we may then compute a class polynomial with smaller coefficients than H_D (by a factor of up to 72), and, in certain cases, with smaller degree (by a factor of 2). Remarkably, the actual running time with the CRT method is typically *better* than the size difference would suggest. Fewer CRT moduli are needed, and we may choose a subset for which the computation is substantially faster than on average.

We start §2 with a brief overview of the CRT method, and then describe a new technique to improve its performance, which also turns out to be crucial for certain class invariants. After discussing families of invariants in §3, we consider CRT-based approaches applicable to the different families and give a general algorithm in §4. Computational results and performance data appear in §5.

2 Hilbert Class Polynomials via the CRT

2.1 The Algorithm of Belding, Bröker, Enge, Lauter and Sutherland

The basic idea of the CRT-based algorithm for Hilbert class polynomials is to compute H_D modulo many small primes p, and then lift its coefficients by Chinese remaindering to integers, or to their reductions modulo a large (typically prime) integer P, via the explicit CRT [4, Thm. 3.1]. The latter approach suffices for most applications, and while it does not substantially reduce the running time (the same number of small primes is required), it can be accomplished using only $O(|D|^{1/2+\epsilon} \log P)$ space with the method of [30, §6].

For future reference, we summarise the algorithm to compute $H_D \bmod p$ for a prime p that splits completely in the ring class field $K_{\mathcal{O}}$. Let $h = h(D)$.

Algorithm 1 (Computing $H_D \bmod p$)

1. *Find the j-invariant j_1 of an elliptic curve $E/\mathbb{F}_p$ with* $\mathrm{End}(E) \cong \mathcal{O}$.
2. *Enumerate the other roots $j_2, \ldots, j_h$ of $H_D \bmod p$.*
3. *Compute* $H_D(X) \bmod p = (X - j_1) \cdots (X - j_h)$.

The first step is achieved by varying j_1 (systematically or randomly) over the elements of $\mathbb{F}_p$ until it corresponds to a suitable curve; details and many practical improvements are given in [2, 30]. The third step is a standard building block of computer algebra. Our interest lies in Step 2.

2.2 Enumerating the Roots of H_D mod p

The key idea in [2] leading to a quasi-linear complexity is to apply the Galois action of $\mathrm{Cl}(\mathcal{O}) \simeq \mathrm{Gal}(K_{\mathcal{O}}/K)$. The group $\mathrm{Cl}(\mathcal{O})$ acts on the roots of H_D, and when p splits completely in $K_{\mathcal{O}}$ there is a corresponding action on the set $\mathrm{Ell}_{\mathcal{O}}(\mathbb{F}_p) = \{j_1, \ldots, j_h\}$ containing the roots of H_D mod p. For an ideal class $[\mathfrak{a}]$ in $\mathrm{Cl}(\mathcal{O})$ and a j-invariant $j_i \in \mathrm{Ell}_{\mathcal{O}}(\mathbb{F}_p)$, let us write $[\mathfrak{a}]j_i$ for the image of j_i under the Galois action of $[\mathfrak{a}]$. We then have $\mathrm{Ell}_{\mathcal{O}}(\mathbb{F}_p) = \{[\mathfrak{a}]j_1 : [\mathfrak{a}] \in \mathrm{Cl}(\mathcal{O})\}$.

As in [30, §5], we use a polycyclic presentation defined by a sequence of ideals $\mathfrak{l}_1, \ldots, \mathfrak{l}_m$ with prime norms $\ell_1, \ldots, \ell_m$ whose classes generate $\mathrm{Cl}(\mathcal{O})$. The *relative order* r_k is the least positive integer for which $[\mathfrak{l}_k^{r_k}] \in \langle [\mathfrak{l}_1], \ldots, [\mathfrak{l}_{k-1}] \rangle$. We may then uniquely write $[\mathfrak{a}] = [\mathfrak{l}_1^{e_1}] \cdots [\mathfrak{l}_m^{e_m}]$, with $0 \le e_k < r_k$. To maximise performance, we use a presentation in which $\ell_1 < \cdots < \ell_m$, with each ℓ_k as small as possible subject to $r_k > 1$. Note that the relative order r_k divides the order n_k of $[\mathfrak{l}_k]$ in $\mathrm{Cl}(\mathcal{O})$, but for $k > 1$ we can (and often do) have $r_k < n_k$.

For each $j_i \in \mathrm{Ell}_{\mathcal{O}}(\mathbb{F}_p)$ and each $\mathcal{O}$-ideal $\mathfrak{l}$ of prime norm ℓ, the j-invariant $[\mathfrak{l}]j_i$ corresponds to an ℓ-isogenous curve, which we may obtain as a root of $\Phi_\ell(j_i, X)$, where $\Phi_\ell \in \mathbb{Z}[J, J_\ell]$ is the *classical modular polynomial* [31, §69]. The polynomial Φ_ℓ has the pair of functions $\big(j(z), j(\ell z)\big)$ as roots, and parameterises isogenies of degree ℓ.

Fixing an isomorphism $\mathrm{End}(E) \cong \mathcal{O}$, we let $\pi \in \mathcal{O}$ denote the Frobenius endomorphism. When the order $\mathbb{Z}[\pi]$ is maximal at ℓ, the univariate polynomial $\Phi_\ell(j_i, X) \in \mathbb{F}_p[X]$ has exactly two roots $[\mathfrak{l}]j_i$ and $[\bar{\mathfrak{l}}]j_i$ when ℓ splits in $\mathcal{O}$, and a single root $[\mathfrak{l}]j_i$ if ℓ is ramified [25, Prop. 23]. To simplify matters, we assume here that $\mathbb{Z}[\pi]$ is maximal at each ℓ_k, but this is not necessary, see [30, §4].

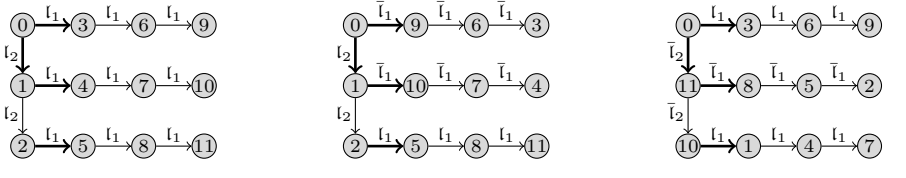
We may enumerate $\mathrm{Ell}_{\mathcal{O}}(\mathbb{F}_p) = \{[\mathfrak{a}]j_1 : [\mathfrak{a}] \in \langle [\mathfrak{l}_1], \ldots, [\mathfrak{l}_m] \rangle\}$ via [30, Alg. 1.3]:

Algorithm 2 (Enumerating $\mathrm{Ell}_{\mathcal{O}}(\mathbb{F}_p$) — Step 2 of Algorithm 1)

1. *Let j_2 be an arbitrary root of $\Phi_{\ell_m}(j_1, X)$ in $\mathbb{F}_p$.*
2. *For i from 3 to r_m, let j_i be the root of $\Phi_{\ell_m}(j_{i-1}, X)/(X - j_{i-2})$ in $\mathbb{F}_p$.*
3. *If $m > 1$, then for i from 1 to r_m:*
   Recursively enumerate the set $\{[\mathfrak{a}]j_i : [\mathfrak{a}] \in \langle [\mathfrak{l}_1], \ldots, [\mathfrak{l}_{m-1}] \rangle\}$.

In general there are two distinct choices for j_2, but either will do. Once j_2 is chosen, $j_3, \ldots, j_{r_m}$ are determined. The sequence $(j_1, \ldots, j_{r_m})$ corresponds to a path of ℓ_m-isogenies; we call this path an *ℓ_m-thread*.

The choice of j_2 in Step 1 may change the order in which $\mathrm{Ell}_{\mathcal{O}}(\mathbb{F}_p)$ is enumerated. Three of the sixteen possibilities when $m = 2$, $r_1 = 4$, and $r_2 = 3$ are shown below; we assume $[\mathfrak{l}_2^3] = [\mathfrak{l}_1]$, and label each vertex $[\mathfrak{l}_2^e]j_1$ by the exponent e.



Bold edges indicate where a choice was made. Regardless of these choices, Algorithm 2 correctly enumerates $\mathrm{Ell}_{\mathcal{O}}(\mathbb{F}_p)$ in every case [30, Prop. 5].

2.3 Finding Roots with Greatest Common Divisors (gcds)

The potentially haphazard manner in which Algorithm 2 enumerates $\mathrm{Ell}_{\mathcal{O}}(\mathbb{F}_p)$ is not a problem when computing H_D, but it can complicate matters when we wish to compute other class polynomials. We could distinguish the actions of $\mathfrak{l}$ and $\bar{\mathfrak{l}}$ using an Elkies kernel polynomial [10], as suggested in [7, §5], however this slows down the algorithm significantly. An alternative approach using polynomial gcds turns out to be much more efficient, and actually speeds up Algorithm 2, making it already a useful improvement when computing H_D.

We need not distinguish the actions of $\mathfrak{l}$ and $\bar{\mathfrak{l}}$ at this stage, but we wish to ensure that our enumeration of $\mathrm{Ell}_{\mathcal{O}}(\mathbb{F}_p)$ makes a consistent choice of direction each time it starts an ℓ-thread. The first ℓ-thread may be oriented arbitrarily, but for each subsequent ℓ-thread $(j'_1, j'_2, \ldots, j'_r)$, we apply Lemma 1 below. This allows us to "square the corner" by choosing j'_2 as the unique common root of $\Phi_\ell(X, j'_1)$ and $\Phi_{\ell'}(X, j_2)$, where $(j_1, \ldots, j_r)$ is a previously computed ℓ-thread and j_1 is ℓ'-isogenous to j'_1. The edge (j_1, j'_1) lies in an ℓ'-thread that has already been computed, for some $\ell' > \ell$.

Having computed j'_2, we could compute $j'_3, \ldots, j'_r$ as before, but it is usually better to continue using gcds, as depicted above. Asymptotically, both root-finding and gcd computations are dominated by the $O(\ell^2 \mathsf{M}(\log p))$ time it takes to instantiate $\Phi_\ell(X, j_i) \bmod p$, but in practice ℓ is small, and we effectively gain a factor of $O(\log p)$ by using gcds when $\ell \approx \ell'$. This can substantially reduce the running time of Algorithm 2, as may be seen in Table 1 of §5.

With the gcd approach described above, the total number of root-finding operations can be reduced from $\prod_{k=1}^{m} r_k$ to $\sum_{k=1}^{m} r_k$. When m is large, this is a big improvement, but it is no help when $m = 1$, as necessarily occurs when $h(D)$ is prime. However, even in this case we can apply gcds by looking for an auxiliary ideal $\mathfrak{l}'_1$, with prime norm ℓ'_1, for which $[\mathfrak{l}'_1] = [\mathfrak{l}_1^e]$. When r_1 is large, such an $\mathfrak{l}'_1$ is easy to find, and we may choose the best combination of ℓ'_1 and e available. This idea generalises to ℓ_k-threads, where we seek $[\mathfrak{l}'_k] \in \langle [\mathfrak{l}_1] \ldots, [\mathfrak{l}_k] \rangle \backslash \langle [\mathfrak{l}_1] \ldots, [\mathfrak{l}_{k-1}] \rangle$.

Lemma 1. *Let $j_1, j_2 \in \mathrm{Ell}_{\mathcal{O}}(\mathbb{F}_p)$, and let $\ell_1, \ell_2 \neq p$ be distinct primes with $4\ell_1^2\ell_2^2 < |D|$. Then $\gcd\big(\Phi_{\ell_1}(j_1, X), \Phi_{\ell_2}(j_2, X)\big)$ has degree at most 1.*

Proof. It follows from [25, Prop. 23] that $\Phi_{\ell_1}(X, j_1)$ and $\Phi_{\ell_2}(X, j_2)$ have at most two common roots in the algebraic closure $\overline{\mathbb{F}}_p$, which in fact lie in $\mathrm{Ell}_{\mathcal{O}}(\mathbb{F}_p)$. If there are exactly two, then both $\ell_1 = \mathfrak{l}_1\bar{\mathfrak{l}}_1$ and $\ell_2 = \mathfrak{l}_2\bar{\mathfrak{l}}_2$ split in $\mathcal{O}$, and one of $\mathfrak{l}_1^2\mathfrak{l}_2^2$ or $\mathfrak{l}_1^2\bar{\mathfrak{l}}_2^2$ is principal with a non-rational generator. We thus have a norm equation $4\ell_1^2\ell_2^2 = a^2 - b^2 D$ with $a, b \in \mathbb{Z}$ and $b \neq 0$, and the lemma follows.

3 Class Invariants

Due to the large size of H_D, much effort has been spent seeking smaller generators of $K_{\mathcal{O}}$. For a modular function f and $\mathcal{O} = \mathbb{Z}[\tau]$, with τ in the upper half plane, we call $f(\tau)$ a *class invariant* if $f(\tau) \in K_{\mathcal{O}}$. The *class polynomial* for f is

$$H_D[f](X) = \prod_{[\mathfrak{a}] \in \mathrm{Cl}(\mathcal{O})} \left(X - [\mathfrak{a}]f(\tau)\right).$$

The contemporary tool for determining class invariants is Shimura's reciprocity law; see [28, Th. 4] for a fairly general result. Class invariants arising from many different modular functions have been described in the literature; we briefly summarise some of the most useful ones.

Let η be Dedekind's function, and let $\zeta_n = \exp(2\pi i/n)$. Weber considered

$$\mathfrak{f} = \zeta_{48}^{-1} \frac{\eta\left(\frac{z+1}{2}\right)}{\eta(z)}, \qquad \mathfrak{f}_1(z) = \frac{\eta\left(\frac{z}{2}\right)}{\eta(z)}, \qquad \mathfrak{f}_2(z) = \sqrt{2}\,\frac{\eta(2z)}{\eta(z)},$$

powers of which yield class invariants when $\left(\frac{D}{2}\right) \neq -1$, and also $\gamma_2 = \sqrt[3]{j}$, which is a class invariant whenever $3 \nmid D$. The Weber functions can be generalised [15, 16, 21, 20, 23], and we have the simple and double η-quotients

$$\mathfrak{w}_N(z) = \frac{\eta\left(\frac{z}{N}\right)}{\eta(z)}; \qquad \mathfrak{w}_{p_1,p_2} = \frac{\eta\left(\frac{z}{p_1}\right)\eta\left(\frac{z}{p_2}\right)}{\eta\left(\frac{z}{p_1 p_2}\right)\eta(z)} \text{ with } N = p_1 p_2,$$

where p_1 and p_2 are primes. Subject to constraints on D, including that no prime dividing N is inert in $\mathcal{O}$, suitable powers of these functions yield class invariants, see [15, 16]. For $s = 24/\gcd\big(24, (p_1-1)(p_2-1)\big)$, the canonical power $\mathfrak{w}_{p_1,p_2}^s$ is invariant under the Fricke involution $W|_N : z \mapsto \frac{-N}{z}$ for $\Gamma^0(N)$, equivalently, the Atkin-Lehner involution of level N, by [17, Thm. 2].

The theory of [28] applies to any functions for $\Gamma^0(N)$, in particular to those of prime level N invariant under the Fricke involution, which yield class invariants when $\left(\frac{D}{N}\right) \neq -1$. Atkin developed a method to compute such functions A_N, which are conjectured to have a pole of minimal order at the unique cusp [10, 26]. These are used in the SEA algorithm, and can be found in Magma or Pari/GP.

The functions above all yield algebraic integers, so $H_D[f] \in \mathcal{O}_K[X]$. Except for $\mathfrak{w}_N^e$ or when $\gcd(N, D) \neq 1$, in which cases additional restrictions may apply, one actually has $H_D[f] \in \mathbb{Z}[X]$, cf. [16, Cor. 3.1]. The (logarithmic) *height* of $H_D[f] = \sum a_i X^i$ is $\log \max |a_i|$, which determines the precision needed to compute the a_i. We let $c_D(f)$ denote the ratio of the heights of $H_D[j]$ and $H_D[f]$.

With $c(f) = \lim_{|D|\to\infty} c_D(f)$, we have: $c(\gamma_2) = 3$; $c(\mathfrak{f}) = 72$ (when $\left(\frac{D}{2}\right) = 1$);

$$c(\mathfrak{w}_N^e) = \frac{24(N+1)}{e(N-1)}; \qquad c(\mathfrak{w}_{p_1,p_2}^s) = \frac{12\psi(p_1p_2)}{s(p_1-1)(p_2-1)}; \qquad c(A_N) = \frac{N+1}{2|v_N|},$$

where e divides the exponent s defined above, v_N is the order of the pole of A_N at the cusp, and $\psi(p_1p_2)$ is $(p_1+1)(p_2+1)$ when $p_1 \neq p_2$, and $p_1(p_1+1)$ when

$p_1 = p_2$. Morain observed in [27] that $c(A_{71}) = 36$, which is so far the best value known when $\left(\frac{D}{2}\right) = -1$. We conjecture that in fact for all primes $N > 11$ with $N \equiv 11 \bmod 60$ we have $c(A_N) = 30\frac{N+1}{N-11}$, and that for $N \equiv -1 \bmod 60$ we have $c(A_N) = 30$. This implies that given an arbitrary discriminant D, we can always choose N so that A_N yields class invariants with $c_D(A_N) \geq 30 + o(1)$.

When the prime divisors of N are all ramified in K, both $\mathfrak{w}_{p_1,p_2}$ and A_N yield class polynomials that are squares in $\mathbb{Z}[X]$, see [11, §1.6] and [18]. Taking the square root of such a class polynomial reduces both its degree and its height by a factor of 2. For a composite fundamental discriminant D (the most common case), this applies to $H_D[A_N]$ for any prime $N \mid D$. In the best case, D is divisible by 71, and we obtain a class polynomial that is 144 times smaller than H_D.

3.1 Modular Polynomials

Each function $f(z)$ considered above is related to $j(z)$ by a modular polynomial $\Psi_f \in \mathbb{Z}[F, J]$ satisfying $\Psi_f(f(z), j(z)) = 0$. For primes ℓ not dividing the level N, we let $\Phi_{\ell,f}$ denote the minimal polynomial satisfying $\Phi_{\ell,f}(f(z), f(\ell z)) = 0$; it is a factor of $\mathrm{Res}_{J_\ell}\big(\mathrm{Res}_J(\Phi_\ell(J, J_\ell), \Psi_f(F, J)), \Psi_f(F_\ell, J_\ell)\big)$, and as such, an element of $\mathbb{Z}[F, F_\ell]$. Thus $\Phi_{\ell,f}$ generalises the classical modular polynomial $\Phi_\ell = \Phi_{\ell,j}$.

The polynomial $\Phi_{\ell,f}$ has degree $d(\ell+1)$ in F and F_ℓ, where d divides $\deg_J \Psi_f$, see [6, §6.8], and $2d$ divides $\deg_J \Psi_f$ when f is invariant under the Fricke involution. In general, d is maximal, and $d = 1$ is achievable only in the relatively few cases where $X_0(N)$, respectively $X_0^+(N)$, is of genus 0 and, moreover, f is a hauptmodul, that is, it generates the function field of the curve. Happily, this includes many cases of practical interest.

The polynomial Ψ_f characterises the analytic function f in an algebraic way; when $d = 1$, the polynomials Φ_ℓ and $\Phi_{\ell,f}$ algebraically characterise ℓ-isogenies between elliptic curves given by their j-invariants, or by class invariants derived from f, respectively. These are key ingredients for the CRT method.

4 CRT Algorithms for Class Invariants

To adapt Algorithm 1 to class invariants arising from a modular function $f(z)$ other than $j(z)$, we only need to consider Algorithm 2. Our objective is to enumerate the roots of $H_D[f] \bmod p$ for suitable primes p, which we are free to choose. This may be done in one of two ways. The most direct approach computes an "f-invariant" f_1, corresponding to j_1, then enumerates $f_2, \ldots, f_h$ using the modular polynomials $\Phi_{\ell,f}$. Alternatively, we may enumerate $j_1, \ldots, j_h$ as before, and from these derive $f_1, \ldots, f_h$. The latter approach is not as efficient, but it applies to a wider range of functions, including two infinite families.

Several problems arise. First, an elliptic curve $E/\mathbb{F}_p$ with CM by $\mathcal{O}$ unambiguously defines a j-invariant $j_1 = j(E)$, but not the corresponding f_1. The f_1 we seek is a root of $\psi_f(X) = \Psi_f(X, j_1) \bmod p$, but ψ_f may have other roots, which may or may not be class invariants. The same problem occurs for the

p-adic lifting algorithm and can be solved generically [6, §6]; we describe some more efficient solutions, which are in part specific to certain types of functions.

When ψ_f has multiple roots that are class invariants, these may be roots of distinct class polynomials. We are generally happy to compute any one of these, but it is imperative that we compute the reduction of "the same" class polynomial $H_D[f]$ modulo each prime p.

The lemma below helps to address these issues for at least two infinite families of functions: the double η-quotients $\mathfrak{w}_{p_1,p_2}$ and the Atkin functions A_N.

Lemma 2. *Let f be a modular function for $\Gamma^0(N)$, invariant under the Fricke involution $W|_N$, such that $f(z)$ and $f\left(\frac{-1}{z}\right)$ have rational q-expansions. Let the imaginary quadratic order $\mathcal{O}$ have conductor coprime to N and contain an ideal $\mathfrak{n} = \left(N, \frac{B_0+\sqrt{D}}{2}\right)$. Let $A_0 = \frac{B_0^2-D}{4N}$ and $\tau_0 = \frac{-B_0+\sqrt{D}}{2A_0}$, and assume that $\gcd(A_0, N) = 1$. Then $f(\tau_0)$ is a class invariant, and if $f(\tau)$ is any of its conjugates under the action of $\mathrm{Gal}(K_\mathcal{O}/K)$ we have*

$$\Psi_f\big(f(\tau), j(\tau)\big) = 0 \qquad \text{and} \qquad \Psi_f\big(f(\tau), [\mathfrak{n}]j(\tau)\big) = 0.$$

Proof. By definition, $\Psi_f\big(f(z), j(z)\big) = 0$. Applying the Fricke involution yields $0 = \Psi_f\left((W|_N f)(z), (W|_N j)(z)\right) = \Psi_f\left(f(z), j\left(\frac{-N}{z}\right)\right) = \Psi_f\left(f(z), j\left(\frac{z}{N}\right)\right)$. The value $f(\tau_0)$ is a class invariant by [28, Th. 4]. By the same result, we may assume that τ is the basis quotient of an ideal $\mathfrak{a} = \left(A, \frac{-B+\sqrt{D}}{2}\right)$ with $\gcd(A, N) = 1$ and $B \equiv B_0 \bmod 2N$. Then $\frac{\tau}{N}$ is the basis quotient of $\mathfrak{a}\overline{\mathfrak{n}} = \left(AN, \frac{-B+\sqrt{D}}{2}\right)$. It follows that $[\mathfrak{n}]j(\tau) = j\left(\frac{\tau}{N}\right)$, and replacing z above by τ completes the proof.

If we arrange the roots of H_D into a graph of $\mathfrak{n}$-isogeny cycles corresponding to the action of $\mathfrak{n}$, the lemma yields a dual graph defined on the roots of $H_D[f]$, in which vertices $f(\tau)$ correspond to edges $\big(j(\tau), [\mathfrak{n}]j(\tau)\big)$.

In computational terms, $f(\tau)$ is a root of $\gcd\big(\Psi_f\big(X, j(\tau)\big), \Psi_f\big(X, [\mathfrak{n}]j(\tau)\big)\big)$. Generically, we expect this gcd to have no other roots modulo primes p that split completely in $K_\mathcal{O}$. For a finite number of such primes, there may be additional roots. We have observed this for p dividing the conductor of the order generated by $f(\tau)$ in the maximal order of $K_\mathcal{O}$. Such primes may either be excluded from our CRT computations, or addressed by one of the techniques described in §4.3.

4.1 Direct Enumeration

When the polynomials $\Phi_{\ell,f}$ have degree $\ell + 1$ we can apply Algorithm 2 with essentially no modification; the only new consideration is that ℓ must not divide the level N, but we can exclude such ℓ when choosing a polycyclic presentation for $\mathrm{Cl}(\mathcal{O})$. When the degree is greater than $\ell + 1$ the situation is more complex, moreover the most efficient algorithms for computing modular polynomials do not apply [8, 13], making it difficult to obtain $\Phi_{\ell,f}$ unless ℓ is very small. Thus in practice we do not use $\Phi_{\ell,f}$ in this case; instead we apply the methods of §4.3 or §4.4. For the remainder of this subsection and the next we assume that we do have polynomials $\Phi_{\ell,f}$ of degree $\ell + 1$ with which to enumerate $f_1, \ldots, f_h$, and

consider how to determine a starting point f_1, given the j-invariant $j_1 = j(E)$ of an elliptic curve $E/\mathbb{F}_p$ with CM by $\mathcal{O}$.

When $\psi_f(X) = \Psi_f(X, j_1) \bmod p$ has only one root, our choice of f_1 is immediately determined. This is usually not the case, but we may be able to ensure it by restricting our choice of p. As an example, for $f = \gamma_2$ with $3 \nmid D$, if we require that $p \equiv 2 \bmod 3$, then f_1 is the unique cube root of j_1 in $\mathbb{F}_p$. If we additionally have $D \equiv 1 \bmod 8$ and $p \equiv 3 \bmod 4$, then the equation $\gamma_2 = (\mathfrak{f}^{24} - 16)/\mathfrak{f}^8$ uniquely determines the square of the Weber $\mathfrak{f}$ function, by [8, Lem. 7.3]. To treat $\mathfrak{f}$ itself we need an additional trick described in §4.2.

The next simplest case occurs when only one of the roots of ψ_f is a class invariant. This necessarily happens when f is invariant under the Fricke involution and all the primes dividing N are ramified in $\mathcal{O}$. In the context of Lemma 2, each root of $H_D[f]$ then corresponds to an isolated edge $\big(j(\tau), [\mathfrak{n}]j(\tau)\big)$ in the $\mathfrak{n}$-isogeny graph on the roots of H_D, and we compute f_1 as the unique root of $\gcd\big(\Psi_f(X, j_1), \Psi_f(X, [\mathfrak{n}]j_1)\big)$. In this situation $\mathfrak{n} = \bar{\mathfrak{n}}$, and each $f(\tau)$ occurs twice as a root of $H_D[f]$. By using a polycyclic presentation for $\mathrm{Cl}(\mathcal{O})/\langle[\mathfrak{n}]\rangle$ rather than $\mathrm{Cl}(\mathcal{O})$, we enumerate each double root of $H_D[f] \bmod p$ just once.

Even when ψ_f has multiple roots that are class invariants, it may happen that they are all roots of the *same* class polynomial. This applies to the Atkin functions $f = A_N$. When N is a split prime, there are two N-isogenous pairs $(j_1, [\mathfrak{n}]j_1)$ and $([\bar{\mathfrak{n}}]j_1, j_1)$ in $\mathrm{Ell}_{\mathcal{O}}(\mathbb{F}_p)$, and under Lemma 2 these correspond to roots f_1 and $[\bar{\mathfrak{n}}]f_1$ of ψ_f. Both are roots of $H_D[f]$, and we may choose either.

The situation is slightly more complicated for the double η-quotients $\mathfrak{w}_{p_1,p_2}$, with $N = p_1p_2$ composite. If $p_1 = \mathfrak{p}_1\bar{\mathfrak{p}}_1$ and $p_2 = \mathfrak{p}_2\bar{\mathfrak{p}}_2$ both split and $p_1 \neq p_2$, then there are four distinct N-isogenies corresponding to four roots of ψ_f. Two of these roots are related by the action of $[\mathfrak{n}] = [\mathfrak{p}_1\mathfrak{p}_2]$; they belong to the same class polynomial, which we choose as $H_D[f] \bmod p$. The other two are related by $[\mathfrak{p}_1\bar{\mathfrak{p}}_2]$ and are roots of a different class polynomial. We make an arbitrary choice for f_1, explicitly compute $[\mathfrak{n}]f_1$, and then check whether it occurs among the other three roots; if not, we correct the initial choice. The techniques of §4.3 may be used to efficiently determine the action of $[\mathfrak{n}]$.

Listed below are some of the modular functions f for which the roots of $H_D[f] \bmod p$ may be directly enumerated, with sufficient constraints on D and p. In each case p splits completely in $K_{\mathcal{O}}$ and $D < -4N^2$ has conductor u.

(1) γ_2, with $3 \nmid D$ and $p \equiv 2 \bmod 3$;

(2) $\mathfrak{f}^2$, with $D \equiv 1 \bmod 8$, $3 \nmid D$, and $p \equiv 11 \bmod 12$;

(3) $\mathfrak{w}_N^s$, for $N \in \{3, 5, 7, 13\}$ and $s = 24/\gcd(24, N-1)$, with $N \mid D$ and $N \nmid u$;

(4) $\mathfrak{w}_5^2$, with $3 \nmid D$, $5 \mid D$, and $5 \nmid u$;

(5) A_N, for $N \in \{3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 41, 47, 59, 71\}$, with $\left(\frac{D}{N}\right) \neq -1$ and $N \nmid u$.

(6) $\mathfrak{w}_{p_1,p_2}^s$, for $(p_1, p_2) \in \{(2,3), (2,5), (2,7), (2,13), (3,5), (3,7), (3,13), (5,7)\}$ and $s = 24/\gcd\big(24, (p_1-1)(p_2-1)\big)$, with $\left(\frac{D}{p_1}\right), \left(\frac{D}{p_2}\right) \neq -1$ and $p_1, p_2 \nmid u$.

(7) $\mathfrak{w}_{3,3}^6$ with $\left(\frac{D}{3}\right) = 1$ and $3 \nmid u$.

4.2 The Trace Trick

In §4.1 we were able to treat the square of the Weber $\mathfrak{f}$ function but not $\mathfrak{f}$ itself. To remedy this, we generalise a method suggested to us by Reinier Bröker.

We consider the situation where there are two modular functions f and f' that are roots of $\Psi_f(X, j(z))$, both of which yield class invariants for $\mathcal{O}$, and we wish to apply the direct enumeration approach. We assume that p is chosen so that $\psi_f(X) = \Psi_f(X, j_1) \bmod p$ has exactly two roots, and depending on which root we take as f_1, we may compute the reduction of either $H_D[f](X)$ or $H_D[f'](X)$ modulo p. In the case of Weber $\mathfrak{f}$, we have $f' = -f$, and $H_D[f']$ differs from $H_D[f]$ only in the sign of every other coefficient.

Consider a fixed coefficient a_i of $H_D[f](X) = \sum a_i X^i$; most of the time, the trace $t = -a_{h-1} = f_1 + \cdots + f_h$ will do (if $f' = -f$, we need to use a_i with $i \not\equiv h \bmod 2$). The two roots f_1 and f_1' lead to two possibilities t and t' modulo p. However, the elementary symmetric functions $T_1 = t + t'$ and $T_2 = tt'$ are unambiguous modulo p. Computing these modulo many primes p yields T_1 and T_2 as integers (via the CRT), from which t and t' are obtained as roots of the quadratic equation $X^2 - T_1 X + T_2$. If these are different, we arbitrarily pick one of them, which, going back, determines the set of conjugates $\{f_1, \ldots, f_h\}$ or $\{f_1', \ldots, f_h'\}$ to take modulo each of the primes $p \nmid t - t'$. In the unlikely event that they are the same (the suspicion $t = t'$ being confirmed after, say, looking at the second prime), we need to switch to a different coefficient a_i.

If f and f' differ by a simple transformation (such as $f' = -f$), the second set of conjugates and the value t' are obtained essentially for free. As a special case, when h is odd and the class invariants are units (as with Weber $\mathfrak{f}$), we can simply fix $t = a_0 = 1$, and need not compute $T_1 = 0$ and $T_2 = -1$.

The key point is that the number of primes p we use to determine t is much less than the number of primes we use to compute $H_D[f]$. Asymptotically, the logarithmic height of the trace is smaller than the height bound we use for $H_D[f]$ by a factor quasi-linear in $\log|D|$, under the GRH. In practical terms, determining t typically requires less than one tenth of the primes used to compute $H_D[f]$, and these computations can be combined.

The approach described above generalises immediately to more than two roots, but this case does not occur for the functions we examine. Unfortunately it can be used only in conjunction with the direct enumeration approach of §4.1; otherwise we would have to consistently distinguish not only between f_1 and f_1', but also between f_i and f_i' for $i = 2, \ldots, h$.

4.3 Enumeration via the Fricke Involution

For functions f to which Lemma 2 applies, we can readily obtain the roots of $H_D[f] \bmod p$ without using the polynomials $\Phi_{\ell,f}$. We instead enumerate the roots of $H_D \bmod p$ (using the polynomials Φ_ℓ), and arrange them into a graph G of $\mathfrak{n}$-isogeny cycles, where $\mathfrak{n}$ is the ideal of norm N appearing in Lemma 2. We then obtain roots of $H_D[f] \bmod p$ by computing $\gcd\big(\Psi_f(X, j_i), \Psi_f(X, [\mathfrak{n}]j_i)\big)$ for each edge $(j_i, [\mathfrak{n}]j_i)$ in G.

The graph G is composed of h/n cycles of length n, where n is the order of $[\mathfrak{n}]$ in $\mathrm{Cl}(\mathcal{O})$. We assume that the $\mathcal{O}$-ideals of norm N are all non-principal and inequivalent (by requiring $|D| > 4N^2$ if needed). When every prime dividing N is ramified in $\mathcal{O}$ we have $n = 2$; as noted in §4.1, every root of $H_D[f]$ then occurs with multiplicity 2, and we may compute the square-root of $H_D[f]$ by taking each root just once. Otherwise we have $n > 2$.

Let $[\mathfrak{l}_1], \ldots, [\mathfrak{l}_m]$ be a polycyclic presentation for $\mathrm{Cl}(\mathcal{O})$ with relative orders $r_1, \ldots, r_m$, as in §2.2. For k from 1 to m let us fix $\mathfrak{l}_k = \left(\ell_k, \frac{-B_k+\sqrt{D}}{2}\right)$ with $B_k \geq 0$. To each vector $\boldsymbol{e} = (e_1, \ldots, e_m)$ with $0 \leq e_k < r_k$, we associate a unique root $j_{\boldsymbol{e}}$ enumerated by Algorithm 2, corresponding to the path taken from j_1 to $j_{\boldsymbol{e}}$, where e_k counts steps taken along an ℓ_k-thread. For $\boldsymbol{o} = (0, \ldots, 0)$ we have $j_{\boldsymbol{o}} = j_1$, and in general

$$j_{\boldsymbol{e}} = [\mathfrak{l}_1^{\sigma_1 e_1} \cdots \mathfrak{l}_m^{\sigma_m e_m}] j_{\boldsymbol{o}},$$

with $\sigma_k = \pm 1$. Using the method of §2.3 to consistently orient the ℓ_k-threads ensures that each σ_k depends only on the orientation of the first ℓ_k-thread.

To compute the graph G we must determine the signs σ_k. For those $[\mathfrak{l}_k]$ of order 2, we let $\sigma_k = 1$. We additionally fix $\sigma_k = 1$ for the least $k = k_0$ (if any) for which $[\mathfrak{l}_k]$ has order greater than 2, since we need not distinguish the actions of $\mathfrak{n}$ and $\bar{\mathfrak{n}}$. It suffices to show how to determine σ_k, given that we know $\sigma_1, \ldots, \sigma_{k-1}$. We may assume $[\mathfrak{l}_{k_0}]$ and $[\mathfrak{l}_k]$ both have order greater than 2, with $k_0 < k \leq m$.

Let $\mathfrak{l}$ be an auxiliary ideal of prime norm ℓ such that $[\mathfrak{l}] = [\mathfrak{a}\mathfrak{b}] = [\mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_k^{e_k}]$, with $0 \leq e_i < r_i$, where $\mathfrak{b} = \mathfrak{l}_k^{e_k}$, and $[\mathfrak{a}]$ and $[\mathfrak{b}]$ have order greater than 2. Our assumptions guarantee that such an $\mathfrak{l}$ exists, by the Čebotarev density theorem, and under the GRH, ℓ is relatively small [1]. The fact that $[\mathfrak{a}]$ and $[\mathfrak{b}]$ have order greater than 2 ensures that $[\mathfrak{a}\bar{\mathfrak{b}}]$ is distinct from $[\mathfrak{l}]$ and its inverse. It follows that $\sigma_k = 1$ if and only if $\Phi_\ell(j_{\boldsymbol{o}}, j_{\boldsymbol{e}}) = 0$, where $\boldsymbol{e} = (e_1, \ldots, e_k, 0, \ldots, 0)$.

Having determined the σ_k, we compute the unique vector $\boldsymbol{v} = (v_1, \ldots, v_m)$ for which $[\mathfrak{n}] = [\mathfrak{l}_1^{\sigma_1 v_1} \cdots \mathfrak{l}_m^{\sigma_m v_m}]$. We then have $[\mathfrak{n}] j_{\boldsymbol{o}} = j_{\boldsymbol{v}}$, yielding the edge $(j_{\boldsymbol{o}}, j_{\boldsymbol{v}})$ of G. In general, we obtain the vector corresponding to $[\mathfrak{n}] j_{\boldsymbol{e}}$ by computing $\boldsymbol{e} + \boldsymbol{v}$ and using relations $[\mathfrak{l}_k^{r_k}] = [\mathfrak{l}_1^{x_1} \cdots \mathfrak{l}_{k-1}^{x_{k-1}}]$ to reduce the result, cf. [30, §5].

This method may be used with any function f satisfying Lemma 2, and in particular it applies to two infinite families of functions:

(8) A_N, for $N > 2$ prime, with $\left(\frac{D}{N}\right) \neq -1$ and $N \nmid u$.

(9) $\mathfrak{w}^s_{p_1,p_2}$, for p_1, p_2 primes not both 2, with $\left(\frac{D}{p_1}\right), \left(\frac{D}{p_2}\right) \neq -1$ and $p_1, p_2 \nmid u$.

As above, u denotes the conductor of $D < -4N^2$.

As noted earlier, for certain primes p we may have difficulty computing the edges of G when $\gcd\big(\Psi_f(X, j_i), \Psi_f(X, [\mathfrak{n}] j_i)\big)$ has more than one root in $\mathbb{F}_p$. While we need not use such primes, it is often easy to determine the correct root. Here we give two heuristic techniques for doing so.

The first applies when N is prime, as with the Atkin functions. In this case problems can arise when $H_D[f]$ has repeated roots modulo p. By Kummer's criterion, this can happen only when p divides the discriminant of $H_D[f]$, and even then, a repeated root x_1 is only actually a problem when it corresponds to two alternating edges in G, say (j_1, j_2) and (j_3, j_4), with the edge (j_2, j_3) between them.

In this scenario we will get two roots x_1 and x_2 of $\gcd\big(\Psi_f(X, j_2), \Psi_f(X, j_3)\big)$. But if we already know that x_1 corresponds to (j_1, j_2), we can unambiguously choose x_2. In each of the N-isogeny cycles of G, it is enough to find a single edge that yields a unique root. If no such edge exists, then every edge must yield the *same* two roots x_1 and x_2, and we count each with multiplicity $n/2$.

The second technique applies when the roots of $H_D[f]$ are units, as with the double η-quotients [16, Thm. 3.3]. The product of the roots is then ± 1. Assuming that the number of edges in G for which multiple roots arise is small (it is usually zero, and rarely more than one or two), we simply test all the possible choices of roots and see which yield ± 1. If only one combination works, then the correct choices are determined. This is not guaranteed to happen, but in practice it almost always does.

4.4 A General Algorithm

We now briefly consider the case of an arbitrary modular function f of level N, and sketch a general algorithm to compute $H_D[f]$ with the CRT method.

Let us assume that $f(\tau)$ is a class invariant, and let D be the discriminant and u the conductor of the order $\mathcal{O} = [1, \tau]$. The roots of $\Psi_f(X, j(\tau)) \in K_{\mathcal{O}}[X]$ lie in the ray class field of conductor uN over K, and some number n of these, including $f(\tau)$, actually lie in the ring class field $K_{\mathcal{O}}$. We may determine n using the method described in [6, §6.4], which computes the action of $(\mathcal{O}/N\mathcal{O})^*/\mathcal{O}^*$ on the roots of $\Psi_f(X, j(\tau))$. We note that the complexity of this task is essentially fixed as a function of $|D|$.

Having determined n, we use Algorithm 2 to enumerate the roots $j_1, \ldots, j_h$ of $H_D \bmod p$ as usual, but if for any j_i we find that $\Psi_f(X, j_i) \bmod p$ does not have exactly n roots $f_i^{(1)}, \ldots, f_i^{(n)}$, we exclude the prime p from our computations. The number of such p is finite and may be bounded in terms of the discriminants of the polynomials $\Psi_f(X, \alpha)$ as α ranges over the roots of $H_D[f]$. We then compute the polynomial $H(X) = \prod_{i=1}^{h} \prod_{r=1}^{n} \big(X - f_i^{(r)}\big)$ of degree nh in $\mathbb{F}_p[X]$. After doing this for sufficiently many primes p, we can lift the coefficients by Chinese remaindering to the integers. The resulting H is a product of n distinct class polynomials, all of which may be obtained by factoring H in $\mathbb{Z}[X]$. Under suitable heuristic assumptions (including the GRH), the total time to compute $H_D[f]$ is quasi-linear in $|D|$, including the time to factor H.

This approach is practically efficient only when n is small, but then it can be quite useful. A notable example is the modular function g for which

$$\Psi_g(X, J) = (X^{12} - 6X^6 - 27)^3 - JX^{18}.$$

This function was originally proposed by Atkin, and is closely related to certain class invariants of Ramanujan [3, Thm. 4.1]. The function g yields class invariants when $D \equiv 13 \bmod 24$. In terms of our generic algorithm, we have $n = 2$, and for $p \equiv 2 \bmod 3$ we get exactly two roots of $\Psi_g(X, j_i) \bmod p$, which differ only in sign. Thus $H(X) = H_D[g^2](X^2) = H_D[g](X)H_D[g](-X)$, and from this we easily obtain $H_D[g^2]$, and also $H_D[g]$ if desired.

5 Computational Results

This section provides performance data for the techniques developed above. We used AMD Phenom II 945 CPUs clocked at 3.0 GHz for our tests; the software was implemented using the `gmp` [22] and `zn_poly` [24] libraries, and compiled with `gcc` [19].

To compute the class polynomial $H_D[f]$, we require a bound on the size of its coefficients. Unfortunately, provably accurate bounds for functions f other than j are generally unavailable. As a heuristic, we take the bound B on the coefficients of H_D given by [30, Lem. 8], divide $\log_2 B$ by the asymptotic height factor $c(f)$, and add a "safety margin" of 256 bits. We note that with the CM method, the correctness of the final result can be efficiently and unconditionally confirmed [5], so we are generally happy to work with a heuristic bound.

5.1 Class Polynomial Computations Using the CRT Method

Our first set of tests measures the improvement relative to previous computations with the CRT method. We used discriminants related to the construction of a large set of pairing-friendly elliptic curves, see [30, §8] for details. We reconstructed many of these curves, first using the Hilbert class polynomial H_D, and then using an alternative class polynomial $H_D[f]$. In each case we used the explicit CRT to compute H_D or $H_D[f]$ modulo a large prime q (170 to 256 bits).

Table 1 gives results for four discriminants with $|D| \approx 10^{10}$, three of which appear in [30, Table 2]. Each column lists times for three class polynomial computations. First, we give the total time T_{tot} to compute $H_D \bmod q$, including the time T_{enum} spent enumerating $\text{Ell}_D(\mathbb{F}_p)$, for all the small primes p, using Algorithm 2 as it appears in §2.2. We then list the times T'_{enum} and T'_{tot} obtained when Algorithm 2 is modified to use gcd computations whenever it is advantageous to do so, as explained in §2.3. The gcd approach typically speeds up Algorithm 2 by a factor of 2 or more.

For the third computation we selected a function f that yields class invariants for D, and computed $H_D[f] \bmod q$. This polynomial can be used in place of H_D in the CM method (one extracts a root x_0 of $H_D[f] \bmod q$, and then extracts a root of $\Psi_f(x_0, J) \bmod q$). For each function f we give a "size factor", which approximates the ratio of the total size of H_D to $H_D[f]$ (over $\mathbb{Z}$). In the first three examples this is just the height factor $c(f)$, but in Example 4 it is $4c(f)$ because the prime 59 is ramified and we actually work with the square root of $H_D[A_{59}]$, as noted in §4.1, reducing both the height and degree by a factor of 2.

We then list the speedup $T'_{\text{tot}}/T'_{\text{tot}}[f]$ attributable to computing $H_D[f]$ rather than H_D. Remarkably, in each case this speedup is about twice what one would expect from the height factor. This is explained by a particular feature of the CRT method: The cost of computing $H_D \bmod p$ for small primes p varies significantly, and, as explained in [30, §3], one can accelerate the CRT method with a careful choice of primes. When fewer small primes are needed, we choose those for which Step 1 of Algorithm 1 can be performed most quickly.

The last line in Table 1 lists the total speedup $T_{\text{tot}}/T'_{\text{tot}}[f]$ achieved.

Table 1. Example class polynomial computations (times in CPU seconds)

	Example 1	Example 2	Example 3	Example 4
$\lvert D\rvert$	13569850003	11039933587	12901800539	12042704347
$h(D)$	20203	11280	54706	9788
$\lceil \log_2 B \rceil$	2272564	1359134	5469776	1207412
$(\ell_1^{r_1}, \ldots, \ell_k^{r_k})$	(7^{20203})	$(17^{1128}, 19^{10})$	$(3^{27038}, 5^2)$	$(29^{2447}, 31^2, 43^2)$
T_{enum} (roots)	6440	10200	10800	21700
T_{tot}	19900	23700	52200	42400
T'_{enum} (gcds)	2510	2140	3440	4780
T'_{tot}	15900	15500	44700	25300
Function f	A_{71}	A_{47}	A_{71}	A_{59}
Size factor	36	24	36	120*
$T'_{\text{tot}}[f]$	213	305	629	191
Speedup $(T'_{\text{tot}}/T'_{\text{tot}}[f])$	75	51	71	132
Speedup $(T_{\text{tot}}/T'_{\text{tot}}[f])$	**93**	**78**	**83**	**222**

5.2 Comparison to the Complex Analytic Method

Our second set of tests compares the CRT approach to the complex analytic method. For each of the five discriminants listed in Table 2 we computed class polynomials $H_D[f]$ for the double η-quotient $\mathfrak{w}_{3,13}$ and the Weber $\mathfrak{f}$ function, using both the CRT approach described here, and the implementation [14] of the complex analytic method as described in [12]. With the CRT we computed $H_D[f]$ both over $\mathbb{Z}$ and modulo a 256-bit prime q; for the complex analytic method these times are essentially the same.

Table 2. CRT vs. complex analytic (times in CPU seconds)

		complex analytic		CRT		CRT mod q	
$\lvert D\rvert$	$h(D)$	$\mathfrak{w}_{3,13}$	$\mathfrak{f}$	$\mathfrak{w}_{3,13}$	$\mathfrak{f}$	$\mathfrak{w}_{3,13}$	$\mathfrak{f}$
6961631	5000	15	5.4	2.2	1.0	2.1	1.0
23512271	10000	106	33	10	4.1	9.8	4.0
98016239	20000	819	262	52	22	47	22
357116231	40000	6210	1900	248	101	213	94
2093236031	100000	91000	27900	2200	870	1800	770

We also tested a "worst case" scenario for the CRT approach: the discriminant $D = -85702502803$, for which the smallest non-inert prime is $\ell_1 = 109$. Choosing the function most suitable to each method, the complex analytic method computes $H_D[\mathfrak{w}_{109,127}]$ in 8310 seconds, while the CRT method computes $H_D[A_{131}]$

in 7150 seconds. The CRT approach benefits from the attractive height factor of the Atkin functions, $c(A_{131}) = 33$ versus $c(\mathfrak{w}_{109,127}) \approx 12.4$, and the use of gcds in Algorithm 2. Without these improvements, the time to compute H_D with the CRT method is 1460000 seconds. The techniques presented here yield more than a 200-fold speedup in this example.

5.3 A Record-Breaking CM Construction

To test the scalability of the CRT approach, we constructed an elliptic curve using $|D| = 1000000013079299 > 10^{15}$, with $h(D) = 10034174 > 10^7$. This yielded a curve $y^2 = x^3 - 3x + c$ of prime order n over the prime field $\mathbb{F}_q$, where

$$c = 1222944565023569747153953185348208174607248719445203935546780433368429857904 7;$$
$$q = 2894802230932904885589274625217198164611328854890480596109405842425674316903 3;$$
$$n = 2894802230932904885589274625217198164645357091582574442455743303168851140801 3.$$

This curve was obtained by computing the square root of $H_D[A_{71}]$ modulo q, a polynomial of degree $h(D)/2 = 5017087$. The height bound of 21533832 bits was achieved with 438709 small primes p, the largest of which was 53 bits in size. The class polynomial computation took slightly less than a week using 32 cores, approximately 200 days of CPU time. Extracting a root over $\mathbb{F}_q$ took 25 hours of CPU time using NTL [29].

We estimate that the size of $\sqrt{H_D[A_{71}]}$ is over 13 terabytes, and that the size of the Hilbert class polynomial H_D is nearly 2 petabytes. The size of $\sqrt{H_D[A_{71}]} \bmod q$, however, is under 200 megabytes, and less than 800 megabytes of memory (per core) were needed to compute it.

References

[1] Bach, E.: Explicit bounds for primality testing and related problems. Mathematics of Computation 55(191), 355–380 (1990)

[2] Belding, J., Bröker, R., Enge, A., Lauter, K.: Computing Hilbert class polynomials. In: van der Poorten, A.J., Stein, A. (eds.) ANTS-VIII 2008. LNCS, vol. 5011, pp. 282–295. Springer, Heidelberg (2008)

[3] Berndt, B.C., Chan, H.H.: Ramanujan and the modular j-invariant. Canadian Mathematical Bulletin 42(4), 427–440 (1999)

[4] Bernstein, D.J.: Modular exponentiation via the explicit Chinese Remainder Theorem. Mathematics of Computation 76, 443–454 (2007)

[5] Bisson, G., Sutherland, A.V.: Computing the endomorphism ring of an ordinary elliptic curve over a finite field. Journal of Number Theory (2009) (to appear), http://arxiv.org/abs/0902.4670

[6] Bröker, R.: Constructing elliptic curves of prescribed order. Universiteit Leiden, Proefschrift (2006)

[7] Bröker, R.: A p-adic algorithm to compute the Hilbert class polynomial. Mathematics of Computation 77, 2417–2435 (2008)

[8] Bröker, R., Lauter, K., Sutherland, A.V.: Modular polynomials via isogeny volcanoes (2009) (preprint), http://arxiv.org/abs/1001.0402

[9] Couveignes, J.-M., Henocq, T.: Action of modular correspondences around CM points. In: Fieker, C., Kohel, D.R. (eds.) ANTS 2002. LNCS, vol. 2369, pp. 234–243. Springer, Heidelberg (2002)
[10] Elkies, N.D.: Elliptic and modular curves over finite fields and related computational issues. In: Buell, D.A., Teitelbaum, J.T. (eds.) Computational Perspectives on Number Theory, pp. 21–76. AMS, Providence (1998)
[11] Enge, A.: Courbes algébriques et cryptologie. In: Habilitation à diriger des recherches, vol. 7. Université Denis Diderot, Paris (2007)
[12] Enge, A.: The complexity of class polynomial computation via floating point approximations. Mathematics of Computation 78(266), 1089–1107 (2009)
[13] Enge, A.: Computing modular polynomials in quasi-linear time. Mathematics of Computation 78(267), 1809–1824 (2009)
[14] Enge, A.: cm, 0.2 edition (2010), http://cm.multiprecision.org/
[15] Enge, A., Morain, F.: Generalised Weber functions. I. Technical Report 385608, HAL-INRIA (2009), http://hal.inria.fr/inria-00385608
[16] Enge, A., Schertz, R.: Constructing elliptic curves over finite fields using double eta-quotients. Journal de Théorie des Nombres de Bordeaux 16, 555–568 (2004)
[17] Enge, A., Schertz, R.: Modular curves of composite level. Acta Arithmetica 118(2), 129–141 (2005)
[18] Enge, A., Schertz, R.: Singular values of multiple eta-quotients for ramified primes (in preparation 2010)
[19] Free Software Foundation. GNU Compiler Collection, 4.2.4 edition (2008), http://gcc.gnu.org/
[20] Gee, A.: Class fields by Shimura reciprocity. Universiteit Leiden, Proefschrift (2001)
[21] Gee, A., Stevenhagen, P.: Generating class fields using Shimura reciprocity. In: Buhler, J.P. (ed.) ANTS 1998. LNCS, vol. 1423, pp. 441–453. Springer, Heidelberg (1998)
[22] Granlund, T., et al.: gmp, 4.3.1 edition (2009). http://gmplib.org/.
[23] Hajir, F., Villegas, F.R.: Explicit elliptic units, I. Duke Mathematical Journal 90(3), 495–521 (1997)
[24] Harvey, D.: zn_poly: a library for polynomial arithmetic, 0.9 edn. (2008), http://cims.nyu.edu/~harvey/zn_poly
[25] Kohel, D.: Endomorphism rings of elliptic curves over finite fields. PhD thesis, University of California at Berkeley (1996)
[26] Morain, F.: Calcul du nombre de points sur une courbe elliptique dans un corps fini: aspects algorithmiques. Journal de Théorie des Nombres de Bordeaux 7(1), 111–138 (1995)
[27] Morain, F.: Advances in the CM method for elliptic curves. In: Slides of Fields Cryptography Retrospective Meeting, May 11-15 (2009), http://www.lix.polytechnique.fr/~morain/Exposes/fields09.pdf
[28] Schertz, R.: Weber's class invariants revisited. Journal de Théorie des Nombres de Bordeaux 14(1), 325–343 (2002)
[29] Shoup, V.: NTL: A library for doing number theory, 5.5 edn. (2008), http://www.shoup.net/ntl/
[30] Sutherland, A.V.: Computing Hilbert class polynomials with the Chinese Remainder Theorem. Mathematics of Computation (to appear 2010), http://arxiv.org/abs/0903.2785
[31] Weber, H.: Lehrbuch der Algebra, 3rd edn., vol. III. Chelsea, New York (1961)

Short Bases of Lattices over Number Fields

Claus Fieker[1] and Damien Stehlé[1,2]

[1] Magma Computer Algebra Group, School of Mathematics and Statistics, University of Sydney, NSW 2006, Australia
[2] CNRS and Macquarie University
claus.fieker@sydney.edu.au, damien.stehle@gmail.com

Abstract. Lattices over number fields arise from a variety of sources in algorithmic algebra and more recently cryptography. Similar to the classical case of $\mathbb{Z}$-lattices, the choice of a nice, "short" (pseudo)-basis is important in many applications. In this article, we provide the first algorithm that computes such a "short" (pseudo)-basis. We utilize the LLL algorithm for $\mathbb{Z}$-lattices together with the Bosma-Pohst-Cohen Hermite Normal Form and some size reduction technique to find a pseudo-basis where each basis vector belongs to the lattice and the product of the norms of the basis vectors is bounded by the lattice determinant, up to a multiplicative factor that is a field invariant. As it runs in polynomial time, this provides an effective variant of Minkowski's second theorem for lattices over number fields.

1 Introduction

Let K be a number field and $\mathcal{O}_K$ be its maximal order. An $\mathcal{O}_K$-module M is a finitely generated set of elements which is closed under addition and multiplication by elements in $\mathcal{O}_K$. Frequently, we have $M \subseteq K^m$ for some m. In the case of K being $\mathbb{Q}$, we have $\mathcal{O}_K = \mathbb{Z}$, thus $\mathcal{O}_K$-modules are just the classical $\mathbb{Z}$-lattices. Since $\mathbb{Z}$ is a principal ideal domain, every (torsion free) module is free, thus there exists a basis $b_1, \ldots, b_n \in M$ for some $n \leq m$ such that $M = \oplus_{i \leq n} \mathbb{Z} b_i$. Any two bases $(b_i)_i$ and $(c_i)_i$ have the same cardinality and are linked by some unimodular matrix $T \in \mathrm{GL}(n, \mathbb{Z})$. The choice of a *good* basis is crucial for almost all computational problems attached to M. Generally one tries to find a basis whose vectors have short Euclidean norms, using, for example, the LLL algorithm [15].

Replacing $\mathbb{Z}$ by the maximal order $\mathcal{O}_K$ makes the classification more complicated since $\mathcal{O}_K$ may no longer be a principal ideal domain. However, since $\mathcal{O}_K$ is still a Dedekind domain, the modules $M \subseteq K^m$ have a well known structure ([7, Cor. 1.2.25], [23, Th. 81:3]): there exist linearly independent elements $\mathbf{b}_1, \ldots, \mathbf{b}_n \in K^m$ and (non-zero fractional) ideals $\mathfrak{b}_1, \ldots, \mathfrak{b}_n$ such that $M = \oplus_{i \leq n} \mathfrak{b}_i \mathbf{b}_i$, i.e., every $\mathbf{b} \in M$ has a unique representation as $\mathbf{b} = \sum_{i \leq n} x_i \mathbf{b}_i$ with $x_i \in \mathfrak{b}_i$ for all $i \leq n$. Such a representation is commonly called a *pseudo-basis*. It should be noted that $\mathbf{b}_i$ may not belong to M, and in fact $\mathbf{b}_i \in M$ if and only if $1 \in \mathfrak{b}_i$. Similarly to the case of $\mathbb{Z}$-lattices, different pseudo-bases share the same cardinality, and it is known how to move from a pseudo-basis to another.

G. Hanrot, F. Morain, and E. Thomé (Eds.): ANTS-IX 2010, LNCS 6197, pp. 157–173, 2010.

As for $\mathbb{Z}$-lattices, the choice of the pseudo-basis is of utmost importance. However, a key difference is that no analogue of LLL is known, as repeatedly noted in [7]. There have been attempts [10,22,11] but the algorithms are either limited to certain fields or give no guaranteed bounds on the output size. While every $\mathcal{O}_K$-module is also a $\mathbb{Z}$-lattice and can thus be analyzed with all the tools available over $\mathbb{Z}$, for many applications the additional structure as an $\mathcal{O}_K$-module is important. This structure is typically lost when applying techniques over $\mathbb{Z}$.

Originally, $\mathcal{O}_K$-modules mainly came from the study of finite extensions of K but now they occur in a wider range of problems from group theory (matrix groups and representations [9]) to applications in geometry (automorphism algebras of Abelian varieties). $\mathcal{O}_K$-modules also occur in lattice-based cryptography [17,19,24,25,26], and in that context the module rank n is usually polylogarithmic in the degree of the number field. Cryptography based on $\mathcal{O}_K$-modules is increasingly popular, as on one side they lead to compact representations and to fast operations, and on the other side they enjoy a worst-case to average-case reduction for variants of the shortest vector problem, which allows the cryptographic security to be based on worst-case hardness assumptions.

As diverse as the applications are the requirements: only one (or more) short module element(s) may be needed, or a short (pseudo)-basis may be required, some applications rely on canonical representations, while any representation may suffice for others. We note that canonical representations tend to have components that are much larger than short representations as obtained by lattice reduction or our techniques. To find one short element it suffices to consider the underlying $\mathbb{Z}$-module (of dimension nd with $d = [K : \mathbb{Q}]$). For $\mathbb{Z}$-lattices contained in $\mathbb{Q}^m$, a canonical representation is the Hermite Normal Form (HNF). It has been generalized (BPC-HNF) to $\mathcal{O}_K$-modules contained in K^m by Bosma and Pohst [4] and Cohen [7, Chap. 1.4] (see also [12]).

Our results. In the present work, we describe an algorithm that computes a pseudo-basis made of short vectors. Given an arbitrary pseudo-basis $[(\mathbf{a}_i)_i, (\mathfrak{a}_i)_i]$ of a module $M \subseteq K^m$, it returns a pseudo-basis $[(\mathbf{b}_i)_i, (\mathfrak{b}_i)_i]$ such that:

$$\forall i \leq n: \quad \mathbf{b}_i \in M,\ \mathcal{N}(\mathfrak{b}_i) \in [2^{-O(d^2)}, 1] \ \text{ and } \|\mathbf{b}_i\| \leq 2^{O(dn)} \lambda_i(M),$$

where the $O(\cdot)$'s depend only on the field K and the choice of a given LLL-reduced integral basis, the euclidean norm $\|\cdot\|$ is a module extension of the T_2-norm over K, and the $\lambda_i(M)$'s correspond to the module minima. We refer to Corollary 1 for a precise statement. Overall, this provides a module equivalent to LLL-reduced bases of $\mathbb{Z}$-lattices in the sense that the vectors cannot be arbitrarily longer than the minima. Since it runs in polynomial time, it can also be interpreted as an effective approximate variant of the adaptation to $\mathcal{O}_K$-modules of Minkowski's second theorem (given in Theorem 2). We also study the representation of one-dimensional $\mathcal{O}_K$-modules, i.e., modules that are isomorphic to ideals of $\mathcal{O}_K$. We show how to modify Belabas' 2-element representation algorithm [2, Alg. 6.15] so that the output is provably small. Combining the latter and our module pseudo-reduction algorithm leads to compact representations of $\mathcal{O}_K$-modules.

The most natural approach to obtain reduced pseudo-bases consists in trying to generalize LLL, but as mentioned earlier all previous attempts have only partially succeeded. In contrast, we start by viewing the $\mathcal{O}_K$-module as a high-dimensional $\mathbb{Z}$-lattice. We find short module elements by applying LLL to a basis of the latter lattice and interpreting the output as module elements. At this point, we have a pseudo-basis (the input) and a full-rank set of short module vectors (produced by LLL). If we had a $\mathbb{Z}$-lattice instead of an $\mathcal{O}_K$-module, we would then use a technique common in the lattice-based cryptography community (see, e.g., [20, Le. 7.1]), consisting in using the HNF to convert a full rank set of short lattice vectors to a short basis. We adapt this technique to number fields, using the BPC-HNF and introducing a size-reduction algorithm for pseudo-bases.

Let us compare (pseudo-)LLL-reduced and BPC-HNF pseudo-bases. A theoretical advantage of the LLL approach is that it is not restricted to K^m but also works in a continuous extension (similarly to LLL-reduction being well-defined for real lattices). It should also be significantly more efficient to work with pseudo-bases made of short vectors because smaller integers and polynomials of smaller degrees are involved. On the other side, (pseudo-)LLL-reduced pseudo-bases are far from being unique, and seem more expensive to obtain.

Road-map. In Section 2, we give some reminders and elementary results on lattices, number fields and modules. In Section 3, we modify Belabas' 2-element representation algorithm for ideals of $\mathcal{O}_K$, as described above. We then give our module reduction algorithm in Section 4. Finally, in Section 5 we describe our implementation and give some examples.

Implementation. The algorithms have been implemented in the Magma computer algebra system [3,18] and are available on request. They will be part of upcoming releases.

2 Preliminaries

We assume the reader is familiar with the geometry of numbers and algebraic number theory. We refer to [16,20], [5,21] and [7, Chap. 1] for introductions to the computational aspects of lattices, elementary algebraic number theory and to modules over Dedekind domains, respectively.

2.1 Lattices

In this work, we will call any finitely generated free $\mathbb{Z}$-module L a lattice. A usual lattice corresponds to the case where L is a discrete additive subgroup of $\mathbb{R}^n$ for some n. Any lattice can be written $L = \oplus_{i \leq d} \mathbb{Z} b_i$. If the b_i's are $\mathbb{Z}$-free, they are called a basis of L. A given lattice may have infinitely many bases but their cardinality d is constant and called rank. Any two bases are related by a unimodular transformation, i.e., one is obtained from the other by multiplying by a matrix in $\mathbb{Z}^{d \times d}$ of determinant ± 1.

If $L \subseteq \mathbb{Q}^n$ is of rank d, then there exists a basis $B = (\mathbf{b}_i)_i \in \mathbb{Q}^{n \times d}$ of L such that $\mu_j = \min\{i : B_{i,j} \neq 0\}$ (strictly) increases with j, and for all $j > k$ we

have $B_{\mu_j,j} > B_{\mu_j,k} \geq 0$. If $d = n$, this means that B is a row-wise diagonally strictly dominant lower triangular matrix and that its entries are non-negative. This basis is unique and called the Hermite Normal Form (HNF) of L. It can be computed in polynomial time from any basis [13].

In order to quantify the smallness of an element of a lattice L, we associate to L a positive definite bilinear form $q : L_{\mathbb{R}} \times L_{\mathbb{R}} \mapsto \mathbb{R}$. We use it to map a basis $(b_i)_i$ to its Gram matrix $G_q(b_1, \ldots, b_d) := (q(b_i, b_j))_{i,j}$. We denote $\sqrt{q(b,b)}$ by $\|b\|_q$, and may omit the subscript if it is clear from the context. The determinant of L, defined as $\det_q(L) = \det(G_q(b_1, \ldots, b_d))^{1/2}$, does not depend on the particular choice of the basis of L. Note that if $L \subseteq \mathbb{R}^n$ and q is the euclidean inner product, then $\det(L)$ is the d-dimensional volume of the parallelepiped $\{\sum_i y_i b_i : y_i \in [0,1]\}$. We define the lattice minima as follows:

$$\forall i \leq d,\ \lambda_{i,q}(L) = \min\{r : \exists c_1, \ldots, c_i \in L \text{ free},\ \max_{k \leq i} \|c_k\|_q \leq r\}.$$

Minkowski's second theorem states that $\prod_{i \leq d} \lambda_{i,q}(L) \leq \sqrt{d}^d \det_q(L)$. Frequently one tries to represent a lattice L by a basis that approximates the minima. In this article, we assume that we have an algorithm `LatRed` that takes as input an arbitrary basis of L and returns a reduced basis satisfying $\|b_i\| \leq \gamma \lambda_i(L)$, for all $i \leq d$. For example, if we use the LLL algorithm [15], then we can take $\gamma = 2^{d/2}$. We proceed as follows: compute the Gram matrix G of the input basis; use the Gram matrix LLL algorithm (see, e.g., [5, p. 88]), to find U unimodular such that $U^t G U$ is reduced; apply U to the input lattice basis. If the arithmetic over L is efficient, and if q can be efficiently computed or approximated with high accuracy, then this provides an efficient algorithm. Apart from being well-defined for more general lattices (not only for lattices on a rational vector space), a significant advantage of the LLL-reduction over the HNF is that it provides small lattice elements. However, it seems more expensive to obtain and the uniqueness of the representation is lost. Taking the HKZ-reduction instead of the LLL-reduction allows one to take $\gamma = 1/2\sqrt{d+3}$ (see [14]), but the complexity of the best algorithm for computing it [1] is exponential in d.

Let $(b_i)_{i \leq d}$ be a lattice basis. For any $i > j$, we define $\mu_{i,j} = q(b_i, b_j^*)/q(b_j^*, b_j^*)$, where $b_i^* = \text{argmin} \|b_i + \sum_{j<i} \mathbb{R} b_j\|$ thus $\|b_i^*\| = \min\{\|b_i + x\| : x \in \sum_{j<i} \mathbb{R} b_j\}$. We call the $\mu_{i,j}$'s and the b_i^*'s the Gram-Schmidt orthogonalisation (GSO) of the b_i's. If the b_i's are LLL-reduced, then $\|b_i^*\| \geq 2^{-d/2} \|b_i\|$ for all i. In the following, we will assume that `LatRed`-reduced bases also satisfy this property. Size-reduction of a vector $b \in \sum_{i \leq d} \mathbb{R} b_i$ with respect to $(b_i)_{i \leq j}$ consists in subtracting from b integer multiples of these b_i's so that the magnitudes of the first j coordinates of the output vector c when written as a linear combination of all the b_i^*'s belong to $[-1/2, 1/2)$. The latter uniquely defines c, and if $j = d$ we have $\|c\|^2 \leq \sum_{i \leq d} \|b_i^*\|^2 \leq d \max_{i \leq d} \|b_i\|^2$. We call size-reduction of the basis $(b_i)_i$ the process of size-reducing each b_i with respect to the previous b_j's for increasing i. The output remains a basis of the lattice spanned by the b_i's.

A standard technique in the lattice-based cryptography community (see, e.g., [20, Le. 7.1]) allows one to derive a short lattice basis from an arbitrary basis $(a_i)_i$ and a full-rank free set of short lattice vectors $(s_i)_i$. As we will adapt

this technique to modules, we describe it briefly. Since the s_i's belong to the lattice, there exists $T \in \mathbb{Z}^{d\times d}$ such that $(s_i)_i = (a_i)_i \cdot T$. We compute the HNF T'^t of T^t: $T'^t = T^t(U^{-1})^t$ with U unimodular. We thus have $(s_i)_i = (b_i)_i \cdot T'$ where $(b_i)_i := (a_i)_i \cdot U$ is a lattice basis and T' is upper triangular with diagonal entries ≥ 1. The shape of T' implies that for any i we have $\|b_i^*\| \leq \|s_i^*\|$. Size-reducing the basis $(b_i)_i$ leads to a basis $(b'_i)_i$ such that $\max\|b'_i\| \leq \sqrt{d}\max\|s_i^*\| \leq \sqrt{d}\max\|s_i\|$. It can be checked that if $L \subseteq \mathbb{Q}^n$, then all the computations may be performed in polynomial time.

2.2 Number Fields

Let K be a number field of degree d, with real and complex embeddings $(\theta_i)_{i\leq s_1}$, $(\theta_i)_{s_1<i\leq s_1+2s_2}$. Its maximal order $\mathcal{O}_K$ is a lattice: there exists a free set $(r_i)_i \in \mathcal{O}_K^d$ such that $\mathcal{O}_K = \oplus_i \mathbb{Z}r_i$. The r_i's form an integral basis of K, and we have $K = \mathcal{O}_K \otimes \mathbb{Q}$. We define $K_\mathbb{R} = K \otimes \mathbb{R}$, which is isomorphic (as rings) to $\mathbb{R}^{s_1} \times \mathbb{C}^{s_2}$, and extend the θ_i's to $K_\mathbb{R}$. Many quadratic forms may be associated to $K_\mathbb{R}$, but the most natural one derives from $q(x, x') = T_2(x, x') := \sum \theta_i(x)\bar{\theta}_i(x')$. The discriminant of K is defined as $\Delta_K = \det^2_{T_2}(\mathcal{O}_K)$. Note that for any $x, x' \in K_\mathbb{R}$, we have $\|xx'\| \leq \|x\| \cdot \|x'\|$ where $\|x\| := T_2(x)^{1/2}$ is the induced norm. The (field) norm of an element $x \in K_\mathbb{R}$ is defined as $\mathcal{N}(x) = \prod_i |\theta_i(x)|$. Note that with our definition, the norm cannot be negative.

A (fractional) ideal I is any finitely generated $\mathcal{O}_K$-module contained in K. An integral ideal I is a fractional ideal contained in $\mathcal{O}_K$. For any fractional ideal I there exists $r \in \mathbb{Z}$ such that rI is an integral ideal. If $r \in K$, we let (r) denote the (principal) ideal $r\mathcal{O}_K$. The product $IJ = \langle ij : i \in I, j \in J\rangle$ and the sum $I + J = \{i + j : i \in I, j \in J\}$ of two ideals are also ideals. A non-zero integral ideal is said to be prime if it is divisible only by $\mathcal{O}_K$ and itself. As $\mathcal{O}_K$ is a Dedekind domain, any non-zero fractional ideal can be uniquely decomposed as a product of (possibly negative) powers of prime ideals. If $\mathfrak{p}$ is a prime ideal, we define $\nu_\mathfrak{p}(I) = \max(k \in \mathbb{Z} : \mathfrak{p}^k | I)$. The norm of I is defined as $\mathcal{N}(I) = \det(I)/\det(\mathcal{O}_K)$. If $I \neq 0$ is integral, then this is exactly the index of I in $\mathcal{O}_K$, defined as $[\mathcal{O}_K : I] = |\mathcal{O}_K/I|$. We define $\mathcal{N}(0) = 0$, which allows us to assert that $\mathcal{N}(IJ) = \mathcal{N}(I)\mathcal{N}(J)$ for any ideals I and J. Note that if $I = (r)$ is principal, then $\mathcal{N}(I) = \mathcal{N}(r)$. The inverse $I^{-1} = \{r \in K : rI \subseteq \mathcal{O}_K\}$ of a non-zero fractional ideal I is also a fractional ideal, and we have $II^{-1} = \mathcal{O}_K$. Note that the arithmetic over the ideals can be performed in polynomial time (e.g., see [2]).

Any non-zero ideal, including the maximal order, is naturally a free $\mathbb{Z}$-module of rank d thus a lattice under the T_2-norm. By fixing an integral basis for K, we also fix a $\mathbb{Z}$-lattice structure for $\mathcal{O}_K$ that we can then reduce. We say that a basis of a non-zero fractional ideal I is in HNF if the (rational) matrix of the coefficients with respect to a fixed integral basis of K is in HNF. This provides a unique representation for any ideal. In the following, we assume that we know an integral basis $(r_i)_i$ of K that is `LatRed`-reduced with respect to T_2. It can be known for particular K's (e.g., cyclotomic number fields, with $\max\|r_i\|^2 = d$), or can be computed by reducing an arbitrary integral basis. As it is computed

once and for all, it may prove interesting to strongly reduce it. We have the following result.

Lemma 1. *If $(r_i)_i$ is a* `LatRed`*-reduced integral basis of K, then* $\max \|r_i\| \leq \sqrt{d}\gamma^d \sqrt{\Delta_K}$.

Proof. Using the reducedness and Minkowski's second theorem, we get $\prod \|r_i\|^2 \leq \gamma^{2d} d^d \Delta_K$. The arithmetic-geometric inequality gives $1 \leq \mathcal{N}(r_i)^{2/d} \leq \|r_i\|^2/d$ for all i, which provides the result. □

The bounds of our main results involve the quantity $\max \|r_i\|$. Lemma 1 allows one to express them with field invariants only. We choose to keep $\max \|r_i\|$ in our bounds since it can be much smaller, as in the case of cyclotomic number fields.

With our a choice of integral basis, any element of $\mathcal{O}_K$ with small T_2-norm can be represented with a small number of bits.

Lemma 2. *Assume that $(r_i)_i$ is a* `LatRed`*-reduced integral basis of K. If $x = \sum x_i r_i \in K$, then* $\max |x_i| \leq 2^{3d/2}\|x\|$.

Proof. We show by induction of i that

$$\forall i : |x_i| \leq 2^{d-i} \frac{\|x\|}{\min_j \|r_j^*\|}.$$

First, we have $\|x\| \geq |x_d|\|r_d^*\|$. Suppose now that $i < d$ and that the result holds for any $j > i$. The GSO of the r_i's shows that $\|x\| \geq |x_i + \sum_{j>i} \mu_{j,i} x_j| \|r_i^*\|$. Therefore, we have $|x_i| \leq \|x\|/\|r_i^*\| + \sum_{j>i} |x_j|$, which gives the bound. To complete the proof, note that the reducedness of the r_i's gives $\min_j \|r_j^*\| \geq 2^{-d/2} \min_j \|r_j\|$, and that $\|r_j\| \geq \sqrt{d}$ for all j. □

2.3 $\mathcal{O}_K$-Modules

Let $\mathbf{b}_1, \ldots, \mathbf{b}_n \in K_{\mathbb{R}}^m$ with $n = \mathrm{rank}_K(\mathbf{b}_i)_i$, and $\mathfrak{b}_1, \ldots, \mathfrak{b}_n$ be fractional ideals of $\mathcal{O}_K$. The $\mathcal{O}_K$-module $M[(\mathbf{b}_i)_i, (\mathfrak{b}_i)_i]$ spanned by the pseudo-basis $[(\mathbf{b}_i)_i, (\mathfrak{b}_i)_i]$ is $\sum \mathfrak{b}_i \mathbf{b}_i$. The $\mathfrak{b}_i$'s are called the coefficient ideals. As each $\mathfrak{b}_i$ is a $\mathbb{Z}$-lattice, so is M. More precisely, if $\mathfrak{b}_i = \sum_{j \leq d} \mathbb{Z}\beta_i^{(j)}$, then $M = \sum_{i,j} \mathbb{Z}\beta_i^{(j)} \mathbf{b}_i$. Two pseudo-bases $[(\mathbf{b}_i)_i, (\mathfrak{b}_i)_i]$ and $[(\mathbf{c}_i)_i, (\mathfrak{c}_i)_i]$ represent the same $\mathcal{O}_K$-module M if and only if there exists a non-singular $U \in K^{n \times n}$ with ([23, §81 C]):

1. $(\mathbf{c}_1, \ldots, \mathbf{c}_n) = (\mathbf{b}_1, \ldots, \mathbf{b}_n)U$;
2. For all i, j, we have $U_{i,j} \in \mathfrak{b}_i \mathfrak{c}_j^{-1}$;
3. For all i, j, we have $U'_{i,j} \in \mathfrak{c}_i \mathfrak{b}_j^{-1}$, where $U' = U^{-1}$.

Cohen [6] generalized the HNF to modules in K^m. The algorithm of [4] may also be interpreted as such a generalization. We refer to [12, Chap. 4] for a detailed exposure and comparison.

Theorem 1. *Let $M \subseteq K^m$ be an $\mathcal{O}_K$-module of rank n. There exists a pseudo-basis $[(\mathbf{b}_i)_i, (\mathfrak{b}_i)_i]$ of M such that $\mu_j = \min\{i : B_{i,j} \neq 0\}$ (strictly) increases with j, for all j we have $B_{\mu_j,j} = 1$ and for all $j > k$ the entry $B_{\mu_j,k} \in K$ is size-reduced modulo the HNF of $\mathfrak{b}_j\mathfrak{b}_k^{-1}$. This unique pseudo-basis is called the HNF of M. It can be computed in polynomial time from any pseudo-basis of M.*

Similarly to the HNF for lattices, the above HNF can only handle $\mathcal{O}_K$-modules $M \subseteq K^m$ (as opposed to $K_{\mathbb{R}}^m$) and does not necessarily contain small elements of M. We now define the concept of small-ness for elements of $K_{\mathbb{R}}^m$. For any two vectors $\mathbf{b} = (b_1, \ldots, b_m)^t, \mathbf{b}' = (b'_1, \ldots, b'_m)^t \in K_{\mathbb{R}}^m$, we define $T_2^{\otimes m}(\mathbf{b}, \mathbf{b}') = \sum_{i \leq m} T_2(b_i, b'_i)$, and we denote $\sqrt{T_2^{\otimes m}(\mathbf{b}, \mathbf{b})}$ by $\|\mathbf{b}\|$. Notice that for any $(r, \mathbf{b}) \in K_{\mathbb{R}} \times K_{\mathbb{R}}^m$, we have $\|r\mathbf{b}\| \leq \|r\| \cdot \|\mathbf{b}\|$. With this definition at hand, we can define the minima of M:

$$\forall i \leq n,\ \lambda_i(M) = \min\{r : \exists \mathbf{c}_1, \ldots, \mathbf{c}_i \in M, \mathrm{rank}_K(\mathbf{c}_k)_k = i \ \text{ and } \ \max \|\mathbf{c}_k\| \leq r\}.$$

Let $[(\mathbf{b}_i)_i, (\mathfrak{b}_i)_i]$ be a pseudo-basis of an $\mathcal{O}_K$-module $M \subseteq K_{\mathbb{R}}^m$. Assume that $\mathfrak{b}_i = \sum_{j \leq d} \mathbb{Z}\beta_i^{(j)}$. We define $\det(M)$ as the square root of the determinant of the $nd \times nd$ symmetric positive definite matrix $T_2^{\otimes m}(\beta_i^{(j)}\mathbf{b}_i, \beta_{i'}^{(j')}\mathbf{b}_{i'})_{i,j;i',j'}$. This is a module invariant. When M is a non-zero fractional ideal of $\mathcal{O}_K$, this matches $\det_{T_2}(M)$. It should be noted that $\det(M)$ is not immediately related to the (Steinitz) class of M nor to the maximal exterior power of M. The following is a direct consequence of Minkowski's second theorem over $\mathbb{Z}$-lattices.

Theorem 2. *Let $M \subseteq K_{\mathbb{R}}^m$ be an $\mathcal{O}_K$-module of rank n. Then $\prod_{i \leq n} \lambda_i(M) \leq \sqrt{dn}^n \det(M)^{1/d}$.*

Proof. The module M can be seen as a lattice L of dimension nd, with $\det(M) = \det(L)$. Minkowski's second theorem asserts that $\prod_{i \leq nd} \lambda_i(L) \leq \sqrt{dn}^{dn} \det(L)$. Let $c_1, \ldots, c_{nd} \in M$ be free over the integers such that $\|c_i\| = \lambda_i(L)$ holds for all i. For all $i \leq n$, let $\phi(i) = \min(j : \mathrm{rank}_K(c_1, \ldots, c_j) = i)$. As $\mathcal{O}_K$ has rank d as a $\mathbb{Z}$-module, we have $\phi(i) \leq (i-1)d + 1$. We conclude with the following sequence of inequalities:

$$\prod_{i \leq n} \lambda_i(M) \leq \prod_{i \leq n} \|c_{\phi(i)}\| \leq \prod_{i \leq n} \lambda_{(i-1)d+1}(L) \leq \prod_{i \leq dn} \lambda_i(L)^{\frac{1}{d}} \leq \sqrt{dn}^n \det(M)^{\frac{1}{d}}. \quad \square$$

We now extend the concept of GSO. Let $[(\mathbf{b}_i)_i, (\mathfrak{b}_i)_i]$ be a pseudo-basis of an $\mathcal{O}_K$-module M. We define $\mathbf{b}_i^* = \mathrm{argmin}\|\mathbf{b}_i + \sum_{j<i} K_{\mathbb{R}}\mathbf{b}_j\|$ for all $i \leq n$, and let $\mu_{i,1}, \ldots, \mu_{i,i-1} \in K_{\mathbb{R}}$ be such that $\mathbf{b}_i = \mathbf{b}_i^* + \sum_{j<i} \mu_{i,j}\mathbf{b}_j^*$.

3 Small 2-Element Representation of an Ideal

We start our study of $\mathcal{O}_K$-modules by the one-dimensional case, i.e., fractional ideals of K. There are several ways of representing an ideal $I \neq 0$. A natural approach is to provide a basis $(b_i)_{i \leq d} \in K^d$, or the coordinates matrix

of a basis with respect to an integral basis $(r_i)_i$ of K. This coordinates matrix belongs to $\mathbb{Q}^{d\times d}$, and it may prove interesting to find the basis of kI such that the coordinates matrix is in HNF, for the smallest non-zero integer k such that kI is integral. This representation requires a space of $O(d\log\mathcal{N}(kI)+\log k+d^2)=O(d\log\mathcal{N}(I)+d^2+d^2\log k)$ bits. Alternatively, one may use the so-called two-element representation: any ideal I may be written $I=(x_1)+(x_2)$ for some $x_1,x_2\in I$. A classical way to obtain such a representation consists in taking an arbitrary $x_1\in I$ and then choosing x_2 uniformly in I modulo (x_1) (the latter being a full-rank sublattice of the former). This succeeds with probability $\geq\prod(1-1/\mathcal{N}(\mathfrak{p}))$, where the product is taken over the prime ideals $\mathfrak{p}$ that divide $(x_1)/I$ (see [2, Le. 6.14]). If $\mathcal{N}(x_1)/\mathcal{N}(I)$ is small and if there do not exist too many prime ideals of small norm, then the success probability is large. Belabas [2, Alg. 6.15] proposed a probabilistic polynomial time variant, which always succeeds with high probability. However, the obtained representation of I may still be of bit-size $\Omega(d\log\mathcal{N}(I)+d+d^2\log k)$.

We modify Belabas' algorithm to provide a 2-element representation made of small elements: $I=(x_1)+(x_2)$ with both $\|x_1\|$ and $\|x_2\|$ small. For instance, the first element x_1 is chosen to be the first element of a `LatRed`-reduced basis of I. This may be seen as a rigorous variant of [7, Alg. 1.3.15], in which smallness was provided but the success probability could be small. Although our analysis is close to Belabas', we give a full proof, as there are quite a few small differences.

Theorem 3. *Let $(r_i)_i$ be an integral basis of a number field K. There exists a probabilistic polynomial time algorithm that takes as inputs a $\mathbb{Z}$-basis of a non-zero fractional ideal I of $\mathcal{O}_K$ and a success parameter t (in unary), and returns $x_1,x_2\in I$ such that $I=(x_1)+(x_2)$ holds with probability $1-2^{-t}$, and:*

$$\|x_1\|,\|x_2\|\leq 4\gamma^8\Delta_K^{\frac{4}{d}}\max\|r_i\|^4\cdot\mathcal{N}(I)^{\frac{4}{d}}, \tag{1}$$

where $\|\cdot\|$ corresponds to the T_2 norm and γ is the `LatRed` *approximation constant. As a consequence, the ideal I may be represented on $5\log_2\mathcal{N}(I)+O(\log\Delta_K+d(d+\log k+\log\max\|r_i\|))$ bits, where k is the smallest non-zero integer such that kI is integral and the r_i's are assumed* `LatRed`*-reduced.*

Let us comment on (1). The quantity $4\gamma^8\Delta_K^{\frac{4}{d}}$ is an invariant of the field, and $\max\|r_i\|^4$ is independent from I (and can be bounded using Lemma 1). The only term that is not an invariant is $\mathcal{N}(I)^{\frac{4}{d}}$. If x_1 and x_2 were basis vectors of a reduced basis of I, we would expect $\mathcal{N}(I)^{\frac{1}{d}}$ instead of $\mathcal{N}(I)^{\frac{4}{d}}$ (see (2) below). We do not know how to reach this bound for x_2.

Let us now prove Theorem 3. Since the smallest integer k such that kI is integral can be computed efficiently, we assume that I is integral. As the ideal I is given by a $\mathbb{Z}$-basis, we can find a basis of it that is `LatRed`-reduced (for T_2). The algorithm of Figure 1 is an adaptation of [2, Alg. 6.15]. We follow the

Inputs: A `LatRed`-reduced basis of a non-zero integral ideal I of $\mathcal{O}_K$;
a success parameter t.
Output: $x_1, x_2 \in I$ such that $I = (x_1) + (x_2)$, or `Fail`.

1. Let x_1 be the first basis element; $\mathfrak{a} := (x_1)$. If $I = \mathfrak{a}$, return x_1 and $x_2 := 0$.
2. Find y such that $y \log y = \log \mathcal{N}(\mathfrak{a})$; $S := \{\mathfrak{p} \text{ prime} : \mathcal{N}(\mathfrak{p}) \leq y\}$.
3. $\mathfrak{a}_0 := \prod_{\mathfrak{p} \in S} \mathfrak{p}^{\nu_\mathfrak{p}(\mathfrak{a})}$; $I_0 := \prod_{\mathfrak{p} \in S} \mathfrak{p}^{\nu_\mathfrak{p}(I)}$; $\mathfrak{a}_1 := \mathfrak{a}\mathfrak{a}_0^{-1}$; $I_1 := I I_0^{-1}$.
4. For $i := 1$ to $2t$ do
5. Sample π_1 uniformly in $I_1/\mathfrak{a}_1$. If $I_1 = \mathfrak{a}_1 + (\pi_1)$, then go to Step 7.
6. Return `Fail`.
7. Let b be the first element of a `LatRed`-reduced basis of $\mathfrak{a}_1$.
8. Size-reduce π_1 with respect to the $b \cdot r_i$'s.
9. Using [2, Alg. 6.8], find $\pi_0 \in \mathcal{O}_K$ such that $\nu_\mathfrak{p}(\pi_0) = \nu_\mathfrak{p}(I_0)$ for all $\mathfrak{p} \in S$.
10. Let b be the first element of a `LatRed`-reduced basis of $\prod_{\mathfrak{p} \in S} \mathfrak{p}^{\nu_\mathfrak{p}(I_0)+1}$.
11. Size-reduce π_0 with respect to the $b \cdot r_i$'s.
12. Using [2, Alg. 5.4], find $\alpha_0 \in \mathfrak{a}_0$ and $\alpha_1 \in \mathfrak{a}_1$ such that $\alpha_0 + \alpha_1 = 1$.
13. Let b be the first element of a `LatRed`-reduced basis of $\mathfrak{a}$.
14. Size-reduce α_0 and α_1 with respect to the $b \cdot r_i$'s.
15. Return x_1 and $x_2 := (\pi_0\alpha_1 + \alpha_0)(\pi_1\alpha_0 + \alpha_1)$.

Fig. 1. Computing a small 2-element representation of an integral ideal

algorithm step by step. The reducedness of the input directly gives that $\|x_1\| \leq \gamma \Delta_K^{1/2d} \mathcal{N}(I)^{1/d}$. By using the arithmetic-geometric inequality, we obtain:

$$\mathcal{N}(\mathfrak{a})^{\frac{1}{d}} = \mathcal{N}(x_1)^{\frac{1}{d}} \leq \frac{1}{\sqrt{d}} \|x_1\| \leq \frac{\gamma \Delta_K^{\frac{1}{2d}}}{\sqrt{d}} \mathcal{N}(I)^{\frac{1}{d}}. \tag{2}$$

As a consequence, the variable y of Step 2, can be bounded by a polynomial in d, $\log \mathcal{N}(I)$ and $\log \Delta_K$. This ensures that the computation of S can be done in polynomial time. At Step 3, the computations of $\mathfrak{a}_0$, I_0, $\mathfrak{a}_1$ and I_1 can be performed in polynomial time: this follows from the above study of S. We have $\mathfrak{a} = \mathfrak{a}_0\mathfrak{a}_1$ and $I = I_0 I_1$. We also have $I_i | \mathfrak{a}_i$ and $I_i + \mathfrak{a}_{1-i} = \mathcal{O}_K$ for $i \in \{0, 1\}$.

As $\mathfrak{a}_1$ is a full-rank sublattice of I_1, sampling π_1 uniformly in $I_1/\mathfrak{a}_1$ can be done in polynomial time. The equality $I_1 = \mathfrak{a}_1 + (\pi_1)$ can also be tested in polynomial time (see, e.g., [20, Prop. 8.2]). By adapting the analysis of [2, Le. 6.1], we obtain:

$$\Pr\left[I_1 = \mathfrak{a}_1 + (\pi_1)\right] \geq \prod_{\mathfrak{p} \text{ prime},\ \mathfrak{p} | \mathfrak{a}_1} \left(1 - \frac{1}{\mathcal{N}(\mathfrak{p})}\right) \geq \left(1 - \frac{1}{y}\right)^{\log_y \mathcal{N}(\mathfrak{a})} \geq \frac{1}{\mathrm{e}}.$$

As a consequence, the algorithm returns `Fail` at Step 6 with probability $\leq 2^{-t}$.

At Step 8, the $b \cdot r_i$'s are a basis of a sublattice of $\mathfrak{a}_1$. Therefore, after Step 8, we still have $I_1 = \mathfrak{a}_1 + (\pi_1)$. After the size-reduction of π_1 with respect to the $b \cdot r_i$'s, we have:

$$\|\pi_1\| \leq \sqrt{d} \max_i \|b r_i\| \leq \sqrt{d} \|b\| \max_i \|r_i\| \leq \sqrt{d} \gamma \Delta_K^{\frac{1}{2d}} \mathcal{N}(\mathfrak{a}_1)^{\frac{1}{d}} \max_i \|r_i\|.$$

It is shown in [2] that Step 9 can be performed in polynomial time. The bounds on S imply that Step 10 can be done in polynomial time. Step 11 ensures that

$$\|\pi_0\| \leq \sqrt{d}\gamma\Delta_K^{\frac{1}{2d}}\mathcal{N}\left(\prod_{\mathfrak{p}\in S}\mathfrak{p}^{\nu_\mathfrak{p}(I_0)+1}\right)^{\frac{1}{d}}\max_i\|r_i\| \leq \sqrt{d}\gamma\Delta_K^{\frac{1}{2d}}\mathcal{N}(I_0)^{\frac{2}{d}}\max_i\|r_i\|.$$

After Step 11, we still have that $\nu_\mathfrak{p}(\pi_0) = \nu_\mathfrak{p}(I_0)$, for all $\mathfrak{p} \in S$, and thus $I_0 = \mathfrak{a}_0 + (\pi_0)$. It is shown in [2] that Step 12 can be performed in polynomial time. Step 14 ensures that $\|\alpha_0\|, \|\alpha_1\| \leq \sqrt{d}\gamma\Delta_K^{\frac{1}{2d}}\mathcal{N}(\mathfrak{a})^{\frac{1}{d}}\max_i\|r_i\|$. Since $\mathfrak{a} = \mathfrak{a}_0\mathfrak{a}_1$, we still have $\alpha_i \in \mathfrak{a}_i$ after Step 14, for $i \in \{0,1\}$. At Step 15, we have:

$$\begin{aligned}\|x_2\| &\leq (\|\pi_0\|\|\alpha_1\| + \|\alpha_0\|)\,(\|\pi_1\|\|\alpha_0\| + \|\alpha_1\|)\\ &\leq d^2\gamma^4\Delta_K^{\frac{2}{d}}\max_i\|r_i\|^4\mathcal{N}(\mathfrak{a})^{\frac{2}{d}}\left(\mathcal{N}(I_0)^{\frac{2}{d}}+1\right)\left(\mathcal{N}(\mathfrak{a}_1)^{\frac{1}{d}}+1\right)\\ &\leq 4d^2\gamma^4\Delta_K^{\frac{2}{d}}\max_i\|r_i\|^4\mathcal{N}(\mathfrak{a})^{\frac{4}{d}},\end{aligned}$$

where we used the fact that $\mathcal{N}(\mathfrak{a}_1) = \mathcal{N}(\mathfrak{a})/\mathcal{N}(\mathfrak{a}_0) \leq \mathcal{N}(\mathfrak{a})/\mathcal{N}(I_0)$. Combining the latter with (2) provides the upper bound on $\|x_2\|$ from Theorem 3.

Also, we have that $\pi_i' := \pi_i\alpha_{1-i} + \alpha_i$ is congruent to π_i modulo $\mathfrak{a}_i$ and to 1 modulo $\mathfrak{a}_{1-i}$, for $i \in \{0,1\}$. Therefore, we have $I_i = \mathfrak{a}_i + (\pi_i')$ and $I_i + (\pi_{i-1}') = \mathcal{O}_K$. Finally, we obtain $I = I_0I_1 = \mathfrak{a}_0\mathfrak{a}_1 + (\pi_0'\pi_1') = (x_1) + (x_2)$, thus proving the correctness of the algorithm.

We now consider the amount of space needed to represent the coordinates of x_1 and x_2 with respect to the integral basis $(r_i)_i$. We write $x_j = \sum y_i^{(j)}r_i$ with $y_i^{(j)} \in \mathbb{Z}$ and $j \in \{1,2\}$. Using Lemma 2, we have that each $y_i^{(j)}$ may be stored on $\log_2\|x_j\| + O(d)$ bits. Combining the latter with (2) and (1) provides the result. □

4 Computing Short Pseudo-bases

In this section, we (constructively) show that any $\mathcal{O}_K$-module $M \subseteq K_\mathbb{R}^m$ always has a pseudo-basis $[(\mathbf{b}_i)_i, (\mathfrak{b}_i)_i]$ such that the $\mathbf{b}_i$'s belong to M and are not much longer than the module minima.

4.1 From a Short Basis of a Submodule to a Short Pseudo-basis

We are going to generalize to $\mathcal{O}_K$-modules the technique we mentioned at the end of Section 2.1, that takes as inputs a basis of a lattice L and a short basis of a full-rank sub-lattice of L, and returns a short basis of L. We split the algorithm into several smaller ones that may be of independent interest.

The algorithm of Figure 2 takes as inputs a pseudo-basis $[(\mathbf{a}_i)_i, (\mathfrak{a}_i)_i]$ of an $\mathcal{O}_K$-module $M \subseteq K_\mathbb{R}^m$ and a full-rank set of short module vectors $(\mathbf{s}_i)_i$, and returns a pseudo-basis $[(\mathbf{b}_i)_i, (\mathfrak{b}_i)_i]$ of M such that $\mathbf{b}_i \in \mathrm{span}_{j\leq i}\,\mathbf{s}_j$. This can be interpreted

Inputs: A pseudo-basis $[(\mathbf{a}_i)_i, (\mathfrak{a}_i)_i]$ of an $\mathcal{O}_K$-module $M \subseteq K_{\mathbb{R}}^m$,
a full-rank set $(\mathbf{s}_i)_i$ of vectors in M.
Output: A pseudo-basis of M.
1. Compute $T \in K^{n\times n}$ such that $(\mathbf{s}_1, \ldots, \mathbf{s}_n) = (\mathbf{a}_1, \ldots, \mathbf{a}_n)T$.
2. Let $\mathbf{t}_1, \ldots, \mathbf{t}_n$ be the columns of T^t.
3. Compute the BPC-HNF $[(\mathbf{t}'_i)_i, (\mathfrak{b}_i^{-1})_i]$ of the pseudo-basis $[(\mathbf{t}_i)_i, (\mathfrak{a}_i^{-1})_i]$.
4. Let T' be the matrix whose rows are the $(\mathbf{t}'_i)^t$'s, and $U = T(T')^{-1} \in K^{n\times n}$.
5. Let $(\mathbf{b}_1, \ldots, \mathbf{b}_n) = (\mathbf{a}_1, \ldots, \mathbf{a}_n)U$.
6. Return $[(\mathbf{b}_i)_i, (\mathfrak{b}_i)_i]$.

Fig. 2. Constructing a pseudo-basis with small GSO

as a constructive variant of [23, Th. 81.3]. The HNF over lattices is replaced by the BPC-HNF (Theorem 1), with special care being taken for the coefficient ideals.

Theorem 4. *If given as inputs a pseudo-basis* $[(\mathbf{a}_i)_i, (\mathfrak{a}_i)_i]$ *of a module* $M \subseteq K_{\mathbb{R}}^m$ *and a full-rank set* $(\mathbf{s}_i)_i$ *of vectors in* M*, then the algorithm of Figure 2 returns a pseudo-basis* $[(\mathbf{b}_i)_i, (\mathfrak{b}_i)_i]$ *of* M*, which satisfies, for all* $i \leq n$*:* $\mathbf{b}_i \in M$*;* $\mathbf{b}_i \in \mathrm{span}_{j\leq i}\, \mathbf{s}_j$*;* $\mathbf{b}_i^* = \mathbf{s}_i^*$*. If* $M \subseteq K^m$*, then it terminates in polynomial time.*

Proof. We first prove that $[(\mathbf{b}_i)_i, (\mathfrak{b}_i)_i]$ is a pseudo-basis of M. We have $(\mathbf{b}_i)_i = (\mathbf{a}_i)_i \cdot U$, with $U \in K^{n\times n}$ non-singular. It therefore suffices to prove that for any i, j, we have $U_{i,j} \in \mathfrak{a}_i\mathfrak{b}_j^{-1}$ and $U'_{i,j} \in \mathfrak{b}_i\mathfrak{a}_j^{-1}$, where $U' = U^{-1}$. This is ensured by Theorem 1: as the pseudo-bases $[(\mathbf{t}'_i)_i, (\mathfrak{b}_i^{-1})_i]$ and $[(\mathbf{t}_i)_i, (\mathfrak{a}_i^{-1})_i]$ span the same module, we have $U'_{j,i} \in \mathfrak{a}_i^{-1}\mathfrak{b}_j$ and $U_{j,i} \in \mathfrak{b}_i^{-1}\mathfrak{a}_j$, for any i, j.

Because of the definitions of T, T', U and $(\mathbf{b}_i)_i$, we have $(\mathbf{s}_i)_i = (\mathbf{b}_i)_i \cdot T'$. Furthermore, by Theorem 1, the matrix T' is upper triangular with diagonal coefficients equal to 1. We thus have $\mathbf{b}_i \in \mathrm{span}_{j\leq i}\, \mathbf{s}_j$, for all i. In fact, we even have $\mathbf{b}_i + \sum_{j<i} K_{\mathbb{R}}\mathbf{b}_j = \mathbf{s}_i + \sum_{j<i} K_{\mathbb{R}}\mathbf{s}_j$, which gives $\mathbf{b}_i^* = \mathbf{s}_i^*$. Finally, the shape of T' gives that $\mathbf{s}_i = \mathbf{b}_i + \sum_{j<i} T'_{j,i}\mathbf{b}_j$. As the $\mathbf{s}_i$'s belong to M, so must the $\mathbf{b}_i$'s (the decomposition of $\mathbf{s}_i$ as an element of $\sum_j K\mathbf{b}_j$ is unique). □

The algorithm of Figure 3 generalizes size-reduction to $\mathcal{O}_K$-modules.

Theorem 5. *If given as input a pseudo-basis* $[(\mathbf{a}_i)_i, (\mathfrak{a}_i)_i]$ *of a module* $M \subseteq K_{\mathbb{R}}^m$*, then the algorithm of Figure 3 returns a pseudo-basis* $[(\mathbf{b}_i)_i, (\mathfrak{b}_i)_i]$ *of* M*, such that for all* i *we have* $\mathbf{b}_i^* = \mathbf{a}_i^*$*,* $\mathfrak{b}_i = \mathfrak{a}_i$ *and*

$$\|\mathbf{b}_i\| \leq \sqrt{dn}\gamma\Delta_K^{\frac{1}{2d}} \max_k \|r_k\| \left(\frac{\max_{j\leq i} \mathcal{N}(\mathfrak{b}_j)}{\mathcal{N}(\mathfrak{b}_i)}\right)^{\frac{1}{d}} \max_{j\leq i} \|\mathbf{a}_j^*\|.$$

If $M \subseteq K^m$ *and* `LatRed` *is LLL, then it terminates in polynomial time.*

Proof. The operations performed on the pseudo-basis can be checked to preserve the generated module and the $\mathbf{b}_i^*$'s. Steps 2, 6 and 7 ensure that the $\mu_{i,j}$'s of the output pseudo-basis satisfy $\|\mu_{i,j}\| \leq \sqrt{d}\gamma\Delta_K^{\frac{1}{2d}}\mathcal{N}(\mathfrak{b}_i^{-1}\mathfrak{b}_j)^{\frac{1}{d}} \max_k \|r_k\|$. Pythagoras' theorem then provides the result. □

Input: A pseudo-basis $[(\mathbf{a}_i)_i, (\mathfrak{a}_i)_i]$ of an $\mathcal{O}_K$-module $M \subseteq K_{\mathbb{R}}^m$.
Output: A pseudo-basis of M.
1. $[(\mathbf{b}_i)_i, (\mathfrak{b}_i)_i] := [(\mathbf{a}_i)_i, (\mathfrak{a}_i)_i]$.
2. For $j \leq i$, let $x_{i,j}$ be the first element of a `LatRed` basis of $\mathfrak{b}_i^{-1}\mathfrak{b}_j$.
3. For i from 2 to n, do
4. For j from $i-1$ to 1, do
5. Compute the GSO decomposition $\mathbf{b}_i = \mathbf{b}_i^* + \sum_{j<i} \mu_{i,j}\mathbf{b}_j^*$,
6. Let y be the size-reduction of $\mu_{i,j}$ with respect to the $x_{i,j}r_k$'s,
7. $\mathbf{b}_i := \mathbf{b}_i - (\mu_{i,j} - y)\mathbf{b}_j$.
8. Return $[(\mathbf{b}_i)_i, (\mathfrak{b}_i)_i]$.

Fig. 3. Size-reducing a pseudo-basis of an $\mathcal{O}_K$-module

The adaptation to $\mathcal{O}_K$-modules of [20, Le. 7.1] is given in Figure 4. The aim of Steps 2–4 is to allow us to bound the term $\frac{\max_{j\leq i}\mathcal{N}(\mathfrak{b}_j)}{\mathcal{N}(\mathfrak{b}_i)}$ from Theorem 5.

Inputs: A pseudo-basis $[(\mathbf{a}_i)_i, (\mathfrak{a}_i)_i]$ of an $\mathcal{O}_K$-module $M \subseteq K_{\mathbb{R}}^m$,
a free full-rank set $(\mathbf{s}_i)_i$ of vectors in M.
Output: A pseudo-basis of M.
1. Use the algorithm of Figure 2 to obtain a pseudo-basis $[(\mathbf{b}_i)_i, (\mathfrak{b}_i)_i]$ of M.
2. For any $i \leq n$,
3. Let $x \in \mathfrak{b}_i$ be the first vector of a `LatRed` basis of $\mathfrak{b}_i$,
4. $\mathfrak{b}_i := (x)^{-1}\mathfrak{b}_i$; $\mathbf{b}_i := x\mathbf{b}_i$.
5. Return the output of the algorithm of Figure 3, given $[(\mathbf{b}_i)_i, (\mathfrak{b}_i)_i]$ as input.

Fig. 4. From small vectors to a small pseudo-basis

Theorem 6. *If given as inputs a pseudo-basis* $[(\mathbf{a}_i)_i, (\mathfrak{a}_i)_i]$ *of an* $\mathcal{O}_K$*-module* $M \subseteq K_{\mathbb{R}}^m$ *and a full-rank set* $(\mathbf{s}_i)_i$ *of vectors in* M*, then the algorithm of Figure 4 returns a pseudo-basis* $[(\mathbf{b}_i)_i, (\mathfrak{b}_i)_i]$ *of* M*, such that for all* i*:* $\mathbf{b}_i \in M$, $\mathrm{span}_{j\leq i}\,\mathbf{b}_j = \mathrm{span}_{j\leq i}\,\mathbf{s}_j$, $\|\mathbf{b}_i^*\| \leq \gamma\Delta_K^{\frac{1}{2d}}\|\mathbf{s}_i^*\|$, $\mathcal{N}(\mathfrak{b}_i) \in \left[\left(\frac{\sqrt{d}}{\gamma}\right)^d \frac{1}{\sqrt{\Delta_K}}, 1\right]$ *and*

$$\|\mathbf{b}_i\| \leq \sqrt{n}\gamma^3\Delta_K^{\frac{3}{2d}} \max_k \|r_k\| \cdot \max_{j\leq i}\|\mathbf{s}_j\|.$$

If $M \subseteq K^m$ *and* `LatRed` *is LLL, then it terminates in polynomial time.*

Proof. The fact that the algorithm returns a pseudo-basis of M is easy to check. Also, at the end of Step 1, we have that $\mathbf{b}_i \in M$, for all i. Since the x of Step 3 belongs to $\mathfrak{b}_i$, the latter fact is preserved throughout the rest of the execution. The equality $\mathrm{span}_{j\leq i}\,\mathbf{b}_j = \mathrm{span}_{j\leq i}\,\mathbf{s}_j$ directly derives from Theorems 4 and 5.

At any time after Step 1, we have $\mathcal{O}_K \subseteq \mathfrak{b}_i$ and thus $\mathcal{N}(\mathfrak{b}_i) \leq 1$. At Step 3, we have $\|x\| \leq \gamma \Delta_K^{\frac{1}{2d}} \mathcal{N}(\mathfrak{b}_i)^{\frac{1}{d}}$. This gives that after Step 4 we have $\|\mathbf{b}_i^*\| \leq \gamma \Delta_K^{\frac{1}{2d}} \|\mathbf{s}_i^*\|$, which is preserved throughout Step 5. Also, the arithmetic-geometric inequality implies that $\mathcal{N}(x) \leq (\gamma/\sqrt{d})^d \sqrt{\Delta_K} \mathcal{N}(\mathfrak{b}_i)$. Therefore, after Step 4 the quantity $\mathcal{N}(\mathfrak{b}_i)$ has been divided by $\mathcal{N}(x)$ and we have $\mathcal{N}(\mathfrak{b}_i) \geq \left(\frac{\sqrt{d}}{\gamma}\right)^d \frac{1}{\sqrt{\Delta_K}}$. Using Theorem 5, this allows us to derive that at the end of the execution we have:

$$\|\mathbf{b}_i\| \leq \sqrt{dn} \gamma \Delta_K^{\frac{1}{2d}} \max_k \|r_k\| \left(\frac{\sqrt{\Delta_K}}{(\sqrt{d}/\gamma)^d}\right)^{\frac{1}{d}} \cdot \left(\gamma \Delta_K^{\frac{1}{2d}} \max_{j \leq i} \|\mathbf{s}_j^*\|\right).$$

The inequalities $\|\mathbf{s}_j^*\| \leq \|\mathbf{s}_j\|$ lead to the result. □

4.2 Computing a Short Pseudo-basis

Suppose we have a pseudo-basis of an $\mathcal{O}_K$-module M of rank n. We can expand it to obtain a basis of M as a $\mathbb{Z}$-module. By LLL-reducing the latter with respect to T_2, we obtain dn module vectors whose integer linear combinations span M. By using linear algebra over K, it is possible to select n module vectors $\mathbf{s}_1, \ldots, \mathbf{s}_n$ among these dn vectors, such that $\mathrm{rank}_K(\mathbf{s}_i) = n$. Furthermore, thanks to the initial reduction, these vectors are also small, and we can apply Theorem 6.

Corollary 1. *There exists an algorithm that takes as input a pseudo-basis of an $\mathcal{O}_K$-module $M \subseteq K_{\mathbb{R}}^m$ and returns a pseudo-basis $[(\mathbf{b}_i)_i, (\mathfrak{b}_i)_i]$ of M, such that for all i: $\mathbf{b}_i \in M$, $\mathcal{N}(\mathfrak{b}_i) \in \left[\left(\frac{\sqrt{d}}{\gamma}\right)^d \frac{1}{\sqrt{\Delta_K}}, 1\right]$ and*

$$\|\mathbf{b}_i\| \leq 2^{\frac{dn}{2}} \sqrt{n} \gamma^3 \Delta_K^{\frac{3}{2d}} \max_k \|r_k\|^2 \cdot \lambda_i(M).$$

Therefore:

$$\prod_i \|\mathbf{b}_i\| \leq 2^{\frac{dn^2}{2}} (\sqrt{dn})^n \gamma^{3n} \Delta_K^{\frac{3n}{2d}} \max_k \|r_k\|^{2n} \cdot (\det(M))^{\frac{1}{d}}.$$

If $M \subseteq K^m$ and `LatRed` *is LLL, then it terminates in polynomial time, and the output may be stored on a number of bits bounded by*

$$m \log_2 \det(M) + O\left(md^2n^2 + nm \log \Delta_K + mdn \log \max_k \|r_k\|\right).$$

Proof. Let L denote M when considered as a lattice. Let $(\mathbf{s}_i)_{i \leq dn}$ be a LLL-reduced basis of L. We have $\|\mathbf{s}_i\| \leq 2^{dn/2} \lambda_i(L)$, for all i. Let $\psi(i) = \min(j : \mathrm{rank}_K(\mathbf{s}_k)_{k \leq j} = i)$. Since K has degree d, we have $\psi(i) \leq d(i-1)+1$, for all i. We use the $\mathbf{s}_{\psi(i)}$'s as input to the algorithm of Figure 4. The first statement on the $\|\mathbf{b}_i\|$'s derives from Theorem 6 and the fact that $\lambda_{\psi(i)}(L) \leq \max_k \|r_k\| \cdot \lambda_{\lceil \psi(i)/d \rceil}(M) \leq \max_k \|r_k\| \cdot \lambda_i(M)$. By combining Theorem 2 and the latter, we obtain the second statement on the $\|\mathbf{b}_i\|$'s.

We now consider the bit-size of the representation when $M \subseteq K^m$. Using Lemma 2 for the first component of the pseudo-basis, we obtain that the bit-size of the latter is $\leq md(\log_2 \prod \|\mathbf{b}_i\| + O(nd))$. To represent the ideal coefficients, we use Theorem 3 with the inverses of the ideals. The latter are integral, and have norms $\leq 2^{d^2} \Delta_K$. Therefore, each of these can be represented on $O(d^2 + \log \Delta_K + d \log \max_k \|r_k\|)$ bits. This completes the proof of the theorem. □

Note that the norm bound on the ideals depends only on the field and the choice for `LatRed` and is, in particular, independent of M.

By applying Corollary 1 with $m = n = 1$, we obtain yet another compact representation of ideals of K. If I is an ideal and k is the smallest non-zero integer such that kI is integral, then we see that I can be represented on $\log_2 \mathcal{N}(I) + O(\log \Delta_K + d(d + \log k + \log \max_i \|r_i\|))$ bits. If $\mathcal{N}(I)$ is large, this representation is smaller than the one from Theorem 3, but for a small $\mathcal{N}(I)$, this is the opposite as the $O(\cdot)$ constant is larger. Considering $((x_1) + (x_2))$ instead of its inverse leads to a representation whose bit-size grows faster with respect to d.

4.3 Short almost Free Pseudo-bases

A common strengthening of the properties of a pseudo-basis is to pass to an almost free (or Steinitz) representation: For any M, there exists a pseudo-basis $[(\mathbf{b}_i)_i, (\mathfrak{b}_i)_i]$ of M with $\mathfrak{b}_i = \mathcal{O}_K$ for $i < n$. We explain here how to obtain an almost free pseudo-basis consisting of short vectors. We first use Corollary 1 to find a "short" pseudo-basis. We then use the following lemma, from [7, Prop. 1.3.12, Alg. 1.3.16], which allows us to pass from a module with coefficient ideals $(\mathfrak{a}, \mathfrak{b})$ to a representation of this module with ideals $(1, \mathfrak{ab})$.

Lemma 3. *Let $\mathfrak{a}$ and $\mathfrak{b}$ be non-zero fractional ideals. There exists a polynomial-time algorithm that finds $a \in \mathfrak{a}$, $b \in \mathfrak{b}$, $x \in \mathfrak{a}^{-1}$, $y \in \mathfrak{b}^{-1}$ such that $ax - by = 1$.*

One can use Lemma 3 to progressively change the short pseudo-basis obtained in Corollary 1 into a short almost free pseudo-basis, collecting all the coefficient ideals into the last one. The corresponding algorithm is given in Figure 5. It can be checked that the output is an almost free pseudo-basis of the input module.

Furthermore, if the input of the algorithm is a module pseudo-basis such as in Corollary 1, then during the execution, Lemma 3 is applied to ideals whose norms can be bounded independently of the module M. As a consequence, the

Input: A pseudo-basis $[(\mathbf{a}_i)_i, (\mathfrak{a}_i)_i]$ of an $\mathcal{O}_K$-module $M \subseteq K_{\mathbb{R}}^m$.
Output: An almost free pseudo-basis of M.
1. For $i = 1$ to $n - 1$ do
2. Use Lemma 3 with $\mathfrak{a} := \mathfrak{a}_i$, $\mathfrak{b} := \mathfrak{a}_{i+1}$ to find a, b, x, y as indicated,
3. Replace $\mathbf{a}_i$ by $a\mathbf{a}_i + b\mathbf{a}_{i+1}$ and $\mathbf{a}_{i+1}$ by $y\mathbf{a}_i + x\mathbf{a}_{i+1}$,
4. Replace $\mathfrak{a}_{i+1}$ by $\mathfrak{a}_i\mathfrak{a}_{i+1}$ and $\mathfrak{a}_i$ by $\mathcal{O}_K$.

Fig. 5. From a pseudo-basis to an almost free pseudo-basis

obtained transformation coefficients a, b, x, y have T_2-norms that can be bounded independently of M. At the end of the execution, we still have $\mathbf{a}_i \in M$ for all i, and the quantity $\prod_i \|\mathbf{a}_i\|$ (resp. each $\|\mathbf{a}_i\|$) remains bounded by $\det(M)^{\frac{1}{d}}$ (resp. by the corresponding $\lambda_i(M)$) up to a multiplicative factor that is independent of M. Similarly, the norm of the non-trivial coefficient ideal can also be bounded independently of M.

Finally, it should be noted that the basis generated by the algorithm of Figure 5 satisfies $\mathbf{b}_i \in \mathrm{span}_{j \le i+1} \mathbf{a}_j$ for $i < n$, and thus can be compared to the results from [8].

5 Examples

We start by some example coming from group theory, focusing only on the use of lattice reduction. Representations of finite groups give easy access to non-trivial and interesting lattices. In general starting with a finite subgroup $G < \mathrm{GL}(m, K)$ and any $\mathcal{O}_K$-module N we obtain a G-invariant $\mathcal{O}_K$-module M via $M := \sum_{g \in G} Ng$. Next we change G to act on M, $G \in \mathrm{GL}(M)$ and, fixing a complex conjugation on K, obtain a G-invariant Hermitean form on K^m from $H := \sum_{g \in G} g^* g$. The main application is to find a reduced (short) pseudo-basis $S = MT$ for M and then replace G by $G^T = \{T^{-1} g T : g \in G\}$ to find an isomorphic version of G where the elements are (hopefully) "smaller".

Let G be the quaternion group Q_8 with 8 elements. As a subgroup of $\mathrm{GL}(2, K)$ for $K := \mathbb{Q}(i)$, it can be generated by

$$\frac{1}{5}\begin{pmatrix} i+2 & 2i-6 \\ 2i+4 & -i-2 \end{pmatrix} \quad \text{and} \quad \frac{1}{2}\begin{pmatrix} -i-1 & 3i+1 \\ i-1 & i+1 \end{pmatrix}.$$

Computing the $\mathcal{O}_K$-module generated by $g\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ for all $g \in G$, we use $M := \mathcal{O}_K \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \left(\frac{1+3i}{10}\mathcal{O}_K\right)\begin{pmatrix} 3 \\ 1 \end{pmatrix}$. As a Hermitean form, we compute $\sum_{g \in G} gg^*$ where g^* denotes the transposed complex conjugate. We then normalize the matrix to have 1 as the top left entry and obtain

$$H := \frac{1}{5}\begin{pmatrix} 5 & i+2 \\ -i+2 & 3 \end{pmatrix}.$$

We reduce the corresponding $\mathbb{Z}$-lattice and use the following short $\mathbb{Q}(i)$-independent basis elements:

$$\frac{1}{10}\begin{pmatrix} -3i-1 \\ -i+3 \end{pmatrix} \quad \text{and} \quad -\frac{1}{5}\begin{pmatrix} 2i-1 \\ -i+3 \end{pmatrix}.$$

The two elements can be seen to freely generate the module. Using the transformation to change G, we now get

$$\begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

which is a "nicer" version of G.

Let $G := SZ_8$ the 8th Suzuki group with 29 120 elements. This group has 11 characters, and we consider the second among them. The latter defines a representation of degree 14 over some field containing i. For theoretical reasons, the representation can be defined over $\mathbb{Q}(i)$, but it is initially computed over $\mathbb{Q}(\zeta_{52})$, of degree 24. A complicated procedure will now find a representation over $\mathbb{Q}(i)$, i.e., we have three matrices (one for each generator) over $\mathbb{Q}(i)$ generating G. The coefficients of the original matrix entries over $\mathbb{Q}(\zeta_{52})$ have about 100 digits each, and over $\mathbb{Q}(i)$ this increases to about 200 digits. In this representation the group G fixes a Hermitean form M which has again entries with about 200 digits each. Since the representation is absolutely irreducible, the quadratic form is unique up to multiplication by scalars. We normalized the form to have 1 as the entry in position $(1, 1)$. After application of our reduction technique, the form as well as the representation now have only 1 digit entries. The module used here is generated by $Ge_1 \subseteq \mathbb{Q}(i)^2$.

We used the following Magma code to generate the second example:

```
> G := Sz(8);
> T := CharacterTable(G);
> M := GModule(T[2]:SparseCyclo := false);
> N := AbsoluteModuleOverMinimalField(M);
> IsAlmostIntegral(N); //computes the module
true
> _ := InvariantForm(N);  // compute the form
> SetVerbose("RLLL", 1);
> O := Nice(N);
> #Sprint(ActionGenerators(M));
1359862
> #Sprint(ActionGenerators(N));
327378
> #Sprint(ActionGenerators(O));
4577
```

The function `Nice` implements the procedure outlined above. Note that the actual result can vary substantially as several parts use randomized algorithms. The `Sprint` statements are only used as a very crude indication of the output size, they simply give the number of characters neccessary to write the generating matrices for G.

References

1. Ajtai, M., Kumar, R., Sivakumar, D.: A sieve algorithm for the shortest lattice vector problem. In: Proc. STOC 2001, pp. 601–610. ACM, New York (2001)
2. Belabas, K.: Topics in computational algebraic number theory. J. théorie des nombres de Bordeaux 16, 19–63 (2004)
3. Bosma, W., Cannon, J., Playoust, C.: The Magma algebra system. I. The user language. J. Symbolic Comput. 24(3-4), 235–265 (1997)

4. Bosma, W., Pohst, M.: Computations with finitely generated modules over Dedekind domains. In: Proc. ISSAC 1991, pp. 151–156. ACM, New York (1991)
5. Cohen, H.: A Course in Computational Algebraic Number Theory. Springer, Heidelberg (1995)
6. Cohen, H.: Hermite and Smith normal form algorithms over Dedekind domains. Math. Comp. 65, 1681–1699 (1996)
7. Cohen, H.: Advanced topics in Computational Number Theory. Springer, Heidelberg (2000)
8. Evertse, J.-H.: Reduced bases of lattices over number fields. Indag. Mathem. N.S. 2(3), 153–168 (1992)
9. Fieker, C.: Minimizing representations over number fields II: Computations in the Brauer group. J. Algebra 3(322), 752–765 (2009)
10. Fieker, C., Pohst, M.E.: Lattices over number fields. In: Cohen, H. (ed.) ANTS 1996. LNCS, vol. 1122, pp. 147–157. Springer, Heidelberg (1996)
11. Gan, Y.H., Ling, C., Mow, W.H.: Complex lattice reduction algorithm for low-complexity full-diversity MIMO detection. IEEE Trans. Signal Processing 57, 2701–2710 (2009)
12. Hoppe, A.: Normal forms over Dedekind domains, efficient implementation in the computer algebra system KANT. PhD thesis, Technical University of Berlin (1998)
13. Kannan, R., Bachem, A.: Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix. SIAM J. Comput. 8(4), 499–507 (1979)
14. Lagarias, J.C., Lenstra Jr., H.W., Schnorr, C.P.: Korkine-Zolotarev bases and successive minima of a lattice and its reciprocal lattice. Combinatorica 10, 333–348 (1990)
15. Lenstra, A.K., Lenstra Jr., H.W., Lovász, L.: Factoring polynomials with rational coefficients. Math. Ann. 261, 515–534 (1982)
16. Lovász, L.: An Algorithmic Theory of Numbers, Graphs and Convexity. CBMS-NSF Regional Conference Series in Applied Mathematics. SIAM, Philadelphia (1986)
17. Lyubashevsky, V., Micciancio, D.: Generalized compact knapsacks are collision resistant. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 144–155. Springer, Heidelberg (2006)
18. Magma. The Magma computational algebra system for algebra, number theory and geometry, http://magma.maths.usyd.edu.au/magma/
19. Micciancio, D.: Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. Comput. Complexity 16(4), 365–411 (2007)
20. Micciancio, D., Goldwasser, S.: Complexity of lattice problems: a cryptographic perspective. Kluwer Academic Press, Dordrecht (2002)
21. Mollin, R.A.: Algebraic Number Theory. Chapman and Hall/CRC Press (1999)
22. Napias, H.: A generalization of the LLL-algorithm over Euclidean rings or orders. J. théorie des nombres de Bordeaux 2, 387–396 (1996)
23. O'Meara, O.T.: Introduction to Quadratic Forms. In: Grundlehren der Mathematischen Wissenschaften, vol. 117. Springer, Heidelberg (1963)
24. Peikert, C., Rosen, A.: Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 145–166. Springer, Heidelberg (2006)
25. Peikert, C., Rosen, A.: Lattices that admit logarithmic worst-case to average-case connection factors. In: Proc. STOC 2007, pp. 478–487. ACM, New York (2007)
26. Stehlé, D., Steinfeld, R., Tanaka, K., Xagawa, K.: Efficient public key encryption based on ideal lattices. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 617–635. Springer, Heidelberg (2009)

On the Complexity of the Montes Ideal Factorization Algorithm

David Ford and Olga Veres

Concordia University,
1455 de Maisonneuve Boulevard West,
Montréal, Québec, Canada H3G 1J1
ford@cse.concordia.ca,
overes@mathstat.concordia.ca

Abstract. Let p be a rational prime and let $\Phi(X)$ be a monic irreducible polynomial in $\mathbf{Z}[X]$, with $n_\Phi = \deg \Phi$ and $\delta_\Phi = v_p(\operatorname{disc} \Phi)$. In [13] Montes describes an algorithm for the decomposition of the ideal $p\mathcal{O}_K$ in the algebraic number field K generated by a root of Φ. A simplified version of the Montes algorithm, merely testing $\Phi(X)$ for irreducibility over $\mathbf{Q}_p$, is given in [19], together with a full MAPLE implementation and a demonstration that in the worst case, when $\Phi(X)$ is irreducible over $\mathbf{Q}_p$, the expected number of bit operations for termination is $O(n_\Phi^{3+\epsilon}\delta_\Phi^{2+\epsilon})$. We now give a refined analysis that yields an improved estimate of $O(n_\Phi^{3+\epsilon}\delta_\Phi + n_\Phi^{2+\epsilon}\delta_\Phi^{2+\epsilon})$ bit operations. Since the worst case of the simplified algorithm coincides with the worst case of the original algorithm, this estimate applies as well to the complete Montes algorithm.

1 Introduction

In an algebraic number field K with ring of integers $\mathcal{O}_K$, factorization of the ideal $p\mathcal{O}_K$, for p prime, can be determined via polynomial factorization over the field of p-adic numbers $\mathbf{Q}_p$ [12].

If $K = \mathbf{Q}(\alpha)$ for a given $\alpha \in \mathcal{O}_K$ such that the index $\left[\mathcal{O}_K : \mathbf{Z}[\alpha]\right]$ is not divisible by p then the factorization of the ideal $p\mathcal{O}_K$ can be determined by polynomial factorization modulo p [5,6,7]. In practice, efficient techniques for polynomial factorization modulo p [1,2,4] combined with Hensel lifting [12,20] solve the problem of factoring $p\mathcal{O}_K$ in a straightforward and effective manner when p does not divide the index.

The complications arising when p divides the index $\left[\mathcal{O}_K : \mathbf{Z}[\alpha]\right]$ have been the subject of considerable study. Current ideas are derived from the "Round Four" algorithm of Zassenhaus [20], which has evolved into two main variations, the "one-element" method [8] and the "two-element" method [16]. Versions of the one-element method are used by MAPLE and PARI. The two-element method is used, *e.g.*, by Magma.

The algorithm of Montes [13] is in a separate category.

Given a monic irreducible polynomial $\Phi(X)$ in $\mathbf{Z}[X]$, the Montes algorithm determines the number of irreducible factors of $\Phi(X)$ in $\mathbf{Z}_p[X]$ and their respective degrees. The algorithm exploits classical results of Ore [15,14] on Newton

G. Hanrot, F. Morain, and E. Thomé (Eds.): ANTS-IX 2010, LNCS 6197, pp. 174–185, 2010.

polygons and provides an alternative to the methods based on ideas of Zassenhaus.

A familiar application of Newton polygons gives the p-adic valuations of roots of a polynomial in $\mathbf{Z}_p[X]$. If $\Phi(X) \in \mathbf{Z}_p[X]$ has two roots with different p-adic values then Hensel-lifting techniques can be applied to construct a non-trivial p-adic factorization of Φ to any desired degree of precision.

This process constitutes "level 0" of the Montes algorithm.

For each factor of Φ revealed at level 0, the algorithm proceeds to higher levels, either to discover a refined factorization or to establish irreducibility.

At level r, with $\varphi_r(X)$ an irreducible monic polynomial in $\mathbf{Z}_p[X]$ and V_r a valuation of $\mathbf{Q}_p[X]$, the algorithm constructs the φ_r-adic expansion of a given polynomial and then computes

- a finite field $\mathbf{F}_{q_r}$,
- the Newton polygon $\mathcal{N}_r(\Phi)$ of Φ with respect to the valuation V_r,
- a slope $-d_r/e_r$, with d_r and e_r coprime positive integers, of an edge of $\mathcal{N}_r(\Phi)$,
- the "associated polynomial" $\Psi^{(r)}_{\mathcal{S},\Phi}(Y) \in \mathbf{F}_{q_r}[Y]$ for each segment $\mathcal{S}$ of $\mathcal{N}_r(\Phi)$,
- a monic irreducible factor ψ_r of $\Psi^{(r)}_{\mathcal{S},\Phi}$ with ξ_r a root of ψ_r and $f_r = \deg \psi_r$,
- a valuation V_{r+1} of $\mathbf{Q}_p[X]$,
- an irreducible monic polynomial $\varphi_{r+1}(X) \in \mathbf{Z}_p[X]$.

The number of edges of $\mathcal{N}_r(\Phi)$ and the number of distinct irreducible factors of $\Psi^{(r)}_{\mathcal{S},\Phi}$ give information for the factorization of Φ; if either is greater than one then Φ is reducible.

Our goal being to give an estimate of the complexity of the worst case of the Montes algorithm, we have restricted the algorithm merely to decide the question of irreducibility of a given polynomial. When Φ is irreducible over $\mathbf{Q}_p$ the Newton polygon at each level is a single segment. It is apparent that this is the most costly case, *i.e.*, the case that reaches the highest level, for the full algorithm. So our restricted algorithm operates under the assumption that $\mathcal{N}_r(\Phi)$ has just one edge at each level r; the failure of this condition terminates the restricted algorithm.

In [19, Chapter 3] a complete MAPLE implementation of the restricted Montes algorithm is given, together with a demonstration that in the worst case, when Φ is irreducible over $\mathbf{Q}_p$, the expected number of bit operations for termination is $O(n_\Phi^{3+\epsilon}\delta_\Phi^{2+\epsilon})$, with $n_\Phi = \deg \Phi$ and $\delta_\Phi = v_p(\operatorname{disc} \Phi)$. In the present paper we give a refined analysis that yields an improved estimate of $O(n_\Phi^{3+\epsilon}\delta_\Phi + n_\Phi^{2+\epsilon}\delta_\Phi^{2+\epsilon})$ bit operations. Since the worst case of the simplified algorithm coincides with the worst case of the original algorithm, this estimate applies as well to the full Montes algorithm.

2 Definitions and Notation

Definition 1. *Let $\varphi_0(X) = X$ and let V_0 denote the standard p-adic valuation of $\mathbf{Q}_p$. For $K(X) \in \mathbf{Q}_p[X]$ and $r \geq 1$, the level-r Newton polygon of K, denoted*

$\mathcal{N}_r(K)$, is the Newton polygon of K with respect to the valuation V_r of $\mathbf{Q}_p[X]$, which can be defined recursively as

$$V_r(K) = \min\left\{ e_{r-1}V_{r-1}\big(A_{r-1,k}\big) + kV_r\big(\varphi_{r-1}\big) \;\middle|\; 0 \le k \le n \right\}$$

with $K(X) = \sum_{k=0}^{n} A_{r-1,k}(X)\,\varphi_{r-1}(X)^k$ the φ_{r-1}-adic expansion of $K(X)$.

Remark 1. $\mathcal{N}_r(K)$ is the lower convex hull of the set

$$\left\{ (k, V_r(A_{r,k}\,\varphi_r^k)) \mid 0 \le k \le n,\ A_{r,k}(X) \ne 0 \right\},$$

and if $\deg K < \deg \varphi_r$ then $\mathcal{N}_r(K) = \{(0, V_r(K))\}$ and $V_{r+1}(K) = e_r V_r(K)$.

Definition 2. *For $r \ge 1$ and $K(X)$ a nonzero polynomial in $\mathbf{Z}_p[X]$ we define $\mathcal{S}_{r,K}$ to be the segment of $\mathcal{N}_r(K)$ having slope $-d_r/e_r$.*

Definition 3. *For positive integers r and ν we define*

$$\begin{aligned}
\alpha_{r,\nu} &= \nu d_r^{-1} \bmod e_r\,,\\
\beta_{r,\nu} &= (\nu - \alpha_{r,\nu} d_r)/e_r\,,\\
\mathcal{T}_{r,\nu} &= \left\{ (\alpha_{r,\nu} + \lambda e_r,\ \beta_{r,\nu} - \lambda d_r) \mid 0 \le \lambda \le \lfloor \beta_{r,\nu}/d_r \rfloor \right\}.
\end{aligned}$$

Remark 2. If $\mathcal{L}$ is the line through the point $(0, \nu/e_r)$ with slope $-d_r/e_r$ then $\mathcal{T}_{r,\nu}$ is the longest segment of $\mathcal{L}$ with endpoints having nonnegative integer coordinates.

Definition 4. *For $r \ge 0$ we define*

$$\begin{aligned}
&\overline{\mu}_r = 0\,, && \overline{\nu}_r = 0\,, && \text{if } r = 0\,,\\
&\overline{\mu}_r = d_{r-1} + e_{r-1}\overline{\nu}_{r-1}\,, && \overline{\nu}_r = e_{r-1}f_{r-1}\overline{\mu}_r\,, && \text{if } r \ge 1\,.
\end{aligned}$$

Remark 3. For $r \ge 1$ it is easily seen that $\overline{\mu}_r = V_r(\varphi_{r-1})$ and $\overline{\nu}_r = V_r(\varphi_r)$.

Definition 5 (*Associated Polynomial*). *Let $r \ge 0$, let α and β be nonnegative integers, and let $\mathcal{S}$ be an arbitrary segment of slope $-d_r/e_r$ with left endpoint (α, β). Let $m_0 = 0$ and for $r \ge 1$ and $k \ge 0$ define*

$$\begin{aligned}
m_r &= (1/d_r) \bmod e_r\,,\\
\Omega_r &= \begin{cases} 1 & \text{if } r = 1\,,\\ \Omega_{r-1}^{e_{r-1}f_{r-1}}\,\xi_{r-1}^{m_{r-1}f_{r-1}\overline{\mu}_r} & \text{if } r > 1\,, \end{cases}\\
\Theta(\mathcal{S}, r, k) &= \left\lfloor m_{r-1}\frac{(\beta - kd_r) - (\alpha + ke_r)\,\overline{\nu}_r}{e_{r-1}} \right\rfloor,\\
\Gamma_{\mathcal{S},r,k} &= \Omega_r^{\alpha+ke_r}\,\xi_{r-1}^{\Theta(\mathcal{S},r,k)} \in \mathbf{F}_{q_r}\,.
\end{aligned}$$

Let $K(X) \in \mathbf{Z}_p[X]$ have φ_r-adic expansion

$$K(X) = A_0(X) + A_1(X)\,\varphi_r(X) + \cdots + A_n(X)\,\varphi_r(X)^n$$

with $d_r j + e_r V_r(A_j\,\varphi_r^j) \geq d_r\alpha + e_r\beta$ *for* $j = 0, \ldots, n$ *and let*

$$J = \left\{\, k \;\middle|\; 0 \leq k \leq \lfloor (n-\alpha)/e_r \rfloor,\ \left(\alpha + ke_r,\ V_r(A_{\alpha+ke_r}\,\varphi_r^{\alpha+ke_r})\right) \in \mathcal{S} \,\right\}.$$

We define the level-r associated polynomial of K with respect to $\mathcal{S}$ *to be*

$$\Psi_{\mathcal{S},K}^{(r)}(Y) = \textstyle\sum_{k\in J} \eta_k Y^k$$

with $\eta_k \in \mathbf{F}_{q_r}$ *defined as*

$$\eta_k = \begin{cases} \overline{A}_{\alpha+ke_0} & \text{if } r = 0\,, \\ \overline{B}_k(\xi_0)\,, \quad \text{with } B_k(X) = A_{\alpha+ke_1}(X) \,/\, p^{\beta-kd_1}\,, & \text{if } r = 1\,, \\ \Gamma_{\mathcal{S},r,k}^{-1}\,\Psi_{\mathcal{T}_{r-1,\nu_k},A_{\alpha+ke_r}}^{(r-1)}(\xi_{r-1})\,, \quad \text{with } \nu_k = V_r(A_{\alpha+ke_r})\,, & \text{if } r \geq 2\,. \end{cases}$$

We further define the natural level-r associated polynomial of K *to be*

$$\widetilde{\Psi}_K^{(r)}(Y) = \Psi_{\mathcal{S}_{r,K},K}^{(r)}(Y)\,.$$

Remark 4. The polynomial $\widetilde{\Psi}_K^{(r)}(Y)$ has nonzero constant term.

3 Outline of the Restricted Montes Algorithm

A complete Maple implementation of the restricted Montes algorithm, with proofs and explanatory comments interspersed, is given in [19]. Here we give an outline showing the three major phases of the algorithm. The algorithm begins in phase M_0 (level 0), then alternates between phase M_1 and phase M_2 (level r, for $r = 1, 2, \ldots$) until reaching a terminating condition.

- input: $\Phi(X) \in \mathbf{Z}[X]$ monic and irreducible, $p \in \mathbf{Z}$ prime
- output: $\begin{cases} \text{TRUE} & \text{if } \Phi(X) \text{ is irreducible over } \mathbf{Q}_p[X], \\ \text{FALSE} & \text{if } \Phi(X) \text{ is reducible over } \mathbf{Q}_p[X]. \end{cases}$

$\mathbf{M_0}$:

1. Factorize Φ modulo p:
$$\Phi \equiv \psi_{0,1}^{a_{0,1}} \cdots \psi_{0,\kappa_0}^{a_{0,\kappa_0}} \pmod{p}\,.$$
2. If $\kappa_0 > 1$ then **return** FALSE.
If $\kappa_0 = 1$ and $a_{0,1} = 1$ then **return** TRUE.
3. Define $\varphi_0(X) = X,\ n_0 = 1,\ d_0 = 0,\ e_0 = 1,$
$\psi_0 = \psi_{0,1},\ f_0 = \deg\psi_0,\ \xi_0$ a root of ψ_0.
4. Set $r \leftarrow 1$.

$\mathbf{M_1}$:

5. If $r = 1$ let $\varphi_1(X)$ be a monic polynomial in $\mathbf{Z}[X]$ such that $\overline{\varphi}_1 = \psi_0$.
If $r > 1$ construct H_{r-1} according to Algorithm 1 in Sect. 6 below and let
$$\varphi_r = \varphi_{r-1}^{e_{r-1}f_{r-1}} + H_{r-1}\,.$$

6. Define $n_r = e_{r-1}f_{r-1}n_{r-1} = \deg \varphi_r$.
7. If $r > 1$ and $e_{r-1}f_{r-1} = 1$ then replace $\varphi_{r-1} \leftarrow \varphi_r$ and $r \leftarrow r - 1$.

M₂ : 8. If $\varphi_r = \Phi$ then **return** TRUE.
If $\varphi_r \mid \Phi$ and $\varphi_r \neq \Phi$ then **return** FALSE.

9. Let $\mathcal{S}_{r,1}, \ldots, \mathcal{S}_{r,\lambda_r}$ be the segments of $\mathcal{N}_r(\Phi)$ and let $\zeta_{r,k} + 1$ be the number of points on $\mathcal{S}_{r,k}$ with integer coordinates, for $k = 1, \ldots, \lambda_r$.
10. If $\lambda_r > 1$ then **return** FALSE.
If $\lambda_r = 1$ and $\zeta_{r,1} = 1$ then **return** TRUE.
11. Let $-d_r/e_r$ be the slope of $\mathcal{S}_{r,1}$, with d_r and e_r relatively prime and $e_r > 0$, and construct $\widetilde{\Psi}_{\Phi}^{(r)}(Y) \in \mathbf{F}_{q_r}[Y]$.
12. Factorize
$$\widetilde{\Psi}_{\Phi}^{(r)} = c_r \, \psi_{r,1}^{a_{r,1}} \cdots \psi_{r,\kappa_r}^{a_{r,\kappa_r}}$$
over $\mathbf{F}_{q_r}$, with $c_r \in \mathbf{F}_{q_r}$ a nonzero constant.
13. If $\kappa_r > 1$ then **return** FALSE.
If $\kappa_r = 1$ and $a_{r,1} = 1$ then **return** TRUE.
14. Define $\psi_r = \psi_{r,1}$, $f_r = \deg \psi_r$, ξ_r a root of ψ_r.
15. Replace $r \leftarrow r + 1$.
Go to M_1.

4 Complexity of Fundamental Operations

Notation. We use $\langle \texttt{alpha} \rangle_{\mathbf{F}_p}$ and $\langle \texttt{alpha} \rangle_{\mathbf{Q}}$ to denote the number of operations in $\mathbf{F}_p$ and $\mathbf{Q}$ respectively required for the execution of the procedure `alpha`. We use the notation
$$f(n) \in O(n^{k+\epsilon})$$
as an alternative to the "soft-O" notation
$$f(n) \in O^{\sim}(n^k) \;\equiv\; f(n) \in O(n^k (\ln n)^c)$$
for some positive constant c (see [9]). For $n \geq 3$ and q a prime power we define the following.
$$\mathsf{L}(n) = \ln n \ln\ln n \qquad \mathsf{F}(n,q) = n\,\mathsf{M}(n)\ln(qn)$$
$$\mathsf{M}(n) = n\,\mathsf{L}(n) \qquad \mathsf{K}(q) = \mathsf{M}(\ln q)\ln\ln q$$
We are concerned with the reducibility of the monic polynomial $\Phi(X) \in \mathbf{Z}_p[X]$ for some prime p. We let δ_Φ denote $v_p(\operatorname{disc}\Phi)$ and we let $p^{\delta_\Phi^*}$ denote the p-adic reduced discriminant of Φ [8, Appendix A]. It is clear that $\delta_\Phi^* \leq \delta_\Phi$.

Magnitude of p. To simplify the subsequent discussion we impose the condition that $p \in O(1)$, by which we mean that p is a small prime, not exceeding the magnitude of a single machine word.

Arithmetic in $\mathbf{Z}_p$. If $F(X) \in \mathbf{Z}[X]$ with $F(X) \equiv \Phi(X) \pmod{p^{2\delta_\Phi^*+1}\mathbf{Z}_p[X]}$ then $\Phi(X)$ is reducible in $\mathbf{Z}_p[X]$ if and only if $F(X)$ is reducible in $\mathbf{Z}_p[X]$. Thus in our computations p-adic integers are represented as rational approximations with $2\delta_\Phi^* + 1$ p-adic digits of precision, *i.e.*, as rational integers reduced modulo $p^{2\delta_\Phi^*+1}$.

Schönhage and Strassen have shown that the time required to perform an arithmetic operation on two rational integers of length m is $O(\mathsf{M}(m))$; see [9, Ch.8, §8.3]. It follows that if we represent p-adic integers in this fashion then the cost of an arithmetic operation is $O(\Delta_\Phi)$, with

$$\Delta_\Phi = \mathsf{M}(\delta_\Phi^* \ln p)\,.$$

Arithmetic in $\mathbf{F}_q$. By [9, Ch.14, §14.7], a single operation in $\mathbf{F}_q$ can be performed in $O(\mathsf{K}(q))$ word operations. If $q = p^{f^*}$ the assumption that $\ln p \in O(1)$ gives $\ln q = f^* \ln p \in O(f^*)$ and thus the cost of an operation in $\mathbf{F}_q$ is

$$O(\mathsf{K}(q)) = O\big(\mathsf{M}(\ln q)\ln\ln q\big) \subseteq O\big(f^*(\ln f^*)^2 \ln\ln f^*\big) \subseteq O\big(f^{*\,(1+\epsilon)}\big)\,.$$

For $\alpha \in \mathbf{F}_q$ and any integer n the cost of computing α^n is

$$O(\ln q\,\mathsf{K}(q)) \subseteq O(f^* f^{*\,(1+\epsilon)}) = O(f^{*\,(2+\epsilon)})$$

since we may assume $0 \le n \le q-1$. By [18, Theorem 10], the asymptotic cost for constructing an irreducible polynomial of degree n over the finite field $\mathbf{F}_q$ is

$$O\big((n^2 \ln n + n \ln q)\,\mathsf{L}(n)\big)\,.$$

Polynomial Arithmetic. The number of operations required to evaluate a polynomial of degree n at a given point using Horner's rule is $O(n)$. By [17] and [3], the number of operations needed to multiply two polynomials of degree at most n is $O(\mathsf{M}(n))$. It follows that the number of operations needed to compute the m^{th} power of a polynomial of degree n is

$$O\big(nm \ln^2(nm)\big) \subseteq O\big((nm)^{1+\epsilon}\big)\,.$$

By [9, Ch 14, §14.4 and §14.5], the expected number of operations in $\mathbf{F}_q$ needed to factorize a polynomial of degree n over $\mathbf{F}_q$ is

$$O(\mathsf{F}(n,q)) \subseteq O(n^{2+\epsilon} \ln q)\,.$$

Let $\varphi(X)$ be a monic polynomial in $\mathbf{Z}_p[X]$ of degree n_φ, let $f(X)$ be a polynomial in $\mathbf{Z}_p[X]$ of degree n, and let $k_\varphi = \lfloor n/n_\varphi \rfloor$. Let $E(f, k_\varphi)$ denote the number of operations in $\mathbf{Z}_p$ needed to compute the φ-adic expansion

$$f(X) = \textstyle\sum_{i=1}^{k_\varphi} a_i(X)\,\varphi^i(X)\,.$$

From [9, Ch 5, §5.11], we have

$$E(f, k_\varphi) \in O(k_\varphi(k_\varphi+1)n_\varphi^2) = O(n_\varphi^2 k_\varphi^2) = O(n^2)\,.$$

5 Complexity of the Algorithm

Finite Fields. For $r \geq 0$ the finite field $\mathbf{F}_{q_{r+1}}$ is implemented as $\mathbf{F}_p[\rho_r]$, with

- ρ_r of a root of ψ_r^*,
- $\psi_r^*(Y)$ an arbitrary irreducible monic polynomial in $\mathbf{F}_p[Y]$ of degree f_r^*,
- $f_r^* = f_0 \cdots f_r$.

Thus $\mathbf{F}_{q_{r+1}} = \mathbf{F}_{q_r}[\xi_r] = \mathbf{F}_p[\xi_0, \ldots, \xi_r] = \mathbf{F}_p[\rho_r]$ and $q_{r+1} = q_r^{f_r} = p^{f_r^*}$.

Computing the Newton Polygon. It follows from [19, Theorem 15] that the recursive computation of $V_r(\Phi)$ requires $O(n_\Phi^{2+\epsilon}\Delta_\Phi)$ operations in $\mathbf{Q}$ and that this dominates the cost of constructing $\mathcal{N}_r(\Phi)$.

Computing $\boldsymbol{\varphi_r}$. The construction of $\varphi_r = \varphi_{r-1}^{e_{r-1}f_{r-1}} + H_{r-1}$ is explained in Sect. 6 below. The cost of computing $\varphi_{r-1}^{e_{r-1}f_{r-1}}$ is

$$\begin{aligned}
&\big\langle \varphi_{r-1}^{e_{r-1}f_{r-1}} \big\rangle_{\mathbf{F}_p} = 0\,,\\
&\big\langle \varphi_{r-1}^{e_{r-1}f_{r-1}} \big\rangle_{\mathbf{Q}} \in O\big((n_{r-1}e_{r-1}f_{r-1})^{1+\epsilon}\Delta_\Phi\big) = O\big(n_r^{1+\epsilon}\Delta_\Phi\big)\,.
\end{aligned}$$

A slight modification of the proof of [19, Theorem 17] shows that the cost of constructing $H_{r-1} = H_{r-1,\overline{\nu}_r,\gamma_{r-1}}$ is

$$\begin{aligned}
&\big\langle H_{r-1} \big\rangle_{\mathbf{F}_p} \in O\big(r f_{r-1} f_{r-2}^{*\,(3+\epsilon)}\big) \subseteq O(r n_r^{3+\epsilon})\,,\\
&\big\langle H_{r-1} \big\rangle_{\mathbf{Q}} \in O\big(r n_r^{1+\epsilon}\Delta_\Phi\big)\,.
\end{aligned}$$

Thus the cost of computing φ_r is dominated by the cost of computing H_{r-1}.

Computing the Associated Polynomial. It follows from [19, Theorem 16] that if $r \geq 2$ then

$$\begin{aligned}
&\big\langle \widetilde{\Psi}_\Phi^{(r)} \big\rangle_{\mathbf{F}_p} \in O(n_\Phi n_r^{1+\epsilon}) \subseteq O(n_\Phi^{2+\epsilon})\,,\\
&\big\langle \widetilde{\Psi}_\Phi^{(r)} \big\rangle_{\mathbf{Q}} \in O\big(n_\Phi n_r^{1+\epsilon}\Delta_\Phi\big) \subseteq O\big(n_\Phi^{2+\epsilon}\Delta_\Phi\big)\,.
\end{aligned}$$

Total Complexity. The cost of phase M_0 is dominated by the cost of factorizing Φ over $\mathbf{F}_p$. Hence

$$\begin{aligned}
&\big\langle \mathrm{M}_0 \big\rangle_{\mathbf{F}_p} \in O(\mathsf{F}(n_\Phi, p)) \subseteq O(n_\Phi^{2+\epsilon})\,,\\
&\big\langle \mathrm{M}_0 \big\rangle_{\mathbf{Q}} \in O(1)\,.
\end{aligned}$$

The cost of phase M_1 is dominated by the cost of constructing φ_r. Hence

$$\begin{aligned}\big\langle \mathrm{M}_1(r)\big\rangle_{\mathbf{F}_p} &\in O(r n_r^{3+\epsilon})\,,\\ \big\langle \mathrm{M}_1(r)\big\rangle_{\mathbf{Q}} &\in O\big(r n_r^{1+\epsilon}\Delta_\Phi\big)\,.\end{aligned}$$

The cost in $\mathbf{Q}$-operations of phase M_2 is dominated by the construction of the Newton polygon $\mathcal{N}_r(\Phi)$ and of the associated polynomial $\widetilde{\Psi}_\Phi^{(r)}$, each of which require $O(n_\Phi^{2+\epsilon}\Delta_\Phi)$ operations in $\mathbf{Q}$. Since $\mathbf{F}_{q_{r+1}} = \mathbf{F}_p[\rho_r]$, the necessity of expressing ξ_r and ρ_{r-1} in terms of ρ_r arises. This is achieved in each case by factoring ψ_{r-1}^* over $\mathbf{F}_p[\rho_r]$, which requires $O(f_r^{*3+\epsilon}) \subseteq O(n_\Phi^{3+\epsilon})$ operations in $\mathbf{F}_p$. These are the dominant finite-field operations in M_2, hence

$$\begin{aligned}\big\langle \mathrm{M}_2(r)\big\rangle_{\mathbf{F}_p} &\in O(n_\Phi^{3+\epsilon})\,,\\ \big\langle \mathrm{M}_2(r)\big\rangle_{\mathbf{Q}} &\in O(n_\Phi^{2+\epsilon}\Delta_\Phi)\,.\end{aligned}$$

We now estimate the number of operations required for the chain of computations

$$\mathrm{M}_0(\Phi) \to \mathrm{M}_1(1) \to \mathrm{M}_2(1) \to \mathrm{M}_1(2) \to \mathrm{M}_2(2) \to \cdots \to \mathrm{M}_1(m) \to \mathrm{M}_2(m)$$

with the algorithm terminating at level m. We note that at level r we have $n_0 < n_1 < \cdots < n_r$ with $n_0 \mid n_1 \mid \cdots \mid n_r$. Hence $2^r \le n_r$ and thus $r \in O(\ln n_r)$. It follows that $m \in O(\ln n_\Phi)$ and we have

$$\begin{aligned}&\big\langle \mathrm{M}_0(F)\big\rangle_{\mathbf{F}_p} + \textstyle\sum_{r=1}^m \big(\big\langle \mathrm{M}_1(r)\big\rangle_{\mathbf{F}_p} + \big\langle \mathrm{M}_2(r)\big\rangle_{\mathbf{F}_p}\big)\\ &\quad= \big\langle \mathrm{M}_0(F)\big\rangle_{\mathbf{F}_p} + \textstyle\sum_{r=1}^m \big\langle \mathrm{M}_1(r)\big\rangle_{\mathbf{F}_p} + \sum_{r=1}^m \big\langle \mathrm{M}_2(r)\big\rangle_{\mathbf{F}_p}\\ &\quad\in O\big(n_\Phi^{2+\epsilon} + m^2 n_\Phi^{3+\epsilon} + m n_\Phi^{3+\epsilon}\big)\\ &\quad\subseteq O\big(n_\Phi^{3+\epsilon}\big)\,,\\ &\big\langle \mathrm{M}_0(F)\big\rangle_{\mathbf{Q}} + \textstyle\sum_{r=1}^m \big(\big\langle \mathrm{M}_1(r)\big\rangle_{\mathbf{Q}} + \big\langle \mathrm{M}_2(r)\big\rangle_{\mathbf{Q}}\big)\\ &\quad= \big\langle \mathrm{M}_0(F)\big\rangle_{\mathbf{Q}} + \textstyle\sum_{r=1}^m \big\langle \mathrm{M}_1(r)\big\rangle_{\mathbf{Q}} + \sum_{r=1}^m \big\langle \mathrm{M}_2(r)\big\rangle_{\mathbf{Q}}\\ &\quad\in O\big(n_\Phi + m^2 n_\Phi^{1+\epsilon}\Delta_\Phi + m n_\Phi^{2+\epsilon}\Delta_\Phi\big)\\ &\quad\subseteq O\big(n_\Phi^{2+\epsilon}\Delta_\Phi\big)\,.\end{aligned}$$

From [16, Proposition 4.1] it follows that the case $e_{r-1}f_{r-1} = 1$ can occur at most

$$2\,\frac{e_{r-2}^*}{n_\Phi}\,v_p(\operatorname{disc}\Phi) \;\le\; 2\,v_p(\operatorname{disc}\Phi)$$

times. Hence the sequence

$$\mathrm{M}_1(r) \to \mathrm{M}_2(r-1) \to \mathrm{M}_1(r)$$

can occur at most $2v_p(\operatorname{disc} \Phi)$ times in the course of the computation. From the results above we have

$$\big\langle \mathrm{M}_1(r) \big\rangle_{\mathbf{F}_p} + \big\langle \mathrm{M}_2(r-1) \big\rangle_{\mathbf{F}_p} \in O(r n_r^{3+\epsilon} + n_\Phi^{3+\epsilon}) \subseteq O(n_\Phi^{3+\epsilon}) \,,$$
$$\big\langle \mathrm{M}_1(r) \big\rangle_{\mathbf{Q}} + \big\langle \mathrm{M}_2(r-1) \big\rangle_{\mathbf{Q}} \in O(r n_r^{1+\epsilon} + n_\Phi^{2+\epsilon} \Delta_\Phi) \subseteq O(n_\Phi^{2+\epsilon} \Delta_\Phi) \,.$$

Since $\delta_\Phi^* \le \delta_\Phi$ and $\ln p \in O(1)$ we have

$$\Delta_\Phi = \mathsf{M}(\delta_\Phi^* \ln p) \in O(\delta_\Phi^{1+\epsilon}) \,.$$

It now follows that the expected number of operations required for the restricted Montes algorithm to terminate is

$$O\big(2\delta_\Phi (n_\Phi^{3+\epsilon} + n_\Phi^{2+\epsilon} \Delta_\Phi)\big) \;\subseteq\; O\big(n_\Phi^{3+\epsilon}\, \delta_\Phi + n_\Phi^{2+\epsilon}\, \delta_\Phi^{2+\epsilon}\big) \,.$$

Remark 5. This is a slight improvement on the estimate $O(n_\Phi^{3+\epsilon} \delta_\Phi^{2+\epsilon})$ from [19]. By way of comparison, Pauli [16] gives an estimate of

$$O\big(n_\Phi^{3+\epsilon}\, \delta_\Phi^{1+\epsilon} + n_\Phi^{2+\epsilon}\, \delta_\Phi^{2+\epsilon}\big)$$

bit operations for factorization of a univariate polynomial over $\mathbf{Q}_p$ via the "two-element" method.

6 The Construction of φ_r

Algorithm 1 (Montes). *Given d_s, e_s, f_s, etc., for $1 \le s \le r$ and given*

- *an integer t in the range $1 \le t \le r$,*
- *an integer $\nu \ge \overline{\nu}_{t+1}$,*
- *a nonzero polynomial $\delta(Y) \in \mathbf{F}_{q_t}[Y]$ of degree less than f_t,*

to construct a polynomial $H_{t,\nu,\delta}(X) \in \mathbf{Z}_p[X]$ such that

- $\deg H_{t,\nu,\delta} < n_{t+1}$,
- $V_{t+1}(H_{t,\nu,\delta}) = \nu$,
- $\Psi^{(t)}_{\mathcal{T}_{t,\nu},\, H_{t,\nu,\delta}}(Y) = \delta(Y)$.

Construction. Let $\zeta_0, \ldots, \zeta_{f_t-1}$ in $\mathbf{F}_{q_t}$ be such that

$$\delta(Y) = \textstyle\sum_{i=0}^{f_t-1} \zeta_i\, Y^i \,.$$

Since $\delta(Y) \ne 0$ the set $J_\delta = \{\, i \mid 0 \le i \le f_t - 1,\, \zeta_i \ne 0 \,\}$ is not empty. For $i \in J_\delta$ we construct $K_i(X)$ as follows.

- We take $\delta_i(Y)$ to be the unique polynomial in $\mathbf{F}_{q_{t-1}}[Y]$ of degree less than f_{t-1} such that $\delta_i(\xi_{t-1}) = \Gamma_{\mathcal{T}_{t,\nu},t,i}\, \zeta_i$.

- If $t = 1$ we take $P_i(X)$ to be a polynomial in $\mathbf{Z}_p[X]$ of degree less than f_0 such that $\overline{P}_i(Y) = \delta_i(Y)$ and we set
$$K_i(X) = p^{\beta_{1,\nu} - id_1} P_i(X) .$$
- If $t \geq 2$ we let $\nu_i = (\beta_{t,\nu} - id_t) - (\alpha_{t,\nu} + ie_t)\overline{\nu}_t$ and we set
$$K_i(X) = H_{t-1,\,\nu_i,\,\delta_i}(X) .$$

Having constructed $K_i(X)$ for $i \in J_\delta$, we set
$$H_{t,\nu,\delta}(X) = \textstyle\sum_{i \in J_\delta} K_i(X)\, \varphi_t(X)^{\alpha_{t,\nu} + ie_t} . \qquad \square$$

Remark 6. It follows from [13, Proposition 3.2] that Algorithm 1 correctly constructs the polynomial $H_{t,\nu,\delta}$ with the indicated properties.

The construction of $\delta_i(Y)$ in Algorithm 1 being rather complicated, we provide some implementation details.

Computing $\boldsymbol{\Upsilon_r}$. If $r > 0$ we construct $\Upsilon_r \in \mathbf{F}_p^{f_r^* \times f_r \times f_{r-1}^*}$ such that
$$\rho_{r-1}^k\, \xi_r^j \;=\; \textstyle\sum_{h=0}^{f_r^*-1} (\Upsilon_r)_{h,j,k}\; \rho_r^h$$
for $j = 0, \ldots, f_r - 1$, $k = 0, \ldots, f_{r-1}^* - 1$. In practice we construct $\widetilde{\Upsilon}_r \in \mathbf{F}_p^{f_r^* \times f_r^*}$ and $\widetilde{M} \in \mathbf{F}_p^{f_r^*}$ such that
$$(\widetilde{\Upsilon}_r)_{1+h,1+j+kf_r} = (\Upsilon_r)_{h,j,k} , \quad \widetilde{M}_{1+j+kf_r} = M_{j,k} ,$$
for $h = 0, \ldots, f_r^* - 1$, $j = 0, \ldots, f_r - 1$, $k = 0, \ldots, f_{r-1}^* - 1$.

Deriving $\boldsymbol{\delta_i}$ from $\boldsymbol{\Upsilon_{t-1}}$. Given $i \in J_\delta$ and $t \geq 2$, let
$$\mathit{\Gamma}_{\mathcal{T}_{t,\nu},t,i}\, \zeta_i \;=\; \kappa_{i,0} + \kappa_{i,1}\, \rho_{t-1} + \cdots + \kappa_{i,f_{t-1}^*-1}\, \rho_{t-1}^{f_{t-1}^*-1} \;\in\; \mathbf{F}_p[\rho_{t-1}] \;=\; \mathbf{F}_{q_t} .$$
For $j = 0, \ldots, f_{t-1} - 1$, $k = 0, \ldots, f_{t-2}^* - 1$, let $M_{j,k} \in \mathbf{F}_p$ satisfy
$$\textstyle\sum_{j=0}^{f_{t-1}-1} \sum_{k=0}^{f_{t-2}^*-1} (\Upsilon_{t-1})_{h,j,k}\; M_{j,k} \;=\; \kappa_{i,h}$$
for $h = 0, \ldots, f_{t-1}^* - 1$, and let
$$\delta_i(Y) \;=\; \textstyle\sum_{j=0}^{f_{t-1}-1} \big(\sum_{k=0}^{f_{t-2}^*-1} M_{j,k}\; \rho_{t-2}^k \big)\, Y^j .$$
Then $\delta_i(Y) \in \mathbf{F}_p[\rho_{t-2}][Y] = \mathbf{F}_{q_{t-1}}[Y]$ and
$$\begin{aligned}
\delta_i(\xi_{t-1}) &= \textstyle\sum_{j=0}^{f_{t-1}-1} \sum_{k=0}^{f_{t-2}^*-1} M_{j,k}\; \rho_{t-2}^k\; \xi_{t-1}^j \\
&= \textstyle\sum_{j=0}^{f_{t-1}-1} \sum_{k=0}^{f_{t-2}^*-1} M_{j,k} \sum_{h=0}^{f_{t-1}^*-1} (\Upsilon_{t-1})_{h,j,k}\; \rho_{t-1}^h \\
&= \textstyle\sum_{h=0}^{f_{t-1}^*-1} \sum_{j=0}^{f_{t-1}-1} \sum_{k=0}^{f_{t-2}^*-1} (\Upsilon_{t-1})_{h,j,k}\; M_{j,k}\; \rho_{t-1}^h \\
&= \textstyle\sum_{h=0}^{f_{t-1}^*-1} \kappa_{i,h}\; \rho_{t-1}^h \\
&= \mathit{\Gamma}_{\mathcal{T}_{t,\nu},t,i}\, \zeta_i .
\end{aligned}$$

The essential properties of φ_r are as follows (see [19, Proposition 9]).

Proposition 1 (Montes). *Let d_s, e_s, f_s, φ_s, ψ_s, etc., be given for $1 \le s \le r-1$ and let*

$$\gamma_{r-1}(Y) = \Omega_{r-1}^{-e_{r-1}f_{r-1}}(\psi_{r-1}(Y) - Y^{f_{r-1}}),$$
$$\varphi_r(X) = \varphi_{r-1}(X)^{e_{r-1}f_{r-1}} + H_{r-1,\overline{\nu}_r,\gamma_{r-1}}(X)\,.$$

Then $\varphi_r(X)$ is a monic polynomial in $\mathbf{Z}_p[X]$ with the following properties.

- $\deg \varphi_r = n_r$.
- $\mathcal{N}_{r-1}(\varphi_r)$ *consists of the single segment* $\mathcal{S}_{r-1,\varphi_r}$.
- $V_r(\varphi_r) = \overline{\nu}_r$.
- $\widetilde{\Psi}_{\varphi_r}^{(r-1)}(Y) = \Omega_{r-1}^{-e_{r-1}f_{r-1}}\psi_{r-1}(Y)$.
- φ_r *is irreducible over* $\mathbf{Z}_p$.

7 Supplementary Remarks

The MAPLE code from [19], including an example, can be found at this URL.

`http://www.mathstat.concordia.ca/faculty/ford/Student/Veres/mmtest.mpl`

Two recent monographs by Guàrdia, Montes, and Nart give a thorough revision of the theory underlying the Montes algorithm [10] and a detailed description of the algorithm [11]. Algorithm 1 and Proposition 1 in Sect. 6 above appear in [10]. A simpler choice for Ω_r (see Definition 5) is also given, but with no effect on the complexity of the algorithm.

References

1. Berlekamp, E.R.: Factoring Polynomials over Finite Fields. Bell Systems Technical Journal 46, 1853–1859 (1967)
2. Berlekamp, E.R.: Factoring Polynomials over Large Finite Fields. Math. Comp. 24, 713–735 (1970)
3. Cantor, D.G., Kaltofen, E.: On Fast Multiplication of Polynomials over Arbitrary Algebras. Acta Informatica 28(7), 693–701 (1991)
4. Cantor, D.G., Zassenhaus, H.: A New Algorithm for Factoring Polynomials Over Finite Fields. Math. Comp. 36, 587–592 (1981)
5. Dedekind, R.: Supplement X to Vorlesungen über Zahlentheorie von P.G. Lejeune Dirichlet (2nd ed.). Vieweg, Braunschweig (1871); Also Werke 3, 223–261 (1932) (in part)
6. Dedekind, R.: Sur la théorie des nombres entiers algébriques. Gauthier-Villars (1877); Also Bull. des Sci. Math. Astron. 11(1), 278–288 (1876); 1(2), 17–41, 69–92, 144–164, 207–248 (1877) and Werke 3, 263–296 (1932) (in part)
7. Dedekind, R.: Über den Zusammenhang zwischen der Theorie der Ideale und der Theorie der höheren Kongruenzen. Abhandlungen der Königlichen Gesellschaft der Wissenschaften zu Göttingen 23, 1–23 (1878)

8. Ford, D., Pauli, S., Roblot, X.-F.: A Fast Algorithm for Polynomial Factorization over $\mathbf{Q}_p$. Journal de Théorie des Nombres de Bordeaux 14, 151–169 (2002)
9. von zur Gathen, J., Gerhard, J.: Modern computer algebra. Cambridge University Press, Cambridge (1999)
10. Guàrdia, J., Montes, J., Nart, E.: Newton polygons of higher order in algebraic number theory (2008), arXiv:0807.2620v2[math.NT]
11. Guàrdia, J., Montes, J., Nart, E.: Higher Newton polygons in the computation of discriminants and prime ideal decomposition in number fields (2008), arXiv:0807.4065v3[math.NT]
12. Hensel, K.: Theorie der algebraischen Zahlen. Teubner, Leipzig (1908)
13. Montes, J.: Polígonos de Newton de orden superior y aplicaciones aritméticas. PhD thesis, Universitat de Barcelona (1999)
14. Montes, J., Nart, E.: On a theorem of Ore. Journal of Algebra 146, 318–334 (1992)
15. Ore, Ø.: Newtonsche Polygone in der Theorie der algebraischen Körper. Math. Ann. 99 (1928)
16. Pauli, S.: Factoring Polynomials over Local Fields. Journal of Symbolic Computation 32(5), 533–547 (2001)
17. Schönhage, A., Strassen, V.: Schnelle Multiplikation großer Zahlen. Computing 7, 281–292 (1971)
18. Shoup, V.: Fast Construction of Irreducible Polynomials over Finite Fields. Journal of Symbolic Computation 17, 371–394 (1994)
19. Veres, O.: On the Complexity of Polynomial Factorization over p-adic Fields. PhD Dissertation, Concordia University (2009), `http://www.mathstat.concordia.ca/faculty/ford/Student/Veres/vthp.pdf`
20. Zassenhaus, H.: On Hensel factorization II. In: Symposia Mathematica XV, Instituto Di Alta Matematica, pp. 499–513. Academic Press, New York (1975)

Congruent Number Theta Coefficients to 10^{12}

William B. Hart[1,⋆], Gonzalo Tornaría[2], and Mark Watkins[3,⋆⋆]

[1] Mathematics Institute, Warwick University, Coventry, United Kingdom
[2] Centro de Matemática, Universidad de la República, Montevideo, Uruguay
[3] Department of Mathematics and Statistics, University of Sydney, Australia

Abstract. We report on a computation of congruent numbers, which subject to the Birch and Swinnerton-Dyer conjecture is an accurate list up to 10^{12}. The computation involves multiplying long theta series as per Tunnell (1983). The method, which we describe in some detail, uses a multimodular disk based technique for multiplying polynomials out-of-core which minimises expensive disk access by keeping data truncated.

1 History

The congruent number problem first makes its appearance in the literature of the classical Islamic period, e.g. in al-Karaji's text the al-Fakhri. Dickson [11] states that an anonymous Arab manuscript written before 972 A.D. contains reference to the problem.

The problem was initially studied in terms of squares of rational numbers: a natural number n is *congruent* iff there exist rational numbers x, y, z, w such that

$$x^2 + ny^2 = z^2 \quad \text{and} \quad x^2 - ny^2 = w^2.$$

In other words n is congruent iff there exist three rational squares in arithmetic progression with common difference n. It suffices to consider squarefree n.

Bachet, in translating Diophantus' Arithmetica, wrote an appendix of problems on right triangles. Problem 20 was "to find a right-angled triangle such that its area is equal to a given number". This equivalent problem refers to right triangles with rational sides whose area n is a natural number.

The problem was studied by Fermat and Fibonacci the latter of which referred to a common difference of squares in arithmetic progression as a *congruum*. Euler referred to such numbers as *congruere* meaning to "come together".

Many authors have contributed to the study of the properties of and computation of congruent numbers, including Alter, Curtz and Kubota [1] who conjectured that if n is congruent to 5, 6 or 7 modulo 8 then n is a congruent number. This was shown to be true, subject to the weak Birch and Swinnerton-Dyer conjecture by Stephens [35] in 1975.

⋆ Supported by EPSRC grant number EP/G004870/1.
⋆⋆ All authors were supported at workshops administered by AIMath under NSF Grant number DMS-0757627.

G. Hanrot, F. Morain, and E. Thomé (Eds.): ANTS-IX 2010, LNCS 6197, pp. 186–200, 2010.

The earliest computations of congruent numbers are due to the classical Islamic mathematicians, the congruent numbers 5, 6, 14, 15, 21, 30, 34, 65, 70, 110, 154, 190, 210, 221, 231, 246, 290, 390, 429, 546 and ten other substantially larger congruent numbers being known to them. Fibonacci, Genocchi and Gérardin added 7, 22, 41, 69, 77 and forty-three other values below 1000.

Fermat showed that 1 is not congruent in 1659, something which had been stated but not proved by Fibonacci in 1225. By scaling this is equivalent to the fact that no square number can be congruent.

Bastien [5] observed that numbers which are prime and 3 modulo 8, products of two such primes, twice a prime which is 5 modulo 8, twice a product of two such primes or twice a prime which is 9 modulo 16 are not congruent.

Numerous congruent numbers were demonstrated by Alter, Curtz and Kubota [1] and by Jean Lagrange in his thesis [23]. See Guy [17] for further details on the history of the computation of congruent numbers.

More recently Monsky [28] showed that, for example, two times the product of primes $p \equiv 1 \pmod 8$ and $q \equiv 7 \pmod 8$ with $(p/q) = -1$ is a congruent number. For a history of results along these lines see Feng [13]. Also see [27].

By 1980 there were numerous values below 1000 not yet decided either way. By 1986 Kramarz [26] had handled all cases up to 2000, and Noe's list up to 10000 is included in Sloane's database. Matsuno had reached 300000 in 2005.

Subject to a conjecture of Birch and Swinnerton-Dyer (see Tunnell's Criterion below), Rogers [32] had computed all congruent numbers up to 10^7 by the year 2000 and Mike Rubinstein (personal communication) had computed all congruent numbers up to 10^9 a few years prior to the current work. We had raised that limit to 2×10^{10} by 2008 and with this paper the current plateau is now 10^{12}.

By counting representations of n or $n/2$ by ternary quadratic forms, previous computations had the asymptotic running time $O(N^{\frac{3}{2}})$ for computing coefficients up to a limit N. In this paper we describe a multimodular Fast Fourier Transform technique with quasilinear runtime. We demonstrate that the method is practical as it permits computations whose data is considerably larger than main memory.

2 Relating Congruent Numbers to Elliptic Curves

If three rational squares in arithmetic progression have common difference n, their product is a square:

$$v^2 = (u^2 - n)u^2(u^2 + n) = (u^2)^3 - n^2(u^2).$$

This shows immediately that if n is congruent then it corresponds to a point (u^2, v) on the elliptic curve $E_n : y^2 = x^3 - n^2x$.

Along similar lines, in 1877 Lucas showed that n is congruent iff $y^2 = x^4 - n^2$ has a positive rational solution.

The group of points on the curve E_n is isomorphic to $(\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}) \times \mathbb{Z}^r$ where r is the rank. The three non-trivial 2-torsion points do not yield congruent numbers and so n is congruent iff E_n has positive rank.

There has been considerable interest in verifying that the curves E_n for which n is thought to be congruent do in fact have positive rank. See for example the tables of Elkies [12].

As the sign of the functional equation of $L(E_n/\mathbb{Q}, s)$ is $+1$ for $n \equiv 1, 2, 3 \pmod 8$ and -1 for $n \equiv 5, 6, 7 \pmod 8$ [7] then by the Parity Conjecture (a special case of the Birch and Swinnerton-Dyer Conjecture) we expect that the rank of E_n is even in the $+1$ case and odd in the -1 case. This is an interesting test of the Birch and Swinnerton-Dyer Conjecture.

2.1 Tunnell's Criterion

In 1983 Jerrold Tunnell gave the following criterion:

Theorem 1 (Tunnell). *Let n be an odd squarefree positive integer. Set*

$$a(n) = \#\{(x,y,z) \in \mathbb{Z}^3 \mid x^2 + 2y^2 + 8z^2 = n\} \\ - 2\,\#\{(x,y,z) \in \mathbb{Z}^3 \mid x^2 + 2y^2 + 32z^2 = n\},$$

$$b(n) = \#\{(x,y,z) \in \mathbb{Z}^3 \mid x^2 + 4y^2 + 8z^2 = n\} \\ - 2\,\#\{(x,y,z) \in \mathbb{Z}^3 \mid x^2 + 4y^2 + 32z^2 = n\}.$$

If n is congruent then $a(n) = 0$. If $2n$ is congruent then $b(n) = 0$. Moreover, if the weak BSD conjecture is true for the curve $y^2 = x^3 - n^2x$ then the converses also hold: $a(n) = 0$ implies n is congruent and $b(n) = 0$ implies $2n$ is congruent.

We explain briefly the connection between the curves E_n and Tunnell's criterion.

The curve E_n is a quadratic twist of the curve $E : y^2 = x^3 - x$. Associated to E is a weight 2 newform $F(z) = \eta(4z)^2\eta(8z)^2 \in S_2^{\text{new}}(\Gamma_0(32))$ such that $L(E,s) = L(F,s)$, where $L(E,s)$ is the Hasse-Weil L-series of the elliptic curve E and $L(F,s)$ is the Mellin transform of the modular form F.

If we write $L(E,s) = \sum b_m m^{-s}$ then $L(E_n,s) = L_F(\chi_D,s) = \sum \chi_D(m) b_m m^{-s}$, where $D = n$ if $n \equiv 1 \pmod 4$ and $D = 4n$ if $n \equiv 2, 3 \pmod 4$.

The importance of this fact is that the conjecture of Birch and Swinnerton-Dyer (applied to E_n) then gives a condition on when n can be congruent:

Conjecture 1 (Birch and Swinnerton-Dyer). If E is an elliptic curve defined over $\mathbb{Q}$ then $L(E,1) = 0$ iff E has positive rank.

The following theorem of Shimura gives a link between modular forms of half integer weight $k/2$ and forms of integer weight $k-1$. The correspondence is called a *Shimura lift*. We are interested in this theorem in the case $k = 3$.

Theorem 2 (Shimura). *Let $f(z) = \sum_{m=1}^{\infty} a(m) q^m \in S_{k/2}(4N, \chi)$ be a modular form of weight $k/2$ for $\Gamma_0(4N)$ (actually $\Delta_0(4N)$) with χ a Dirichlet character modulo $4N$ and suppose that $T_p^2(f) = \omega_p f$ for all primes p, where T_p^2 are the*

Hecke operators. Define $F(z) = \sum_{m=1}^{\infty} A(m)q^m$ *where the values* $A(m)$ *are given by*

$$\sum_{m=1}^{\infty} A(m)m^{-s} = \prod_{p}(1 - \omega^p p^s + \chi(p)^2 p^{k-2-2s})^{-1}.$$

Then for some integer N_0 *divisible by the conductor of* χ^2 *we have that* $F(z) \in M_{k-1}(N_0, \chi^2)$*, i.e.* $F(z)$ *is an integer weight modular form of weight* $k-1$*.*

As mentioned above, we are interested in whether or not the L-series $L(E_n, s)$ vanishes at $s = 1$.

Tunnell made use of a result of Waldspurger to access information about the value of these L-series at $s = 1$. The basic idea behind Waldspurger's Theorem and related results is that if $F(z)$ is the Shimura lift of $f(z)$ as per the previous theorem, then the value of $L(F_n, s)$ at $s = (k-1)/2$ for squarefree n, is proportional to the n-th Fourier coefficient of $f(z)$. In particular if suitable forms $f(z)$ can be identified then it is possible to determine when $L(F_n, s)$ vanishes at the centre of the critical strip, $s = (k-1)/2$.

The following result (which is a reformulation of the theorem of Waldspurger, see [30]) formulates this more precisely.

Theorem 3 (Waldspurger). *If* $F(z) = \sum_{m=1}^{\infty} a(m)q^m \in S_{k-1}^{new}(\Gamma_0(M))$ *and* $\delta = \pm 1$ *is the sign of the functional equation of* $L(F, s)$ *then there is a Dirichlet character* χ *modulo* $4N$*, a positive integer* $M|N$*, a nonzero complex number* Ω_F *and a nonzero Hecke eigenform*

$$f(z) = \sum_{m=1}^{\infty} b_F(m)q^m \in S_{k/2}(\Gamma_0(4N), \chi)$$

such that there are fundamental disciminants n*, coprime to* $4N$ *and with the same sign as* δ *that lie in arithmetic progressions and for which*

$$b_F(n_0)^2 = \varepsilon_n \cdot \frac{L(F_n, (k-1)/2)n_0^{k/2}}{\Omega_F},$$

where ε_n *is algebraic and* $n_0 = |n|$ *if* n *is odd, otherwise* $n_0 = |n|/4$*. For all other* n *with the same sign as* δ *the Fourier coefficients* $b_F(n_0)$ *vanish.*

By careful examination of the conditions of Waldspurger's Theorem, Tunnell was able to construct modular forms which allowed for identification of the values of n for which $L(E_n, s)$ vanishes at $s = 1$. Even better yet, he was able to write these weight $3/2$ modular forms as the product of explicit theta series.

Following Tunnel we let $g = (\theta_1 - \theta_4)(\theta_8 - 2\theta_{32})$, where $\theta_t = \sum_{m=-\infty}^{\infty} q^{tm^2}$. Then

$$g\,\theta_2 = \sum_{m=1}^{\infty} a(m)q^m \in S_{\frac{3}{2}}(\Gamma_0(128)),$$

$$g\,\theta_4 = \sum_{m=1}^{\infty} b(m)q^m \in S_{\frac{3}{2}}(\Gamma_0(128), \chi),$$

where $\chi(r) = \left(\frac{8}{r}\right)$. Note this agrees with the formulas for $a(n)$ and $b(n)$ given above for odd n. Tunnell proved that these were Hecke eigenforms whose Shimura lift was $F(z)$. He then showed that if n is an odd positive squarefree integer then

$$L(E_n, 1) = a(n)^2 \cdot \frac{\Omega}{4\sqrt{n}}, \quad \text{and } L(E_{2n}, 1) = b(n)^2 \cdot \frac{\Omega}{2\sqrt{2n}},$$

for a certain real period Ω.

For further information on Tunnell's approach, see Tunnell's original paper [39] and the books by Ono [30] and Koblitz [25].

The above result of Tunnell allows us to determine congruent numbers, subject to the BSD conjecture, simply by checking whether the Fourier coefficients $a(n)$ and $b(n)$ are zero.

Thus the entire problem of determining congruent numbers is reduced to computing the theta series g and θ_t and performing power series multiplications. We actually use slight modifications of these θ-functions, which allow us to exploit additional information on arithmetic progressions.

2.2 Our Θ-Functions

Rather than use the modular forms of Tunnell given above, we note (as suggested to us by N. D. Elkies) that we can split the problem(s) up by a factor of two. The series $g\,\theta_2$ and $g\,\theta_4$ can each be split into a sum of two similar products, each of which is supported on (approximately) half as many coefficients.

Indeed, we have the following product expressions:

$$\begin{aligned}
\theta_8(\theta_1 - \theta_4) \times (\theta_8 - 2\theta_{32}) &= \sum_{n \equiv 1 \pmod 8} a(n)\, q^n, \\
(\theta_2 - \theta_8)(\theta_1 - \theta_4) \times (\theta_8 - 2\theta_{32}) &= \sum_{n \equiv 3 \pmod 8} a(n)\, q^n, \\
\theta_{16}(\theta_1 - \theta_4) \times (\theta_8 - 2\theta_{32}) &= \sum_{n \equiv 1 \pmod 8} b(n)\, q^n, \\
(\theta_4 - \theta_{16})(\theta_1 - \theta_4) \times (\theta_8 - 2\theta_{32}) &= \sum_{n \equiv 5 \pmod 8} b(n)\, q^n.
\end{aligned}$$

As each factor above is a (shifted) power series in q^8, our complexity reduces by a factor of 8. Indeed, the second factor above is $\theta_8 - 2\theta_{32} = C(q^8)$ where $C = \theta_1 - 2\theta_4$ is a sparse power series which can be quickly computed. For the first factor, we can easily compute theta series A_1, A_3, B_1 and B_5 such that

$$\begin{aligned}
\theta_8(\theta_1 - \theta_4) &= q\, A_1(q^8), & (\theta_2 - \theta_8)(\theta_1 - \theta_4) &= q^3\, A_3(q^8), \\
\theta_{16}(\theta_1 - \theta_4) &= q\, B_1(q^8), & (\theta_4 - \theta_{16})(\theta_1 - \theta_4) &= q^5\, B_5(q^8).
\end{aligned}$$

These series can be computed directly by counting lattice points in 2 dimensions, taking approximately linear time. So we only need one convolution for each of

the four cases: two convolutions of 1.25×10^{11} coefficients (for the $a(n)$,) and two convolutions of 6.25×10^{10} coefficients (for the $b(n)$.)

The computation of the Θ-series can be done efficiently in intervals, taking essentially $\sqrt{N}$ time to compute the coefficients between N and $N + \sqrt{N}$. For N up to 1.25×10^{11} this ensures each interval includes less than $500,000$ coefficients, fitting comfortably in a typical L2 cache. This cache locality is essential for the computation.

3 "Out-of-Core" Fast Fourier Transform Methods

The complex FFT algorithm was essentially known to Gauss in 1805 (see [19]) but developed in its current form by Cooley and Tukey in 1965 [9].

In 1971 Schönhage and Strassen presented two algorithms for multiplication of large integers based on the FFT [33]. One of these methods, where the field of complex numbers is replaced by a finite ring $\mathbb{Z}/p\mathbb{Z}$ containing a principal root of unity of order 2^K, has become known as the Schönhage-Strassen method. It can multiply two n bit numbers in asymptotic time $O(n \log n \log \log n)$.

Power series multiplication can be effected by truncating a full polynomial multiplication of two n term polynomials to length n and by encoding the polynomial multiplication as an integer multiplication using Kronecker Segmentation. The latter technique is that of evaluating the polynomials at a power of 2 chosen sufficiently large that the product coefficients can be identified from their binary representation in the output of the large integer multiplication.

In the literature, FFT computations whose data exceeds the size of available memory are referred to as *out-of-core* FFT methods.

The literature is replete with many references to methods for defunct vector architectures, or for distributed memory systems, including those with tree, mesh or hypercube architectures (see [2], [8], [24], [36] and [38] for examples), where the emphasis is often on minimising interprocess communication.

In our case, we used a shared memory system where available memory was a limiting factor for the computation, forcing an "out-of-core" computation.

The principal issue with standard FFT algorithms in a hierarchical memory system (e.g. where disk is one level of the hierarchy) is that at least K complete passes over the data are required for a convolution of length 2^K. However disk access is typically a couple of orders of magnitude slower than memory access, making such algorithms prohibitively slow.

The first FFT technique to deal with a memory hierarchy is that of Gentleman and Sande [20]. The method has become known as Bailey's Four Step method (in the context of complex FFT's), see [3]. The idea is to break the data into a two dimensional array and perform small FFT's in the horizontal and then in the vertical directions, with certain "twiddle factors" applied between the two stages. A final transpose stage then follows. This basic strategy is also sometimes referred to as the Matrix Fourier Algorithm.

Bailey's method can be extended to a six (or five) step three dimensional method and beyond. See the above cited paper of Bailey's for older references,

or [31] for a more recent reference. For applications to integer multiplication, see for example [21].

Some other algorithms for out-of-core FFT's include the algorithm of Cormen [10] based on the in-core method of Swarztrauber, the method of Takahashi [37] for the Parallel Disk Model (PDM) of Vitter and Shriver and the parallel FFT method of Vitter and Shriver [40] for a two level memory system.

Another technique commonly used for out-of-core FFT computations is the method of performing Number Theoretic Transforms (NTTs) with Chinese Remainder Theorem reconstitution.

A Number Theoretic Transform is an FFT in the ring $R = \mathbb{Z}/p\mathbb{Z}$ for a specially chosen small prime p sometimes called an "FFT prime". Usually p is chosen to fit into a single machine word, i.e. 32 or 64 bits. For this to work, R must have sufficiently many roots of unity to support the convolution.

FFT primes p can be chosen to be of the form $p = m2^K + 1$ for some small value m. Let x be a primitive root modulo p, i.e. a value x such that $x^{p-1} \equiv 1 \pmod{p}$, but such that x^a is not $1 \pmod{p}$ for any value of a dividing $p - 1$. Then x^m is a 2^K-th root of unity, supporting convolutions of length 2^K.

In order to perform an out-of-core polynomial multiplication $h(x) = f_A(x) \times g_B(x)$ using NTTs the coefficients of the two polynomials are first reduced modulo a number of FFT primes. Then the Chinese Remainder Algorithm can be used to reconstitute the full product from the results of the NTTs.

The NTT transform method is a standard one for computing large numbers of digits of π. See for example the paper of Bailey, [4] where two FFT primes were used, in that case to avoid the necessity of quad-precision arithmetic in a complex FFT. The same paper also mentions a proposal to use three FFT primes, even avoiding double precision arithmetic in the NTT's, but imposing severe restriction on the length of convolution possible for machines of that era.

More recently Carey Bloodworth's record-holding programs used eight NTTs and CRT, and were topped in 2004 by the program of Xavier Gourdon [16] for greatest number of digits of π computed on a home computer. Gourdon's program uses an unspecified number of NTTs.

More recent than our theta computation is the record π computation of Fabrice Bellard [6], using NTTs and a home computer. For out-of-core operations, his computation made use of eight 64 bit moduli, however for in-core components he made use of floating point arithmetic and unproven, heuristically chosen error bounds on the precision required.

4 The Power Series Multiplication

For any method using FFTs, optimised for out-of-core operation, the main bottleneck becomes disk I/O. To minimise this, it is not only important to minimise the number of passes over the data, but also to minimise the amount of data that must be traversed.

Two issues arise. Firstly, techniques such as the Schönhage-Strassen technique are difficult to optimise for convolution lengths which are not a power of two, in the worst case increasing the disk I/O by a factor of two.

Secondly, when performing a large FFT or a small number of very large NTTs that do not fit into memory, even when combined with Bailey's technique, truncation of the polynomial multiplications occurs *after* each large FFT computation. In other words, the disk I/O occurs for the entire *untruncated* FFT computation.

For multiplication of integers of n bits, these methods require a total disk I/O of $12n$ bits with a peak usage of $8n$ bits. Our technique reduces this to a total disk I/O of just over $6n$ bits with a peak usage of just over $4n$ bits. This is achieved by efficient multimodular reduction and CRT recombination using a large number of small primes p with truncation occurring *in-core*.

One advantage of using NTTs is that the primes p can be chosen in such a way that reduction modulo p can be performed very efficiently. E.g. for primes p of the form $2^K + 1$ reduction modulo p can be performed with subtractions rather than expensive divisions. More generally, many primes of the form $p = m2^K + 1$ for small values of m can be used. Reduction modulo p can still be computed relatively efficiently.

For our computation we chose to use many *general* word sized primes p and an alternative method of performing polynomial multiplications over $\mathbb{Z}/p\mathbb{Z}$. For the largest polynomial multiplications, in the 1 (mod 8) and 3 (mod 8) cases, we used just over 500 primes.

The main reason for this choice was the existence of well-tested, high performance packages for doing such computations, such as FLINT [18] and zn_poly [22]. There was also an advantage in having two separate implementations of arithmetic in $\mathbb{Z}/p\mathbb{Z}[x]$ in that comparisons could be made between the two implementations whilst testing. The implementation of multiplication in $\mathbb{Z}/p\mathbb{Z}[x]$ in zn_poly is highly optimised. It offers a thread-safe, cache-efficient, truncated, Schönhage-Nussbaumer convolution [21], which performs significantly better than other implementations for general primes p.

In contrast, Victor Shoup's NTL package [34] was the only library we were aware of with asymptotically fast NTTs. However NTL is not threadsafe. Also, numerous recent improvements in polynomial arithmetic are not reflected in NTL, which is no longer under active development.

Our implementation made use of 16 CPU cores. The data for all 16 threads must be in memory simultaneously, and thus to benefit from the disk-to-memory ratio of the multimodular approach it was necessary to use a number of primes significantly larger than this.

One disadvantage of using so many primes is that multimodular reduction and CRT reconstruction constitute a significant part of the runtime. The naive approach is to reduce the large coefficients of the polynomials in $\mathbb{Z}[x]$ modulo each of the primes p in turn and to similarly reconstruct each coefficient one prime at a time. However for n_1 coefficients in $\mathbb{Z}$ of n_2 bits, reconstruction using this approach will take time $O(n_1 n_2^2)$. This is asymptotically much worse than the time required to do the actual polynomial multiplications over $\mathbb{Z}/p\mathbb{Z}$.

In order to avoid this, a divide-and-conquer approach was used for the multimodular reduction and recombination phases. This completes the CRT recombination in time $O(n_2 \log^2 n_1 n_2)$ ignoring smaller log log factors. Note that this is asymptotically a log factor greater than the time for the multiplications, however the running time is still quasilinear in the input size.

The extra theoretical complexity of our approach is offset by the "embarrasingly parallel" nature of the multiplications, multimodular reduction and recombinations and the large saving in disk I/O (by far the bottleneck for our computation).

For a straightforward description of the divide-and-conquer approach to the CRT algorithm see [41], pages 57–58. Similar preconditioning and a divide-and-conquer approach was of course applied to the multimodular reduction phase. A slight adjustment was also made to both the reduction and CRT phases to cope with a number of primes which is not a power of 2.

4.1 The Algorithm in Pseudocode

We now describe our algorithm in full. We make use of two sets of disk files, $\mathcal{F} = \{F_i : i = 0, 1, .., FILES - 1\}$ and $\mathcal{G} = \{G_j : j = 0, 1, .., FILES - 1\}$. In our implementation we used FILES = 500 for the 1 (mod 8) and 3 (mod 8) computations and half that in the 2 (mod 16) and 10 (mod 16) computations.

We also set: LIMIT (the length of the theta functions), BLOCK (number of theta coefficients computed at a time), BUNDLE (number of theta coefficients bundled, using Kronecker Segmentation, into each large polynomial coefficient) and THREADS (number of threads used), PRIMES (number of primes used in multimodular reduction and CRT). We experimented with various values for BUNDLE from 500 to 1000. To simplify the computation, PRIMES was rounded up to a multiple THREADS. The value LIMIT, ($10^{12}/8$ in the $1, 3$ (mod 8) cases and $10^{12}/16$ in the $2, 10$ (mod 16) cases), was chosen to be a multiple of FILES×BUNDLE, and a multiple of FILES×BLOCK.

Coefficients of the product of our θ-series comfortably fit into 16 signed bits. Thus the Kronecker Segmentation phase used zero-padded fields of 16 bits.

Throughout the following we write FOR $i = 0$ to A and similar expressions, by which we mean i in $0 <= i < A$.

The algorithm is presented in 3 stages, corresponding to file read/write phases. The first phase bundles coefficients of the θ-functions theta$_A$, theta$_B$ using Kronecker Segmentation, to produce polynomials $f_A, f_B \in \mathbb{Z}[x]$ with multiprecision coefficients. It then reduces each coefficient of f_A, f_B modulo each of the word sized primes, forming a matrix, which is then transposed and written to disk.

Algorithm 1 : Phase 1

PRIMES ←ceil$(2 \times 16 \times$BUNDLE$/62) + 1$
PRIMES ← ceil(PRIMES$/16) \times 16$
primes$[0]$ ← nextprime(2^{62})
for $k = 1$ to PRIMES **do**
  primes$[k]$ ← nextprime(primes$[k-1]$)

end for
blocksize $\leftarrow LIMIT/FILES$
for $i = 0$ to FILES **do**
  for $l = 0$ to blocksize/BLOCK **do**
    for $m = 0$ to BLOCK **do**
      theta[l×BLOCK+m] = theta$_A$(i×blocksize+l×BLOCK+m)
    end for
  end for
  for $j = 0$ to blocksize/BUNDLE **do**
    for $r = 0$ to BUNDLE **do**
      a_r =theta[j×BUNDLE+r]
    end for
    $B \leftarrow 2^{16}$
    $c_j \leftarrow a_0 + a_1 B + a_2 B^2 + \cdots + a_{s-1} B^{s-1}$, s = BUNDLE
  end for
  $f_i \leftarrow c_0 + c_1 x + \cdots + c_{t-1} x^{t-1} \in \mathbb{Z}[x]$, t = blocksize/BUNDLE
  for $j = 0$ to t **do**
    for $k = 0$ to PRIMES **do**
      $M_1[j][k] \leftarrow c_j \pmod{\text{primes}[k]}$
    end for
  end for
  Transpose M_1 and write to file F_i
end for
Repeat above for theta function θ_B, writing transposes of M_2 to files G_i

The second phase of the algorithm reads the data stored in the files F_i and G_j and multiplies the polynomials in $\mathbb{Z}/p\mathbb{Z}$ for each of the PRIMES primes p, truncating the results and storing them back in the files F_i.

Algorithm 1 : Phase 2
for $i = 0$ to PRIMES **do**
  for $j = 0$ to FILES **do**
    Read block j of $M_1[i]$ from line i of file F_j
    for $k = 0$ to blocksize/BUNDLE **do**
      $a_{k+j\cdot t} \leftarrow M_1[i][k + j \cdot t]$, where t = blocksize/BUNDLE
    end for
  end for
  $fp(x) \leftarrow a_0 + a_1 x + \cdots + a_{t-1} x^{t-1}$
  for $j = 0$ to FILES **do**
    Read block j of $M_2[i]$ from line i of file G_j
    for $k = 0$ to blocksize/BUNDLE **do**
      $b_{k+j\cdot t} \leftarrow M_2[i][k + j \cdot t]$, where t = blocksize/BUNDLE
    end for
  end for
  $gp(x) \leftarrow b_0 + b_1 x + \cdots + b_{t-1} x^{t-1}$
  $hp(x) = c_0 + c_1 x + c_2 x^2 \ldots \leftarrow fp(x) \times gp(x)$
  Truncate $hp(x)$ to length blocksize/BUNDLE

for $j = 0$ to FILES **do**
for $k = 0$ to blocksize/BUNDLE **do**
$M_1[i][k + j \cdot t] \leftarrow c_{k+j \cdot t}$, where $t =$ blocksize/BUNDLE
end for
Write block j of $M_1[i]$ to line i of file F_j
Delete file G_j
end for
end for

The final phase of the algorithm reconstitutes the product polynomial $H = f_A \times f_B \in \mathbb{Z}[x]$ using the preconditioned, divide-and-conquer CRT mentioned above, overlaps and adds the coefficients of H to make a large integer (not all stored in memory at once), extracts the theta function product coefficients from bit fields of this integer and counts zeroes and performs other statistical computations on these small product coefficients. We sieved out non-squarefree indices so that we were counting primitive congruent numbers.

Algorithm 1 : Phase 3
Let $t_1 = a_0 + a_1 2^D + a_2 \cdot 2^{2D}$,
Let $t_2 = b_0 + b_1 2^D + b_2 \cdot 2^{2D}$,
Let $t_3 = c_0 + c_1 2^D + c_2 \cdot 2^{2D}$, {with a_i, b_i, c_i fields of D bits initialised to 0 and $a_i, b_i, c_i < 2^D$}
Read block 0 of M from file F_0
Transpose M
for $i = 0$ to blocksize/BUNDLE **do**
$d_i \leftarrow$ CRT($M[i][0] \pmod{\text{primes}[0]}, \ldots, M[i][t-1] \pmod{\text{primes}[t-1]}$)
end for
$t_1 \leftarrow d_0$; $v \leftarrow 0$; carry $\leftarrow 0$; $j \leftarrow 0$
while v < LIMIT **do**
carry, $T \leftarrow a_0 + b_1 + c_2 +$ carry, where T is D bits
Extract BUNDLE coefficients from T, count zeroes, compute stats
$t_3 \leftarrow t_2$
$t_2 \leftarrow t_1$
$v \leftarrow v+$BUNDLE
if $v \equiv 0 \pmod{\text{blocksize}}$ **and** $v <$ LIMIT **then**
$s \leftarrow v/$blocksize
Read block s of M from file F_s
Transpose M
for $i = 0$ to blocksize/BUNDLE, (using THREADS threads) **do**
$d_i \leftarrow$ CRT($M[i][0] \pmod{\text{primes}[0]}, \ldots, M[i][t-1] \pmod{\text{primes}[t-1]}$)
end for
$j \leftarrow 0$
end if
$t_1 \leftarrow d_j$
$j \leftarrow j + 1$
end while

The theta function computation, Kronecker segmentation, multimodular reduction, matrix transposes, $\mathbb{Z}/p\mathbb{Z}[x]$ polynomial multiplications and CRT recombination phases were all parallelised (trivially) using OpenMP pragmas. For disk access, the mmap kernel service was used, allowing memory blocks to be mapped to files. The kernel then schedules reading and writing of the files automatically.

5 Results and Analysis

Our θ-products were all constructed to be divisible by 2 or 4 (with the possible exception of a single -1 value). The frequency of each possible coefficient value from -2^{15} to 2^{15} was recorded. Thus if a coefficient were off by 1 this would be detected as a nonzero count for a value that was not divisible by 2 or 4. In particular, if an overflow occurred, the overflowed value would have the wrong sign. Thus an extra borrow would propagate to the next coefficient (or not propagate when it should). This would be indicated by a value that was out by 1.

The ability to likely detect overflows is important, because no good bound exist for the size of the initial theta coefficients in the series we are multiplying.

The computation was done on a 4 × Quad Core AMD Opteron server running at 2.4GHz. The memory was 128GB of registered ECC memory, capable of detecting and correcting single bit errors. The disk array consisted of 4 drives in RAID 5 arrangement (with parity stripe), for about 1.3TB of available space.

Each of the 1 (mod 8) and 3 (mod 8) computations could be performed by the first algorithm in about 30 hours real time, on this machine. Each of the 2 (mod 16) and 10 (mod 16) computations took around 9 hours.

Around the same time David Harvey, Robert Bradshaw and the third author completed the same computation using an implementation of Bailey's four step algorithm. This allowed for verification of the results. Statistics agreed between the two computations in all congruence classes.

In Tables 1-4 we present some statistics from the computation, namely the number of zeroes in bins from 0 to 10^{12}. The results are presented per residue

Table 1. Congruent numbers in the 1 (mod 8) class

10^9	10^{10}	10^{11}	2×10^{11}	3×10^{11}	4×10^{11}
3801661	21768969	142778019	127475330	115249740	107930081
5×10^{11}	6×10^{11}	7×10^{11}	8×10^{11}	9×10^{11}	10^{12}
102774355	98817294	95656907	93030373	90748990	88803354

Table 2. Congruent numbers in the 3 (mod 8) class

10^9	10^{10}	10^{11}	2×10^{11}	3×10^{11}	4×10^{11}
2921535	17019170	112979066	101436853	91949066	86213764
5×10^{11}	6×10^{11}	7×10^{11}	8×10^{11}	9×10^{11}	10^{12}
82196846	79106503	76626341	74546400	72781203	71239101

Table 3. Congruent numbers in the 2 (mod 16) class

10^9	10^{10}	10^{11}	2×10^{11}	3×10^{11}	4×10^{11}
2110645	12294626	81759844	73445274	66579936	62455317
5×10^{11}	6×10^{11}	7×10^{11}	8×10^{11}	9×10^{11}	10^{12}
59536672	57282587	55504389	53993974	52728711	51619397

Table 4. Congruent numbers in the 10 (mod 16) class

10^9	10^{10}	10^{11}	2×10^{11}	3×10^{11}	4×10^{11}
1842072	10842882	72556705	65378932	59347550	55720114
5×10^{11}	6×10^{11}	7×10^{11}	8×10^{11}	9×10^{11}	10^{12}
53152609	51190025	49599296	48268971	47158661	46159584

class. Note that only primitive, i.e. squarefree, congruent numbers are counted. Each zero is only counted in one bin, e.g. the 10^{10} bin counts all zeroes in $(10^9, 10^{10}]$.

6 Future Improvements

Numerous improvements to our method are possible.

- The matrix transposes could be performed in a cache efficient way.
- The second polynomial is sparse. David Harvey suggested that its multimodular reduction can be stored on disk in a fraction of the space using a sparse representation. This trick roughly halves the peak disk usage and I/O.
- The mmap service does not guarantee reading or writing of the data sequentially. A substantial speedup can be obtained if disk access occurs sequentially and reading of data begins before it is needed.
- It would be interesting to try number theoretic transforms in place of the current zn_poly code for polynomial multiplication over $\mathbb{Z}/p\mathbb{Z}$.
- Our implementation did not try to parallelise the CRT reconstruction phase, and the use of Montgomery's REDC might speed up the recombination here.
- It may be more efficient to allocate one thread for I/O and use 15 threads for computation instead of 16, allowing I/O in parallel with computation.

Numerous other interesting θ-series and modular forms await investigation, e.g. the Mordell curve, or the congruent number-like series of Yoshida [42]. We ourselves have looked at L-series of symmetric powers of elliptic curves.

Acknowledgements

Thanks to Mike Rubinstein for challenging us with the problem of computing the congruent number theta function to 10^{12} in 2008. Thanks to William Stein for allowing us to use the SAGE cluster of machines, funded by National Science

Foundation grant No. DMS-0821725. Thanks to J. B. Tunnell for pointing out a careless mistake in an earlier draft of this manuscript. We thank Noam D. Elkies for pointing out that the theta functions of Tunnell could be decomposed, making the theta products more manageable. We thank the anonymous referees who provided a number of minor corrections.

Many thanks to the American Institute of Mathematics for their involement in the workshops at which much of the collaboration occurred. A special thanks to David Farmer, Estelle Basor, Kent Morrison, Sally Koutsoliotas and Brian Conrey of AIMath for their careful preparation of a web page providing details of our computation for the general public.

References

1. Alter, R., Curtz, T.B., Kubota, K.K.: Remarks and results on congruent numbers. In: Proc. Third Southeastern Conf. on Combinatorics, Graph Theory and Computing, pp. 27–35 (1972)
2. Argüello, F., Amor, M., Zapata, E.L.: Implementation of parallel FFT algorithms on distributed memory machines with a minimum overhead of communication. Parallel Comput. 22(9), 1255–1279 (1996)
3. Bailey, D.H.: FFTs in external or hierarchical memory. J. Supercomput. 4, 23–35 (1990)
4. Bailey, D.H.: The computation of π to $29,360,000$ decimal digits using Borweins' quartically convergent algorithm. Math. Comp. 50(181), 283–296 (1988)
5. Bastien, L.: Nombres congruents. Intermédiaire Math. 22, 231–232 (1915)
6. Bellard, F.: Computation of 2700 billion decimal digits of Pi using a Desktop Computer (2010), http://bellard.org/pi/pi2700e9/pipcrecord.pdf
7. Birch, B.J., Stephens, N.M.: The parity of the rank of the Mordell-Weil group. Topology 5, 295–299 (1966)
8. Calvin, C.: Implementation of parallel FFT algorithms on distributed memory machines with a minimum overhead of communication. Parallel Comput. 22(9), 1255–1279 (1996)
9. Cooley, J.W., Tukey, J.W.: An algorithm for the machine calculation of complex Fourier series. Math. Comput. 19, 297–301 (1965)
10. Cormen, T.H.: Determining an out-of-core FFT decomposition strategy for parallel disks by dynamic programming. In: Algorithms for Parallel Processing IMA, Math. Appl., vol. 105, pp. 307–320. Springer, Heidelberg (1999)
11. Dickson, L.E.: History of the Theory of Numbers II. Carnegie Intitute of Washington (1920); Reprinted Chelsea (1966)
12. Elkies, N.D.: Online tables, http://www.math.harvard.edu/~elkies/compnt.html
13. Feng, K.: Non-congruent Numbers, Odd graphs and the B-S-D Conjecture. Acta Arith. LXXV(1), 71–83 (1996)
14. Feng, K., Xue, Y.: New series of odd non-congruent numbers. Science in China Series A: Mathematics 49(11), 1642–1654 (2006)
15. GMP: The GNU Multi-Precision Library, http://gmplib.org/
16. Gourdon, X.: PiFast prime digit program (2004), http://numbers.computation.free.fr/Constants/PiProgram/pifast.html
17. Guy, R.K.: Unsolved Problems in Number Theory. Springer, Heidelberg (2004)
18. Hart, W.B.: Fast Library for Number Theory (FLINT), www.flintlib.org

19. Heideman, M.T., Johnson, D.H., Burrus, C.S.: Gauss and the history of the fast Fourier transform. IEEE ASSP Magazine 1(4), 14–21 (1984)
20. Gentleman, W.M., Sande, G.: Fast Fourier Transforms - For Fun and Profit. In: AFIPS Proceedings, vol. 29, pp. 563–578 (1966)
21. Harvey, D.: A cache-friendly truncated FFT. Theor. Comput. Sci. 410, 2649–2658 (2009)
22. Harvey, D.: *zn_poly*, http://www.cims.nyu.edu/~harvey/zn_poly/index.html
23. Lagrange, J.: Thèse d'Etat de l'Université de Reims (1976)
24. Johnsson, S.L., Jacquemin, M., Krawitz, R.L.: Communication efficient multiprocessor FFT. J. Comput. Phys. 102(2), 381–397 (1992)
25. Koblitz, N.: Introduction to Elliptic Curves and Modular Forms, 2nd edn. Springer, Heidelberg (1993)
26. Kramarz, G.: All congruent numbers less than 2000. Math. Annalen 273, 337–340 (1986)
27. Lemmermeyer, F.: Some families of non-congruent numbers. Acta. Arith. 110, 15–36 (2003)
28. Monsky, P.: Mock Heegner Points and Congruent Numbers. Math. Z. 204, 45–68 (1990)
29. MPIR: Multiple Precision Integers and Rationals, http://www.mpir.org/
30. Ono, K.: The web of modularity: Arithmetic of the coefficients of modular forms and q-series. In: CBMS Conference Series, vol. 102. Amer. Math. Soc., Providence (2004)
31. Na'mneh, R.A., Pan, D.W.: Five-step FFT algorithm with reduced computational complexity. Inform. Process. Lett. 101(6), 262–267 (2007)
32. Rogers, N.F.: Rank computations for the congruent number elliptic curves. Eperiment. Math. 9(4), 591–594 (2000)
33. Schönhage, A., Strassen, V.: Schnelle Multiplikation grosser Zahlen. Computing 7(3-4), 281–292 (1971)
34. Shoup, V.: NTL: Number Theory Library, http://www.shoup.net/ntl/
35. Stephens, N.M.: Congruence properties of congruent numbers. Bull. London Math. Soc. 7, 182–184 (1975)
36. Swarztrauber, P.: Multiprocessor FFTs. Proceedings of the international conference on vector and parallel computing—issues in applied research and development (Loen, 1986). Parallel Comput. 5(1-2), 197–210 (1987)
37. Takahashi, D.: Calculation of π to 51.5 billion decimal digits on distributed memory parallel processors. Trans. Inform. Process. Soc. Japan 39(7), 2074–2083 (1998)
38. Temperton, C.: Implementation of a prime factor FFT algorithm on CRAY-1. Parallel Comput. 6(1), 99–108 (1988)
39. Tunnell, J.B.: A classical diophantine problem and modular forms of weight 3/2. Invent. Math. 72, 323–334 (1983)
40. Vitter, J.S., Shriver, E.A.M.: Algorithms for parallel memory. I. Two-level memories. Algorithmica 12(2-3), 110–147 (1994)
41. Winkler, F.: Polynomial Algorithms in Computer Algebra. Springer, Heidelberg (1996)
42. Yoshida, S.-i.: Some variants of the congruent number problem, I, II. Kyushu J. Math. 55(2), 387–404 (2001), 56(1), 147–165 (2002)

Pairing the Volcano

Sorina Ionica[1] and Antoine Joux[1,2]

[1] Université de Versailles Saint-Quentin-en-Yvelines, 45 avenue des États-Unis, 78035 Versailles CEDEX, France
[2] DGA
{sorina.ionica,antoine.joux}@m4x.org

Abstract. Isogeny volcanoes are graphs whose vertices are elliptic curves and whose edges are ℓ-isogenies. Algorithms allowing to travel on these graphs were developed by Kohel in his thesis (1996) and later on, by Fouquet and Morain (2001). However, up to now, no method was known, to predict, before taking a step on the volcano, the direction of this step. Hence, in Kohel's and Fouquet-Morain algorithms, we take many steps before choosing the right direction. In particular, ascending or horizontal isogenies are usually found using a trial-and-error approach. In this paper, we propose an alternative method that efficiently finds all points P of order ℓ such that the subgroup generated by P is the kernel of an horizontal or an ascending isogeny. In many cases, our method is faster than previous methods.

1 Introduction

Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$, where $q = p^r$ is a prime power. Let π be the Frobenius endomorphism, i.e. $\pi(x, y) \mapsto (x^q, y^q)$ and denote by t its trace. Assume that E is an ordinary curve and let $\mathcal{O}_E$ denotes its ring of endomorphisms. We know [21, Th. V.3.1] that $\mathcal{O}_E$ is an order in an imaginary quadratic field K. Let $d_\pi = t^2 - 4q$ be the discriminant of π. We can write $d_\pi = g^2 d_K$, where d_K is the discriminant of the quadratic field K. There are only a finite number of possibilities for $\mathcal{O}_E$, since $\mathbb{Z}[\pi] \subset \mathcal{O}_E \subset \mathcal{O}_{d_K}$. Indeed, this requires that f the conductor of $\mathcal{O}_E$ divides g the conductor of $\mathbb{Z}[\pi]$.

The cardinality of E over $\mathbb{F}_q$ is $\#E(\mathbb{F}_q) = q + 1 - t$. Two isogenous elliptic curves over $\mathbb{F}_q$ have the same cardinality, and thus the same trace t. In his thesis [14], Kohel studies how curves in $\mathrm{Ell}_t(\mathbb{F}_q)$, the set of curves defined over $\mathbb{F}_q$ with trace t, are related via isogenies of degree ℓ. More precisely, he describes the structure of the graph of ℓ-isogenies defined on $\mathrm{Ell}_t(\mathbb{F}_q)$. He relates this graph to orders in $\mathcal{O}_K$ and uses modular polynomials to find the conductor of $\mathrm{End}(E)$.

Fouquet and Morain [8] call the connected components of this graph *isogeny volcanoes* and extend Kohel's work. In particular, they give an algorithm that computes the ℓ-adic valuation of the trace t, for $\ell | g$. This can be used in Schoof's algorithm [20]. Recently, more applications of isogeny volcanoes were found: the computation of Hilbert class polynomials [1,23], of modular polynomials [4] and of endomorphism rings of elliptic curves [2].

G. Hanrot, F. Morain, and E. Thomé (Eds.): ANTS-IX 2010, LNCS 6197, pp. 201–218, 2010.

All the above methods make use of algorithms for traveling efficiently on volcanoes. These algorithms either need to walk on the crater, to descend from the crater to the floor or to ascend from the floor to the crater. In many cases, the structure of the ℓ-Sylow subgroup of the elliptic curve, allows, after taking a step on the volcano, to decide whether this step is ascending, descending or horizontal (see [16,17]). Note that, since a large fraction of isogenies are descending, finding one of them is much easier. However, no known method can find horizontal or ascending isogenies without using a trial-and-error approach. In this paper, we describe a first solution to this open problem, which applies when the cardinality of the curve is known, and propose a method that efficiently finds a point P of order ℓ that spans the kernel of an ascending (or horizontal isogeny). Our approach relies on the computation of a few pairings on E. We then show that our algorithms for traveling on the volcano are, in many cases, faster than the ones from [14] and [8]. Moreover, we obtain a simple method that detects most curves on the crater of their volcano. Until now, the only curves that were easily identified were those on the floor of volcanoes.

This paper is organized as follows: sections 2 and 3 present definitions and properties of isogeny volcanoes and pairings. Section 4 explains our method to find ascending or horizontal isogenies using pairing computations. Finally, in Section 5, we use this method to improve the algorithms for ascending a volcano and for walking on its crater.

2 Background on Isogeny Volcanoes

In this paper, we rely on some results from complex multiplication theory and on Deuring's lifting theorems. We denote by $\mathcal{Ell}_d(\mathbb{C})$ the set of $\mathbb{C}$-isomorphism classes of elliptic curves whose endomorphism ring is the order $\mathcal{O}_d$, with discriminant $d < 0$. In this setting there is an action of the class group of $\mathcal{O}_d$ on $\mathcal{Ell}_d(\mathbb{C})$. Let $E \in \mathcal{Ell}_d(\mathbb{C})$, Λ its corresponding lattice and $\mathfrak{a}$ an $\mathcal{O}_d$-ideal. We have a canonical homomorphism from $\mathbb{C}/\Lambda$ to $\mathbb{C}/\mathfrak{a}^{-1}\Lambda$ which induces an isogeny usually denoted by $E \to \hat{\mathfrak{a}} * E$. This action on $\mathcal{Ell}_d(\mathbb{C})$ is transitive and free [22, Prop. II.1.2]. Moreover [22, Cor. II.1.5], the degree of the application $E \to \hat{\mathfrak{a}} * E$ is $N(\mathfrak{a})$, the norm of the ideal $\mathfrak{a}$. Now from Deuring's theorems [6], if p is a prime number that splits completely, we get a bijection $\mathcal{Ell}_d(\mathbb{C}) \to \mathcal{Ell}_d(\mathbb{F}_q)$, where $q = p^r$. Furthermore, the class group action in characteristic zero respects this bijection, and we get an action of the class group also on $\mathcal{Ell}_d(\mathbb{F}_q)$.

Isogeny volcanoes. Consider E an elliptic curve defined over a finite field $\mathbb{F}_q$. Let ℓ be a prime different from $\mathrm{char}(\mathbb{F}_q)$ and $I : E \to E'$ be an ℓ-isogeny, i.e. an isogeny of degree ℓ. As shown in [14], this means that $\mathcal{O}_E$ contains $\mathcal{O}_{E'}$ or $\mathcal{O}_{E'}$ contains $\mathcal{O}_E$ or the two endomorphism rings coincide. If $\mathcal{O}_E$ contains $\mathcal{O}_{E'}$, we say that I is a *descending* isogeny. Otherwise, if $\mathcal{O}_E$ is contained in $\mathcal{O}_{E'}$, we say that I is a *ascending* isogeny. If $\mathcal{O}_E$ and $\mathcal{O}_{E'}$ are equal, then we call the isogeny *horizontal*. In his thesis, Kohel shows that horizontal isogenies exist only if the conductor of $\mathcal{O}_E$ is not divisible by ℓ. Moreover, in this case there are exactly

$\left(\frac{d}{\ell}\right)+1$ horizontal ℓ-isogenies, where d is the discriminant of $\mathcal{O}_E$. If $\left(\frac{d}{\ell}\right)=1$, then ℓ is split in $\mathcal{O}_E$ and the two horizontal isogenies correspond to the two actions $E \to \hat{\mathfrak{l}} * E$ and $E \to \hat{\bar{\mathfrak{l}}} * E$, where the two ideals $\mathfrak{l}$ and $\bar{\mathfrak{l}}$ satisfy $(\ell)=\mathfrak{l}\bar{\mathfrak{l}}$. In a similar way, if $\left(\frac{d}{\ell}\right)=0$, then ℓ is ramified, i.e. $(\ell)=\mathfrak{l}^2$ and there is exactly one horizontal isogeny starting from E. In order to describe the structure of the graph whose vertices are curves with a fixed number of points and whose edges are ℓ-isogenies, we recall the following definition [23].

Definition 1. *An ℓ-volcano is a connected undirected graph with vertices partitioned into levels $V_0, \ldots, V_h$, in which a subgraph on V_0 (the crater) is a regular connected graph of degree at most 2 and*

(a) For $i > 0$, each vertex in V_i has exactly one edge leading to a vertex in V_{i-1}, and every edge not on the crater is of this form.
(b) For $i < h$, each vertex in V_i has degree $\ell + 1$.

We call the level V_h *the floor* of the volcano. Vertices lying on the floor have degree 1. The following proposition [23] follows essentially from [14, Prop. 23].

Proposition 1. *Let p be a prime number, $q = p^r$, and $d_\pi = t^2 - 4q$. Take $\ell \neq p$ another prime number. Let G be the undirected graph with vertex set $Ell_t(\mathbb{F}_q)$ and edges ℓ-isogenies defined over $\mathbb{F}_q$. We denote by ℓ^h the largest power of ℓ dividing the conductor of d_π. Then the connected components of G that do not contain curves with j-invariant 0 or 1728 are ℓ-volcanoes of height h and for each component V, we have :*

(a) The elliptic curve whose j-invariants lie in V_0 have endomorphism rings isomorphic to some $\mathcal{O}_{d_0} \supseteq \mathcal{O}_{d_\pi}$ whose conductor is not divisible by ℓ.
(b) The elliptic curve whose j-invariants lie in V_i have endomorphism rings isomorphic to $\mathcal{O}_{d_i}$, where $d_i = \ell^{2i} d_0$.

Elliptic curves are determined by their j-invariant, up to a twist[1]. Throughout the paper, we refer to a vertex in a volcano by giving the curve or its j-invariant.

Exploring the volcano. Given a curve E on an ℓ-volcano, two methods are known to find its neighbours. The first method relies on the use of modular polynomials. The ℓ-th *modular polynomial*, denoted by $\Phi_\ell(X, Y)$ is a polynomial with integer coefficients. It satisfies the following property: given two elliptic curves E and E' with j-invariants $j(E)$ and $j(E')$ in $\mathbb{F}_q$, there is an ℓ-isogeny defined over $\mathbb{F}_q$, if and only if, $\#E(\mathbb{F}_q) = \#E'(\mathbb{F}_q)$ and $\Phi_\ell(j(E), j(E')) = 0$. As a consequence, the curves related to E via an ℓ-isogeny can be found by solving $\Phi_\ell(X, j(E)) = 0$. As stated in [20], this polynomial[2] may have 0, 1, 2 or $\ell + 1$ roots in $\mathbb{F}_q$. In order to find an edge on the volcano, it suffices to find a root j' of this polynomial. Finally, if we need the equation of the curve E' with j-invariant j', we may use the formula in [20].

The second method to build ℓ-isogenous curves constructs, given a point P of order ℓ on E, the ℓ-isogeny $I : E \to E'$ whose kernel G is generated by P using

[1] For a definition of twists of elliptic curves, refer to [21].
[2] The case where the modular polynomial does not have any root corresponds to a degenerate case of isogeny volcanoes containing a single curve and no ℓ-isogenies.

Vélu's classical formulae [24] in an extension field $\mathbb{F}_{q^r}$. To use this approach, we need the explicit coordinates of points of order ℓ on E. We denote by G_i, $1 \leq i \leq \ell+1$, the $\ell+1$ subgroups of order ℓ of E. In [17], Miret and al. give the degree r_i of the smallest extension field of $\mathbb{F}_q$ such that $G_i \subset \mathbb{F}_{q^{r_i}}$, $1 \leq i \leq \ell+1$. This degree is related to the order of q in the group $\mathbb{F}_\ell^*$, that we denote by $\mathrm{ord}_\ell(q)$.

Proposition 2. *Let E defined over $\mathbb{F}_q$ be an elliptic curve with k rational ℓ-isogenies, $\ell > 2$, and let G_i, $1 \leq i \leq k$, be their kernels, and let r_i be the minimum value for which $G_i \subset E(\mathbb{F}_{q^{r_i}})$.*

(a) If $k = 1$ then $r_1 = ord_\ell(q)$ or $r_1 = 2ord_\ell(q)$.
(b) If $k = \ell + 1$ then either $r_i = ord_\ell(q)$ for all i, or $r_i = 2ord_\ell(q)$ for all i.
(c) If $k = 2$ then $r_i | \ell - 1$ for $i = 1, 2$.

We also need the following corollary [17].

Corollary 1. *Let $E/\mathbb{F}_q$ be an elliptic curve over $\mathbb{F}_q$ and $\tilde{E}$ its twist. If $E/\mathbb{F}_q$ has 1 or $\ell+1$ rational ℓ-isogenies, then $\#E(\mathbb{F}_{q^{ord_\ell q}})$ or $\#\tilde{E}(\mathbb{F}_{q^{ord_\ell q}})$ is a multiple of ℓ. Moreover, if there are $\ell+1$ rational isogenies, then it is a multiple of ℓ^2.*

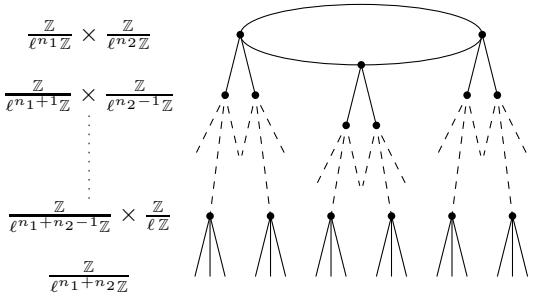


Fig. 1. A regular volcano

The group structure of the elliptic curve on the volcano. Lenstra [13] relates the group structure of an elliptic curve to its endomorphism ring by proving that $E(\mathbb{F}_q) \simeq \mathcal{O}_E/(\pi - 1)$ as $\mathcal{O}_E$-modules. It is thus natural to see how this structure relates to the isogeny volcano. From Lenstra's equation, we can deduce that $E(\mathbb{F}_q) \simeq \mathbb{Z}/M\mathbb{Z} \times \mathbb{Z}/N\mathbb{Z}$. We write $\pi = a + g\omega$, with:

$$a = \begin{cases} (t-g)/2 \\ t/2 \end{cases} \text{ and } \omega = \begin{cases} \frac{1+\sqrt{d_K}}{2} & \text{if } d_K \equiv 1 \pmod 4 \\ \sqrt{d_K} & \text{if } d_K \equiv 2, 3 \pmod 4 \end{cases}$$

where d_K is the discriminant of the quadratic imaginary field containing $\mathcal{O}_E$. Note that N is maximal such that $E[N] \subset E(\mathbb{F}_q)$ and by [19, Lemma 1] we get that $N = \gcd(a-1, g/f)$. Note moreover that $N|M$, $N|(q-1)$ and $MN = \#E(\mathbb{F}_q)$. This implies that on a ℓ-volcano the structure of all the curves in a given level is the same.

Let E be a curve on the isogeny volcano such that $v_\ell(N) < v_\ell(M)$. As explained in [16] (in the case $\ell = 2$, but the result is general), a is such that $v_\ell(a-1) \geq \min\{v_\ell(g), v_\ell(\#E(\mathbb{F}_q))/2\}$.

Since $N = \gcd(a-1, g/f)$ and $v_\ell(N) \leq v_\ell(\#E(\mathbb{F}_q))/2$, it follows that $v_\ell(N) = v_\ell(g/f)$. As we descend, the valuation at ℓ of the conductor f increases by 1 at each level (by proposition 1b). This implies that the ℓ-valuation of N for curves at each level decreases by 1 and is equal to 0 for curves lying on the floor. Note that if $v_\ell(\#E(\mathbb{F}_q))$ is even and the height h of the volcano is greater than $v_\ell(\#E(\mathbb{F}_q))$, the structure of the ℓ-torsion group is unaltered from the crater down to the level $h - v_\ell(\#E(\mathbb{F}_q))/2$. From this level down, the structure of the ℓ-torsion groups starts changing as explained above. In the sequel, we call this level the *first stability level*.[3] A volcano with first stability level equal to 0, i.e. on the crater, is called *regular*.

Notations. Let $n \geq 0$. We denote by $E[\ell^n]$ the ℓ^n-torsion subgroup, i.e. the subgroup of points of order ℓ^n on the curve $E(\bar{\mathbb{F}}_q)$, by $E[\ell^n](\mathbb{F}_{q^k})$ the subgroup of points of order ℓ^n defined over an extension field of $\mathbb{F}_q$ and by $E[\ell^\infty](\mathbb{F}_q)$ the ℓ-Sylow subgroup of $E(\mathbb{F}_q)$.

Given a point $P \in E[\ell^n](\mathbb{F}_q)$, we also need to know the degree of the smallest extension field containing an ℓ^{n+1}-torsion point such that $\ell\tilde{P} = P$. The following result is taken from [7].

Proposition 3. *Let $E/\mathbb{F}_q$ be an elliptic curve which lies on a ℓ-volcano whose height $h(V)$ is different from 0. Then the height of V', the ℓ-volcano of the curve $E/\mathbb{F}_{q^s}$ is $h(V') = h(V) + v_\ell(s)$.*

From this proposition, it follows easily that if the structure of ℓ-torsion on the curve $E/\mathbb{F}_q$ is $\mathbb{Z}/\ell^{n_1}\mathbb{Z} \times \mathbb{Z}/\ell^{n_2}\mathbb{Z}$, then the smallest extension in which the structure of the ℓ-torsion changes is $\mathbb{F}_{q^\ell}$. We sketch here the proof in the case $n_1 = n_2 = n$, which is the only case in which we consider volcanoes over extension fields in this paper[4]. First of all, note that E lies on a ℓ-volcano $V/\mathbb{F}_q$ of height at least n. We consider a curve E' lying on the floor of $V/\mathbb{F}_q$ such that there is a descending path of isogenies between E and E'. Obviously, we have $E'[\ell^\infty](\mathbb{F}_q) \simeq \mathbb{Z}/\ell^{2n}\mathbb{Z}$. By proposition 3, $V/\mathbb{F}_{q^\ell}$ has one extra down level, which means that the curve E' is no longer on the floor, but on the level just above the floor. Consequently, we have that $E'[\ell] \subset E'(\mathbb{F}_{q^\ell})$ and, moreover, $E'[\ell^\infty](\mathbb{F}_{q^\ell}) \simeq \mathbb{Z}/\ell^{2n+\Delta}\mathbb{Z} \times \mathbb{Z}/\ell\mathbb{Z}$. By ascending on the volcano from E' to E, we deduce that the structure of the ℓ-torsion of E over $\mathbb{F}_{q^\ell}$ is necessarily

$$E[\ell^\infty](\mathbb{F}_{q^\ell}) \simeq \mathbb{Z}/\ell^{n+\Delta}\mathbb{Z} \times \mathbb{Z}/\ell^{n+1}\mathbb{Z}.$$

Moreover, $\Delta \geq 1$, because if it were 0, the height of $V/\mathbb{F}_{q^\ell}$ would be n.

[3] Miret et al. call it simply *the stability level*.

[4] For the proof in the general case, see [11].

3 Background on Pairings

Let E be an elliptic curve defined over some finite field $\mathbb{F}_q$, m a number such that $m | \gcd(\#E(\mathbb{F}_q), q-1)$. Let $P \in E[m](\mathbb{F}_q)$ and $Q \in E(\mathbb{F}_q)/mE(\mathbb{F}_q)$. Let $f_{m,P}$ be the function whose divisor[5] is $m(P) - m(O)$, where O is the point at infinity of the curve E. Take R a random point in $E(\mathbb{F}_q)$ such as the support of the divisor $D = (Q+R) - (R)$ is disjoint from the support of $f_{m,P}$. Then we can define the Tate pairing as follows:

$$\begin{aligned} t_m : E[m] \times E(\mathbb{F}_q)/mE(\mathbb{F}_q) &\to \mathbb{F}_q^*/(\mathbb{F}_q^*)^m \\ (P, Q) &\to f_{m,P}(Q+R)/f_{m,P}(R). \end{aligned}$$

The Tate pairing is a bilinear non-degenerate application, i.e. for all $P \in E[m](\mathbb{F}_q)$ different from O there is a $Q \in E(\mathbb{F}_q)/mE(\mathbb{F}_q)$ such that $T_m(P,Q) \neq 1$. The output of the pairing is only defined up to a coset of $(\mathbb{F}_q^*)^m$. However, for implementation purposes, it is useful to have a uniquely defined value and to use the *reduced* Tate pairing, i.e. $T_m(P,Q) = t_m(P,Q)^{(q-1)/m} \in \mu_m$, where μ_m denotes the group of m-th roots of unity. Pairing computation can be done in time $O(\log m)$ using Miller's algorithm [15]. For more details and properties of pairings, the reader can refer to [9]. Note that in the recent years, in view of cryptographic applications, many implementation techniques have been developed and pairings on elliptic curves can be computed very efficiently[6].

Suppose now that $m = \ell^n$, with $n \geq 1$ and ℓ prime. Now let P and Q be two ℓ^n-torsion points on E. We define the following symmetric pairing [12]

$$S(P,Q) = (T_{\ell^n}(P,Q)\, T_{\ell^n}(Q,P))^{\frac{1}{2}}. \tag{1}$$

Note that for any point P, $T_{\ell^n}(P,P) = S(P,P)$. In the remainder of this paper, we call $S(P,P)$ *the self-pairing* of P. We focus on the case where the pairing S is non-constant. Suppose now that P and Q are two linearly independent ℓ^n-torsion points. Then all ℓ^n-torsion points R can be expressed as $R = aP + bQ$. Using bilinearity and symmetry of the S-pairing, we get

$$\log(S(R,R)) = a^2 \log(S(P,P)) + 2ab \log(S(P,Q)) + b^2 \log(S(Q,Q)) \pmod{\ell^n},$$

where log is a discrete logarithm function in μ_{ℓ^n}. We denote by k the largest integer such that the polynomial

$$\mathcal{P}(a,b) = a^2 \log(S(P,P)) + 2ab \log(S(P,Q)) + b^2 \log(S(Q,Q)) \tag{2}$$

is identically zero modulo ℓ^k and nonzero modulo ℓ^{k+1}. Obviously, since S is non-constant we have $0 \leq k < n$. Dividing by ℓ^k, we may thus view $\mathcal{P}$ as a polynomial in $\mathbb{F}_\ell[a,b]$. When we want to emphasize the choice of E and ℓ^n, we write $\mathcal{P}_{E,\ell^n}$ instead of $\mathcal{P}$.

[5] For background on divisors, see [21].
[6] See [10] for a fast recent implementation.

Since $\mathcal{P}$ is a non-zero quadratic polynomial, it has at most two homogeneous roots, which means that that from all the $\ell+1$ subgroups of $E[\ell^n]/E[\ell^{n-1}] \simeq (\mathbb{Z}/\ell\mathbb{Z})^2$, at most 2 have self-pairings in μ_{ℓ^k} (see also [12]). In the remainder of this paper, we denote by N_{E,ℓ^n} the number of zeros of $\mathcal{P}_{E,\ell^n}$. Note that this number does not depend on the choice of the two generators P and Q of the ℓ^n-torsion subgroup $E[\ell^n]$. Moreover, we say that a ℓ^n-torsion point R has *degenerate self-pairing* if $T_{\ell^n}(R,R)$ is a ℓ^k-th root of unity and that R has *non-degenerate self-pairing* if $T_{\ell^n}(R,R)$ is a primitive ℓ^{k+1}-th root of unity. Also, if $T_{\ell^n}(R,R)$ is a primitive ℓ^n-th root of unity, we say that R has *primitive self-pairing.*

4 Determining Directions on the Volcano

In this section, we explain how we can distinguish between different directions on the volcano by making use of pairings. We give some lemmas explaining the relations between pairings on two isogenous curves.

Lemma 1. *Suppose $E/\mathbb{F}_q$ is an elliptic curve and P,Q are points in $E(\mathbb{F}_q)$ of order ℓ^n, $n \geq 1$. Denote by $\tilde{P},\tilde{Q} \in E[\bar{\mathbb{F}}_q]$ the points such that $\ell\tilde{P} = P$ and $\ell\tilde{Q} = Q$. We have the following relations for the Tate pairing*

(a) If $\tilde{P},\tilde{Q} \in E[\mathbb{F}_q]$, then $T_{\ell^{n+1}}(\tilde{P},\tilde{Q})^{\ell^2} = T_{\ell^n}(P,Q)$.

(b) Suppose $\ell \geq 3$. If $\tilde{Q} \in E[\mathbb{F}_{q^\ell}]\backslash E[\mathbb{F}_q]$, then $T_{\ell^{n+1}}(\tilde{P},\tilde{Q})^{\ell} = T_{\ell^n}(P,Q)$.

Proof. a. By writing down the divisors of the functions $f_{\ell^{n+1},\tilde{P}}$, $f_{\ell^n,\tilde{P}}$, $f_{\ell^n,P}$, one can easily check that

$$f_{\ell^{n+1},\tilde{P}} = (f_{\ell,\tilde{P}})^{\ell^n} \cdot f_{\ell^n,P}.$$

We evaluate these functions at some points $Q+R$ and R (where R is carefully chosen) and raise the equality to the power $(q-1)/\ell^n$.
b. Due to the equality on divisors $\mathrm{div}(f_{\ell^{n+1},P}) = \mathrm{div}(f^{\ell}_{\ell^n,P})$, we have

$$T_{\ell^{n+1}}(\tilde{P},\tilde{Q})^{\ell} = T_{\ell^n}^{(\mathbb{F}_{q^\ell})}(P,\tilde{Q}),$$

where $T_{\ell^n}^{(\mathbb{F}_{q^\ell})}$ is the ℓ^n-Tate pairing for E defined over $\mathbb{F}_{q^\ell}$. It suffices then to show that $T_{\ell^n}^{(\mathbb{F}_{q^\ell})}(P,\tilde{Q}) = T_{\ell^n}(P,Q)$. We have

$$\begin{aligned} T_{\ell^n}^{(\mathbb{F}_{q^\ell})}(P,\tilde{Q}) &= f_{\ell^n,P}([\tilde{Q}+R]-[R])^{\frac{(1+q+\cdots+q^{\ell-1})(q-1)}{\ell^n}} \\ &= f_{\ell^n,P}((\tilde{Q}+R)+(\pi(\tilde{Q})+R)+(\pi^2(\tilde{Q})+R)+\ldots \\ &\quad +(\pi^{\ell-1}(\tilde{Q})+R)-\ell(R))^{\frac{(q-1)}{\ell^n}} \end{aligned} \tag{3}$$

where R is a random point defined over $\mathbb{F}_q$. It is now easy to see that for $\ell \geq 3$,

$$\tilde{Q}+\pi(\tilde{Q})+\pi^2(\tilde{Q})+\ldots+\pi^{\ell-1}(\tilde{Q}) = \ell\tilde{Q} = Q,$$

because $\pi(\tilde{Q}) = \tilde{Q} + T$, where T is a point of order ℓ. By applying Weil's reciprocity law [21, Ex. II.2.11], it follows that the equation (3) becomes:

$$T_{\ell^n}^{(\mathbb{F}_{q^\ell})}(P, \tilde{Q}) = \left(\frac{f_{\ell^n,P}(Q+R)}{f_{\ell^n,P}(R)}\right)^{\frac{q-1}{\ell^n}} f((P)-(O))^{q-1},$$

where f is such that $\mathrm{div}(f) = (\tilde{Q}+R) + (\pi(\tilde{Q})+R) + (\pi^2(\tilde{Q})+R) + ... + (\pi^{\ell-1}(\tilde{Q})+R) - (Q+R) - (\ell-1)(R)$. Note that this divisor is $\mathbb{F}_q$-rational, so $f((P)-(O))^{q-1} = 1$. This concludes the proof.

Lemma 2. *(a) Let $\phi : E \to E'$ be a separable isogeny of degree d defined over $\mathbb{F}_q$, P a ℓ-torsion on the curve E such that $\phi(P)$ is a ℓ-torsion point on E', and Q a point on E. Then we have $T_\ell(\phi(P), \phi(Q)) = T_\ell(P,Q)^d$.*
(b) Let $\phi : E \to E'$ be a separable isogeny of degree ℓ defined over $\mathbb{F}_q$, P a $\ell\ell'$-torsion point such that $Ker\,\phi = \langle \ell' P \rangle$ and Q a point on the curve E. Then we have $T_\ell(\phi(P), \phi(Q)) = T_{\ell\ell'}(P,Q)^\ell$.

Proof. Proof omitted for lack of space. See [3, Th. IX.9.4] for (a), [11] for (b).

Proposition 4. *Let E be an elliptic curve defined a finite field $\mathbb{F}_q$ and assume that $E[\ell^\infty](\mathbb{F}_p)$ is isomorphic to $\mathbb{Z}/\ell^{n_1}\mathbb{Z} \times \mathbb{Z}/\ell^{n_2}\mathbb{Z}$ (with $n_1 \geq n_2$). Suppose that there is a ℓ^{n_2}-torsion point P such that $T_{\ell^{n_2}}(P,P)$ is a primitive ℓ^{n_2}-th root of unity. Then the ℓ-isogeny whose kernel is generated by $\ell^{n_2-1}P$ is descending. Moreover, the curve E does not lie above the first stability level of the corresponding ℓ-volcano.*

Proof. Let $I_1 : E \to E_1$ be the isogeny whose kernel is generated by $\ell^{n_2-1}P$ and suppose this isogeny is ascending or horizontal. This means that $E_1[\ell^{n_2}]$ is defined over $\mathbb{F}_q$. Take Q another ℓ^{n_2}-torsion point on E, such that $E[\ell^{n_2}] = \langle P, Q \rangle$ and denote by $Q_1 = I_1(Q)$. One can easily check that the dual of I_1 has kernel generated by $\ell^{n_2-1}Q_1$. It follows that there is a point $P_1 \in E_1[\ell^{n_2}]$ such that $P = \hat{I}_1(P_1)$. By Lemma 2 this means that $T_\ell(P,P) \in \mu_{\ell^{n_2-1}}$, which is false. This proves not only that the isogeny is descending, but also that the structure of the ℓ-torsion is different at the level of E_1. Hence E cannot be above the stability level.

Proposition 5. *Let $\ell \geq 3$ a prime number and suppose that $E/\mathbb{F}_q$ is a curve which lies in a ℓ-volcano and on the first stability level. Suppose $E[\ell^\infty](\mathbb{F}_q) \simeq \mathbb{Z}/\ell^{n_1}\mathbb{Z} \times \mathbb{Z}/\ell^{n_2}\mathbb{Z}$, $n_1 \geq n_2$. Then there is at least one ℓ^{n_2}-torsion point $R \in E(\mathbb{F}_q)$ with primitive self-pairing.*

Proof. Let P be a ℓ^{n_1}-torsion point and Q be a ℓ^{n_2}-torsion point such that $\{P, Q\}$ generates $E[\ell^\infty](\mathbb{F}_q)$.

Case 1. Suppose $n_1 \geq n_2 \geq 2$. Let $E \xrightarrow{I_1} E_1$ be a descending ℓ-isogeny and denote by P_1 and Q_1 the ℓ^{n_1+1} and ℓ^{n_2-1}-torsion points generating $E_1[\ell^\infty](\mathbb{F}_p)$. Moreover, without loss of generality, we may assume that $I_1(P) = \ell P_1$ and $I_1(Q) = Q_1$. If $T_{\ell^{n_2-1}}(Q_1, Q_1)$ is a primitive ℓ^{n_2-1}-th root of unity, $T_{\ell^{n_2}}(Q,Q)$ is

a primitive ℓ^{n_2}-th root of unity by Lemma 2. If not, from the non-degeneration of the pairing, we deduce that $T_{\ell^{n_2-1}}(Q_1, P_1)$ is a primitive ℓ^{n_2-1}-th root of unity, which means that $T_{\ell^{n_2-1}}(Q_1, \ell P_1)$ is a ℓ^{n_2-2}-th primitive root of unity. By applying Lemma 2, we get $T_{\ell^{n_2}}(Q, P) \in \mu_{\ell^{n_2-1}}$ at best. It follows that $T_{\ell^{n_2}}(Q, Q) \in \mu_{\ell^{n_2}}$ by the non-degeneracy of the pairing.
Case 2. If $n_2 = 1$, then consider the volcano defined over the extension field $\mathbb{F}_{q^\ell}$. There is a ℓ^2-torsion point $\tilde{Q} \in E(\mathbb{F}_{q^\ell})$ with $Q = \ell\tilde{Q}$. We obviously have $\ell^2 | q^\ell - 1$ and from Lemma 1, we get $T_{\ell^2}(\tilde{P}, \tilde{P})^\ell = T_\ell(P, P)$. By applying Case 1, we get that $T_{\ell^2}(\tilde{P}, \tilde{P})$ is a primitive ℓ^2-th root of unity, so $T_\ell(P, P)$ is a primitive ℓ-th root of unity.

Two stability levels. Remember that in any irregular volcano, $v_\ell(\#E(\mathbb{F}_q))$ is even and the height h of the volcano is greater than $v_\ell(\#E(\mathbb{F}_q))$. Moreover, all curves at the top of the volcano have $E[\ell^\infty](\mathbb{F}_q) \simeq \mathbb{Z}/\ell^{n_2}\mathbb{Z} \times \mathbb{Z}/\ell^{n_2}\mathbb{Z}$ with $n_2 = v_\ell(\#E(\mathbb{F}_q))$. The existence of a primitive self-pairing of a ℓ^{n_2}-torsion point on any curve lying on the first stability level implies that the polynomial $\mathcal{P}$ is non-zero at every level from the first stability level up to the level $\max(h+1-2n_2, 0)$ (by Lemma 2). We call this level *the second level of stability*. On the second stability level there is at least one point of order ℓ^{n_2} with pairing equal to a primitive ℓ-th root of unity. At every level above the second stability level all polynomials $\mathcal{P}_{E,\ell^{n_2}}$ may be zero[7]. Consider now E a curve on the second stability level and $I : E \to E_1$ an ascending isogeny. Let P be a ℓ^{n_2}-torsion point on E and assume that $T_{\ell^{n_2}}(P, P) \in \mu_\ell^*$. We denote by $\tilde{P} \in E(\mathbb{F}_{q^\ell}) \backslash E(\mathbb{F}_q)$ the point such that $\ell\tilde{P} = P$. By Lemma 1 we get $T_{\ell^{n_2+1}}(\tilde{P}, \tilde{P})$ is a primitive ℓ^2-th root of unity. It follows by Lemma 2 that $T_{\ell^{n_2}}(I(P), I(P))$ is a primitive ℓ-th root of unity. We deduce that $\mathcal{P}_{E_1,\ell^{n_2+1}}$ corresponding to $E_1/\mathbb{F}_{q^\ell}$ is non-zero. Applying this reasoning repeatedly, we conclude that for every curve E above the second stability level there is an extension field $\mathbb{F}_{q^{s\ell}}$ such that the polynomial $\mathcal{P}_{E,\ell^{n_2+s}}$ associated to the curve defined over $\mathbb{F}_{q^{s\ell}}$ is non-zero. When the second stability level of a volcano is 0, we say that the volcano is *almost regular*.

We now make use of a result on the representation of ideal classes of orders in imaginary quadratic fields. This is Corollary 7.17 from [5].

Lemma 3. *Let $\mathcal{O}$ be an order in an imaginary quadratic field. Given a nonzero integer M, then every ideal class in $Cl(\mathcal{O})$ contains a proper $\mathcal{O}$-ideal whose norm is relatively prime to M.*

Proposition 6. *We use the notations and assumptions from Proposition 1. Furthermore, we assume that for all curves E_i lying at a fixed level i in V the curve structure is $\mathbb{Z}/\ell^{n_1}\mathbb{Z} \times \mathbb{Z}/\ell^{n_2}\mathbb{Z}$, with $n_1 \geq n_2$. The value of $N_{E_i,\ell^{n_2}}$, the number of zeros of the polynomial defined at 2, is constant for all curves lying at level i in the volcano.*

Proof. Let E_1 and E_2 be two curves lying at level i in the volcano V. Then by Proposition 1 they both have endomorphism ring isomorphic to some order $\mathcal{O}_{d_i}$.

[7] In all the examples we considered for this case, $\mathcal{P}$ is always 0.

Now by taking into account the fact that the action of $\mathrm{Cl}(\mathcal{O}_{d_i})$ on $\mathcal{Ell}_{d_i}(\mathbb{F}_q)$ is transitive, we consider an isogeny $\phi : E_1 \to E_2$ of degree ℓ_1. By applying Lemma 3, we may assume that $(\ell_1, \ell) = 1$. Take now P and Q two independent ℓ^{n_2}-torsion points on E_1 and denote by $\mathcal{P}_{E_1,\ell^{n_2}}$ the quadratic polynomial corresponding to the ℓ^{n_2}-torsion on E_1 as in (2). We use Lemma 2 to compute $S(\phi(P), \phi(P))$, $S(\phi(P), \phi(Q))$ and $S(\phi(Q), \phi(Q))$ and deduce that a polynomial $\mathcal{P}_{E_2,\ell^{n_2}}(a, b)$ on the curve E_2 computed from $\phi(P)$ and $\phi(Q)$ is such that

$$\mathcal{P}_{E_1,\ell^{n_2}}(a, b) = \mathcal{P}_{E_2,\ell^{n_2}}(a, b).$$

This means that $N_{E_1,\ell^{n_2}}$ and $N_{E_2,\ell^{n_2}}$ coincide, which concludes the proof. Moreover, we have showed that the value of k for two curves lying on the same level of a volcano is the same.

Proposition 7. *Let E be an elliptic curve defined a finite field $\mathbb{F}_q$ and let $E[\ell^\infty](\mathbb{F}_q)$ be isomorphic to $\mathbb{Z}/\ell^{n_1}\mathbb{Z} \times \mathbb{Z}/\ell^{n_2}\mathbb{Z}$ with $\ell \geq 3$ and $n_1 \geq n_2 \geq 1$. Suppose $N_{E,\ell^{n_2}} \in \{1, 2\}$ and let P be a ℓ^{n_2}-torsion point with degenerate self-pairing. Then the ℓ-isogeny whose kernel is generated by $\ell^{n_2-1}P$ is either ascending or horizontal. Moreover, for any ℓ^{n_2}-torsion point Q whose self-pairing is non-degenerate, the isogeny with kernel spanned by $\ell^{n_2-1}Q$ is descending.*

Proof. Case 1. Suppose $T_{\ell^{n_2}}(P, P) \in \mu_{\ell^k}, k \geq 1$ and that $T_{\ell^{n_2}}(Q, Q) \in \mu_{\ell^{k+1}} \backslash \mu_{\ell^k}$. Denote by $I_1 : E \to E_1$ the isogeny whose kernel is generated by $\ell^{n_2-1}P$ and $I_2 : E \to E_2$ the isogeny whose kernel is generated by $\ell^{n_2-1}Q$. By repeatedly applying Lemmas 1 and 2, we get the following relations for points generating the ℓ^{n_2-1}-torsion on E_1 and E_2:

$$T_{\ell^{n_2-1}}(I_1(P), I_1(P)) \in \mu_{\ell^{k-1}},\ T_{\ell^{n_2-1}}(\ell I_1(Q), \ell I_1(Q)) \in \mu_{\ell^{k-2}} \backslash \mu_{\ell^{k-3}}$$
$$T_{\ell^{n_2-1}}(\ell I_2(P), \ell I_2(P)) \in \mu_{\ell^{k-3}},\ T_{\ell^{n_2-1}}(I_2(Q), I_2(Q)) \in \mu_{\ell^k} \backslash \mu_{\ell^{k-1}}$$

with the convention that $\mu_{\ell^h} = \emptyset$ whenever $h \leq 0$. From the relations above, we deduce that on the ℓ-volcano having E, E_1 and E_2 as vertices, E_1 and E_2 do not lie at the same level. Given the fact that there are at least $\ell - 1$ descending rational ℓ-isogenies parting from E and that Q is any of the $\ell - 1$ (or more) ℓ^{n_2}-torsion points with non-degenerate self-pairing, we conclude that I_1 is horizontal or ascending and that I_2 is descending.
Case 2. Suppose now that $k = 0$. Note that the case $n_2 = 1$ was already treated in proposition 4. Otherwise, consider the curve E defined over $\mathbb{F}_{q^\ell}$. By lemma 1 we have $k = 1$ for points on $E/\mathbb{F}_{q^\ell}$, and we may apply Case 1.

A special case. If E is a curve lying under the first stability level and that $E[\ell^\infty](\mathbb{F}_q) \simeq \mathbb{Z}/\ell^{n_1}\mathbb{Z} \times \mathbb{Z}/\ell^{n_2}\mathbb{Z}$, with $n_1 > n_2$, then it suffices to find a point P_1 of order ℓ^{n_1} and the point $\ell^{n_1-1}P_1$ generates the kernel of an horizontal or ascending isogeny (P_1 has degenerate self-pairing).

Crater detection. Assume that $\mathcal{P} \neq 0$. When ℓ is split in $\mathcal{O}_E$, there are two horizontal isogenies from E and this is equivalent, by propositions 6 and 7, to $N_{E,\ell^{n_2}} = 2$. Similarly, when ℓ is inert in $\mathcal{O}_E$, there are neither ascending nor

horizontal isogenies and $N_{E,\ell^{n_2}} = 0$. In these two cases, we easily detect that the curve E is on the crater.

Note. All statements in the proof of *Case* 1 are true for $\ell = 2$ also. The statement in Proposition 4 is also true for $\ell = 2$. The only case that is not clear is what happens when $k = 0$ and $n_2 \geq 1$. We did not find a proof for the statement in proposition 5 for $\ell = 2$, but in our computations with MAGMA we did not find any counterexamples either.

We conclude this section by presenting an algorithm which determines the group structure of the ℓ^∞-torsion group of a curve E and also an algorithm which outputs the kernel of an horizontal (ascending) isogeny from E, when $E[\ell^\infty](\mathbb{F}_q)$ is given.

Algorithm 1. Computing the structure of the ℓ^∞-torsion of E over $\mathbb{F}_q$ (assuming volcano height ≥ 1)

Require: A curve E defined over $\mathbb{F}_q$, a prime ℓ
Compute: Structure $\mathbb{Z}/\ell^{n_1}\mathbb{Z} \times \mathbb{Z}/\ell^{n_2}\mathbb{Z}$, generators P_1 and P_2
1: Check that $q \equiv 1 \pmod{\ell}$ (if not need to move to extension field: **abort**)
2: Let t be the trace of $E(\mathbb{F}_q)$
3: Check $q + 1 - t \equiv 0 \pmod{\ell}$ (if not consider twist or **abort**)
4: Let $d_\pi = t^2 - 4q$, let z be the largest integer such that $\ell^z | d_\pi$ and $h = \lfloor \frac{z}{2} \rfloor$
5: Let n be the largest integer such that $\ell^n | q + 1 - t$ and $N = \frac{q+1-t}{\ell^n}$
6: Take a random point R_1 on $E(\mathbb{F}_q)$, let $P_1 = N \cdot R_1$
7: Let n_1 be the smallest integer such that $\ell^{n_1} P_1 = 0$
8: **if** $n_1 = n$ **then**
9: **Output:** Structure is $\frac{\mathbb{Z}}{\ell^n\mathbb{Z}}$, generator P_1. **Exit**
(E is on the floor, ascending isogeny with kernel $\langle \ell^{n-1} P_1 \rangle$)
10: **end if**
11: Take a random point R_2 on $E(\mathbb{F}_q)$, let $P_2 = N \cdot R_2$ and $n_2 = n - n_1$
12: Let $\alpha = \log_{\ell^{n_2} P_1}(\ell^{n_2} P_2) \pmod{\ell^{n_1 - n_2}}$
13: **if** α is undefined **then**
14: **Goto** 6 ($\ell^{n_2} P_2$ does not belong to $\langle \ell^{n_2} P_1 \rangle$)
15: **end if**
16: Let $P_2 = P_2 - \alpha P_1$
17: **If** $\text{WeilPairing}_\ell(\ell^{n_1 - 1} P_1, \ell^{n_2 - 1} P_2) = 1$ **goto** 6 (This checks linear independence)
18: **Output:** Structure is $\frac{\mathbb{Z}}{\ell^{n_1}\mathbb{Z}} \times \frac{\mathbb{Z}}{\ell^{n_2}\mathbb{Z}}$, generators (P_1, P_2)

We assume that the height of the volcano is $h \leq 2n_2 + 1$, or, equivalently, that the curve E lies on or below the second stability level, which implies that the polynomial $\mathcal{P}$ is non-zero at every level in the volcano. This allows us to distinguish between different directions of ℓ-isogenies parting from E. Of course, similar algorithms can be given for curves lying above the second stability level, but in this case we are compelled to consider the volcano over an extension field $\mathbb{F}_{q^{s\ell}}$. Since computing points defined over extension fields of degree greater than ℓ is expensive, our complexity analysis in section 5 will show that it is more efficient to use Kohel's and Fouquet-Morain algorithms to explore the volcano until the second level of stability is reached and to use algorithms 1 and 2

Algorithm 2. Finding the kernel of ascending or horizontal isogenies (Assuming curve not on floor and below the second stability level)

Require: A curve E, its structure $\frac{\mathbb{Z}}{\ell^{n_1}\mathbb{Z}} \times \frac{\mathbb{Z}}{\ell^{n_2}\mathbb{Z}}$ and generators (P_1, P_2)
if $n_1 > n_2$ **then**
  The isogeny with kernel $\langle \ell^{n_1-1} P_1 \rangle$ is ascending or horizontal
  To check whether there is another, continue the algorithm
end if
Let g be a primitive ℓ-th root of unity in $\mathbb{F}_q$
Let $Q_1 = \ell^{n_1-n_2} P_1$
Let $a = T_{\ell^{n_2}}(Q_1, Q_1)$, $b = T_{\ell^{n_2}}(Q_1, P_2) \cdot T_{\ell^{n_2}}(P_2, Q_1)$ and $c = T_{\ell^{n_2}}(P_2, P_2)$
If $(a, b, c) = (1, 1, 1)$ **abort** (Above the second stability level)
repeat
  Let $a' = a$, $b' = b$ and $c' = c$
  Let $a = a^\ell$, $b = b^\ell$ and $c = c^\ell$
until $a = 1$ and $b = 1$ and $c = 1$
Let $L_a = \log_g(a')$, $L_b = \log_g(b')$ and $L_c = \log_g(c') \pmod{\ell}$
Let $\mathcal{P}(x, y) = L_a x^2 + L_b xy + L_c y^2 \pmod{\ell}$
If $\mathcal{P}$ has no roots modulo ℓ, **Output:** No isogeny (a single point on the crater)
If single root (x_1, x_2) **Output:** One isogeny with kernel $\langle \ell^{n_2-1}(x_1 Q_1 + x_2 P_2) \rangle$
if $\mathcal{P}$ has two roots (x_1, x_2) and (y_1, y_2) **then**
  Two isogenies with kernel $\langle \ell^{n_2-1}(x_1 Q_1 + x_2 P_2) \rangle$ and $\langle \ell^{n_2-1}(y_1 Q_1 + y_2 P_2) \rangle$
end if

afterwards. We assume $\ell \geq 3$, even though in many cases these methods work also for $\ell = 2$.

5 Walking the Volcano: Modified Algorithms

As mentioned in the introduction, several applications of isogeny volcanoes have recently been proposed. These applications require the ability to walk descending and ascending paths on the volcano and also to walk on the crater of the volcano. We recall that a *path* is a sequence of isogenies that never backtracks. We start this section with a brief description of existing algorithms for these tasks, based on methods given by Kohel [14] and by Fouquet and Morain in [8]. We present modified algorithms, which rely on the method presented in Algorithm 2 to find ascending or horizontal isogenies. Then, we give complexity analysis for these algorithms and show that in many cases our method is competitive. Finally, we give two concrete examples in which the new algorithms can walk the crater of an isogeny volcano very efficiently compared to existing algorithms.

A brief description of existing algorithms. Existing algorithms rely on three essential properties in isogeny volcanoes. Firstly, it is easy to detect that a curve lies on the floor of a volcano, since in that case, there is a single isogeny from this curve. Moreover, this isogeny can only be ascending (or horizontal if the height is 0). Secondly, if in an arbitrary path in a volcano there is a descending isogeny,

then all the subsequent isogenies in the path are also descending. Thirdly, from a given curve, there is either exactly one ascending isogeny or at most two horizontal ones. As a consequence, finding a descending isogeny from any curve is easy: it suffices to walk three paths in parallel until one path reaches the floor. This shortest path is necessarily descending and its length gives the level of the starting curve in the volcano. To find an ascending or horizontal isogeny, the classical algorithms try all possible isogenies until they find one which leads to a curve either at the same level or above the starting curve. This property is tested by contructing descending paths from the all the neighbours of the initial curve and picking the curve which gave the longest path.

Note that alternatively, one could walk in parallel all of the $\ell+1$ paths starting from the initial curve and keep the (two) longest as horizontal or ascending. As far as we know, this has not been proposed in the literature, but this variant of existing algorithms offers a slightly better asymptotic time complexity. For completeness, we give a pseudo-code description of this parallel variant of Kohel and Fouquet-Morain algorithms as Algorithm 3.

Algorithm 3. Parallel variant of ascending/horizontal step (using modular polynomials)

Require: A j-invariant j_0 in $\mathbb{F}_q$, a prime ℓ, the modular polynomial $\Phi_\ell(X, Y)$.
Let $f(x) = \Phi_\ell(X, j_0)$
Compute J_0 the list of roots of $f(x)$ in $\mathbb{F}_q$
If $\#J_0 = 0$ **Output:** "Trivial volcano" **Exit**
If $\#J_0 = 1$ **Output:** "On the floor, step leads to:", $J_0[1]$ **Exit**
If $\#J_0 = 2$ **Output:** "On the floor, two horizontal steps to:", $J_0[1]$ and $J_0[2]$ **Exit**

Let $J = J_0$. Let J' and K be empty lists. Let Done = **false**.
repeat
  Perform multipoint evaluation of $\Phi_\ell(X, j)$, for each $j \in J$. Store in list F
  for i from 1 to $\ell + 1$ **do**
    Perform partial factorization of $F[i]$, computing at most two roots r_1 and r_2
    if $F[i]$ has less than two roots **then**
      Let Done = **true**. Append $\perp$ to K (Reaching floor)
    else
      If $r_1 \in J'$ **then** append r_1 to K **else** append r_2 to K. (Don't backtrack)
    end if
  end for
  Let $J' = J$, $J = K$ and K be the empty list
until Done
for each i from 1 to $\ell + 1$ such that $J[i] \neq \perp$ append $J_0[i]$ to K
Output: "Possible step(s) lead to:" K (One or two outputs)

Basic idea of the modified algorithms. In our algorithms, we first need to choose a large enough extension field to guarantee that the kernels of all required isogenies are spanned by ℓ-torsion points defined on this extension field. As explained in

Corollary 1, the degree r of this extension field is the order of q modulo ℓ and it can be computed very quickly after factoring $q-1$. As usual, we choose an arbitrary irreducible polynomial of degree r to represent $\mathbb{F}_{q^r}$. The necessary points of ℓ^∞-torsion are computed in Algorithm 1, multiplying random points over $\mathbb{F}_{q^r}$ by the cardinality of the curve divided by the highest possible power of ℓ. Once this is done, assuming that we are starting from a curve below the second level of stability, we use Algorithms 1 and 2 to find all ascending or horizontal isogenies from the initial curve. In order to walk a descending path, it suffices to choose any other isogeny. Note that, in the subsequent steps of a descending path, in the cases where the group structure satisfies $n_1 > n_2$, it is not necessary to run Algorithm 2 as a whole. Indeed, since we know that we are not on the crater, there is a single ascending isogeny and it is spanned by $\ell^{n_1-1}P_1$.

Finally, above the second stability level, we have two options. In theory, we can consider curves over larger extension fields (in order to get polynomials $\mathcal{P} \neq 0$. Note that this is too costly in practice. Therefore, we use preexisting algorithms, but it is not necessary to follow descending paths all the way to the floor. Instead, we can stop these paths at the second stabilty level, where our methods can be used.

5.1 Complexity Analysis

Computing a single isogeny. Before analyzing the complete algorithms, we first compare the costs of taking a single step on a volcano by using the two methods existing in the literature: modular polynomials and classical Vélu's formulae. Suppose that we wish to take a step from a curve E. With the modular polynomial approach, we have to evaluate the polynomial $f(X) = \Phi_\ell(X, j(E))$ and find its roots in $\mathbb{F}_q$. Assuming that the modular polynomial (modulo the characteristic of $\mathbb{F}_q$) is given as input and using asymptotically fast algorithms to factor $f(X)$, the cost of a step in terms of arithmetic operations in $\mathbb{F}_q$ is $O(\ell^2 + M(\ell)\log q)$, where $M(\ell)$ denotes the operation count of multiplying polynomials of degree ℓ. In this formula, the first term corresponds to evaluation of $\Phi_\ell(X, j(E_{i-1}))$ and the second term to root finding[8].

With Vélu's formulae, we need to take into account the fact that the required ℓ-torsion points are not necessarily defined over $\mathbb{F}_q$. Let r denotes the smallest integer such that the required points are all defined over $\mathbb{F}_{q^r}$. We know that $1 \leq r \leq \ell - 1$. Using asymptotically efficient algorithms to perform arithmetic operations in $\mathbb{F}_{q^r}$, multiplications in $\mathbb{F}_{q^r}$ cost $M(r)$ $\mathbb{F}_q$-operations. Given an ℓ-torsion point P in $E(\mathbb{F}_{q^r})$, the cost of using Vélu's formulae is $O(\ell)$ operations in $\mathbb{F}_{q^r}$. As a consequence, in terms of $\mathbb{F}_q$ operations, each isogeny costs $O(\ell M(r))$ operations. As a consequence, when q is not too large and r is close to ℓ, using Vélu formulae is more expensive by a logarithmic factor.

[8] Completely splitting $f(X)$ to find all its roots would cost $O(M(\ell)\log \ell \log q)$, but this is reduced to $O(M(\ell)\log q)$ because we only need a constant number of roots for each polynomial $f(X)$.

Computing an ascending or horizontal path. With the classical algorithms, each step in an ascending or horizontal path requires to try $O(\ell)$ steps and test each by walking descending paths of height bounded by h. The cost of each descending path is $O(h(\ell^2 + M(\ell)\log q))$ and the total cost is $O(h(\ell^3 + \ell M(\ell)\log q))$ (see [14,23]). When $\ell >> \log q$, this cost is dominated by the evaluations of the polynomial Φ_ℓ at each j-invariant. Thus, by walking in parallel $\ell + 1$ paths from the original curve, we can amortize the evaluation of $\Phi_\ell(X, j)$ over many j-invariants using fast multipoint evaluation, see [18, Section 3.7] or [25], thus replacing ℓ^3 by $\ell M(\ell)\log\ell$ and reducing the complexity of a step to $O(h\ell M(\ell)(\log\ell + \log q))$. However, this increases the memory requirements.

With our modified algorithms, we need to find the structure of each curve, compute some discrete logarithms in ℓ-groups, perform a small number of pairing computations and compute the roots of $\mathcal{P}_{E,\ell^{n_2}}$. Except for the computation of discrete logarithms, it is clear that all these additional operations are polynomial in n_2 and $\log\ell$ and they take negligible time in practice (see Section 5.2). Using generic algorithms, the discrete logarithms cost $O(\sqrt{\ell})$ operations, and this can be reduced to $\log\ell$ by storing a sorted table of precomputed logarithms. After this is done, we have to compute at most two isogenies, ignoring the one that backtracks. Thus, the computation of one ascending or horizontal step is dominated by the computation of isogenies and costs $O(\ell M(r))$.

For completeness, we also mention the complexity analysis of Algorithm 1. The dominating step here is the multiplication by N of randomly chosen points. When we consider the curve over an extension field $\mathbb{F}_{q^r}$, this costs $O(r\log q)$ operations in $\mathbb{F}_{q^r}$, i.e. $O(rM(r)\log q)$ operations in $\mathbb{F}_q$.

Finally, comparing the two approaches on a regular volcano, we see that even in the less favorable case, we gain a factor h compared to the classical algorithms. More precisely, the two are comparable, when the height h is small and r is close to ℓ. In all the other cases, our modified algorithms are more efficient. This analysis is summarized in Table 1. For compactness $O(\cdot)$s are omitted from the table.

Table 1. Walking the volcano: Order of the cost per step

	Descending path		Ascending/Horizontal
	One step	Many steps	
[14,8]	$h(\ell^2 + M(\ell)\log q)$	$(\ell^2 + M(\ell)\log q)$	$h(\ell^3 + \ell M(\ell)\log q)$
Parallel evaluation	–	–	$h\ell M(\ell)(\log\ell + \log q)$
Regular volcanoes	Structure determination		
Best case	$\log q$		$\log q$
Worst case $r \approx \ell/2$	$r M(r)\log q$		$r M(r)\log q$
Regular volcanoes	Isogeny construction		
Best case	ℓ		ℓ
Worst case $r \approx \ell/2$	$r M(r)$		$r M(r)$
Irregular volcanoes (worst case)	No improvement		

Irregular volcanoes. Consider a fixed value of q and let $s = v_\ell(q-1)$. First of all, note that all curves lying on irregular volcanoes satisfy $\ell^{2s}|q+1-t$ and $\ell^{2s+2}|t^2-4q$. For traces that satisfy only the first condition, we obtain a regular volcano. We estimate the total number of different traces of elliptic curves lying on ℓ-volcanoes by $\#\{t \text{ s.t. } \ell^{2s}|q+1-t \text{ and } t \in [-2\sqrt{q}, 2\sqrt{q}]\} \sim \frac{4\sqrt{q}}{\ell^{2s}}$.
Next, we estimate traces of curves lying on irregular volcanoes by

$$\#\{t \text{ s.t. } \ell^{2s}|q+1-t \ , \ell^{2s+2}|t^2-4q \text{ and } t \in [-2\sqrt{q}, 2\sqrt{q}]\} \sim \frac{4\sqrt{q}}{\ell^{2s+2}}.$$

Indeed, by writing $q = 1+\gamma\ell^s$ and $t = 2+\gamma\ell^s+\mu\ell^{2s}$, and imposing the condition $\ell^{2s+2}|t^2-4q$, we find that $t \cong t_0(\gamma,\mu) (\mathrm{mod}\ \ell^{2s+2})$.

Thus, we estimate the probability of picking a curve whose volcano is not regular, among curves lying on volcanoes of height greater than 0, by $\frac{1}{\ell^2}$. (This is a crude estimate because the number of curves for each trace is proportional to the Hurwitz class number[9] $H(t^2-4q)$). This probability is not negligible for small values of ℓ. However, since our method also works everywhere on almost regular volcano, the probability of finding a volcano where we need to combine our modified algorithm with the classical algorithms is even lower. Furthermore, in some applications, it is possible to restrict ourselves to regular volcanoes.

5.2 Two Practical Examples

A favorable case. In order to demonstrate the potential of the modified algorithm, we consider the favorable case of a volcano of height 2, where all the necessary ℓ-torsion points are defined over the base field $\mathbb{F}_p$, where $p = 619074283342666852501391$ is prime. We choose $\ell = 100003$.

Let E be the elliptic curve whose Weierstrass equation is
$y^2 = x^3 + 198950713578094615678321\,x + 32044133215969807107747$.
The group $E[\ell^\infty]$ over $\mathbb{F}_p$ has structure $\frac{\mathbb{Z}}{\ell^4\mathbb{Z}}$. It is spanned by the point
$P = (110646719734315214798587, 521505339992224627932173)$.

Taking the ℓ-isogeny I_1 with kernel $\langle \ell^3 P\rangle$, we obtain the curve
$E_1 : y^2 = x^3 + 476298723694969288644436\,x + 260540808216901292162091$,
with structure of the ℓ^∞-torsion $\frac{\mathbb{Z}}{\ell^3} \times \frac{\mathbb{Z}}{\ell}$ and generators
$P_1 = (226300457529970756040 69, 207694187789705800930332)$ and
$Q_1 = (304782745358080727058129, 193904829837168032791973)$.

The ℓ-isogeny I_2 with kernel $\langle \ell^2 P_1\rangle$ leads to the curve
$E_2 : y^2 = x^3 + 212075995763000386527 90\,x + 471086215466928725193841$,
on the volcano's crater and with structure $\frac{\mathbb{Z}}{\ell^2\mathbb{Z}} \times \frac{\mathbb{Z}}{\ell^2\mathbb{Z}}$ and generators
$P_2 = (545333002760803067576755, 367548280448276783133614)$ and
$Q_2 = (401515368371004856400951, 225420044066280025495795)$.

Using pairings on these points, we construct the polynomial:

$$\mathcal{P}(x,y) = 97540\,x^2 + 68114\,x\,y + 38120\,y^2,$$

having homogeneous roots $(x,y) = (26568, 1)$ and $(72407, 1)$. As a consequence, we have two horizontal isogenies with kernels $\langle \ell(26568\,P_2 + Q_2)\rangle$ and $\langle \ell(72407\,P_2 + Q_2)\rangle$. We can continue and make a complete walk around the

[9] See [5, Th. 14.18] for q prime.

crater which contains 22 different curves. Using a simple implementation under Magma 2.15-15, a typical execution takes about 134 seconds[10] on a single core of an Intel Core 2 Duo at 2.66 GHz. Most of the time is taken by the computation of Vélu's formulas (132 seconds) and the computation of discrete logarithms (1.5 seconds) which are not tabulated in the implementation. The computation of pairings only takes 20 milliseconds.

A less favorable example. We have also implemented the computation for $\ell = 1009$ using an elliptic curve with j-invariant $j = 34098711889917$ in the prime field defined by $p = 953202937996763$. The ℓ-torsion appears in a extension field of degree 84. The ℓ-volcano has height two and the crater contains 19 curves. Our implementation walks the crater in 20 minutes. More precisely, 750 seconds are needed to generate the curves' structures, 450 to compute Vélu's formulas, 28 seconds for the pairings and 2 seconds for the discrete logarithms.

6 Conclusion and Perspectives

In this paper, we have proposed a method which allows, in the regular part of an isogeny volcano, to determine, given a curve E and a ℓ-torsion point P, the type of the ℓ-isogeny whose kernel is spanned by P. In addition, this method also permits, given a basis for the ℓ-torsion, to find the ascending isogeny (or horizontal isogenies) from E. We expect that this method can be used to improve the performance of several volcano-based algorithms, such as the computation of the Hilbert class polynomial [23] or of modular polynomials [4].

Acknowledgments. The authors thank Jean-Marc Couveignes for the idea in the proof of Lemma 1 and two anonymous reviewers for their helpful comments. The first author is grateful to Ariane Mézard for many discussions on number theory and isogeny volcanoes, prior to this work.

References

1. Belding, J., Broker, R., Enge, A., Lauter, K.: Computing Hilbert Class Polynomials. In: van der Poorten, A.J., Stein, A. (eds.) ANTS-VIII 2008. LNCS, vol. 5011, pp. 282–295. Springer, Heidelberg (2008)
2. Bisson, G., Sutherland, A.: Computing the endomorphism ring of an ordinary elliptic curve over a finite field. Journal of Number Theory (to appear 2010)
3. Blake, I.F., Seroussi, G., Smart, N.P.: Advances in Elliptic Curve Cryptography. London Mathematical Society Lecture Note Series, vol. 317. Cambridge University Press, Cambridge (2005)
4. Broker, R., Lauter, K., Sutherland, A.: Computing modular polynomials with the chinese remainder theorem (2009), http://arxiv.org/abs/1001.0402

[10] This timing varies between executions. The reason that we first try one root of $\mathcal{P}$, if it backtracks on the crater, we need to try the other one. On average, 1.5 root is tried for each step, but this varies depending on the random choices.

5. Cox, D.A.: Primes of the Form $x^2 + ny^2$: Fermat, class field theory, and complex multiplication. John Wiley & Sons, Inc., Chichester (1989)
6. Deuring, M.: Die Typen der Multiplikatorenringe elliptischer Funktionenkorper. Abh. Math. Sem. Hansischen Univ., vol. 14 (1941)
7. Fouquet, M.: Anneau d'endomorphismes et cardinalité des courbes elliptiques: aspects algorithmiques. PhD thesis, Ecole Polytechnique (2001)
8. Fouquet, M., Morain, F.: Isogeny Volcanoes and the SEA Algorithm. In: Fieker, C., Kohel, D.R. (eds.) ANTS 2002. LNCS, vol. 2369, pp. 276–291. Springer, Heidelberg (2002)
9. Frey, G.: Applications of arithmetical geometry to cryptographic constructions. In: Proceedings of the Fifth International Conference on Finite Fields and Applications, pp. 128–161. Springer, Heidelberg (2001)
10. Grabher, P., Großschädl, J., Page, D.: On software parallel implementation of cryptographic pairings. In: Avanzi, R.M., Keliher, L., Sica, F. (eds.) SAC 2008. LNCS, vol. 5381, pp. 35–50. Springer, Heidelberg (2009)
11. Ionica, S.: Algorithmique des couplages et cryptographie. PhD thesis, Université de Versailles St-Quentin-en-Yvelines (2010)
12. Joux, A., Nguyen, K.: Separating decision Diffie–Hellman from computational Diffie–Hellman in cryptographic groups. Journal of Cryptology 16(4), 239–247 (2003)
13. Lenstra Jr., H.W.: Complex multiplication structure of elliptic curves. Journal of Number Theory 56(2), 227–241 (1996)
14. Kohel, D.: Endomorphism rings of elliptic curves over finite fields. PhD thesis, University of California, Berkeley (1996)
15. Miller, V.S.: The Weil pairing, and its efficient calculation. Journal of Cryptology 17(4), 235–261 (2004)
16. Miret, J., Moreno, R., Sadornil, D., Tena, J., Valls, M.: An algorithm to compute volcanoes of 2-isogenies of elliptic curves over finite fields. Applied Mathematics and Computation 176(2), 739–750 (2006)
17. Miret, J., Moreno, R., Sadornil, D., Tena, J., Valls, M.: Computing the height of volcanoes of l-isogenies of elliptic curves over finite fields. Applied Mathematics and Computation 196(1), 67–76 (2008)
18. Montgomery, P.L.: A FFT extension of the elliptic curve method of factorization. PhD thesis, University of California (1992)
19. Ruck, H.-G.: A note on elliptic curves over finite fields. Mathematics of Computation 179, 301–304 (1987)
20. Schoof, R.: Counting points on elliptic curves over finite fields. Journal de Theorie des Nombres de Bordeaux 7, 219–254 (1995)
21. Silverman, J.H.: The Arithmetic of Elliptic Curves. Graduate Texts in Mathematics, vol. 106. Springer, Heidelberg (1986)
22. Silverman, J.H.: Advanced Topics in the Arithmetic of Elliptic Curves. Graduate Texts in Mathematics, vol. 151. Springer, Heidelberg (1994)
23. Sutherland, A.: Computing Hilbert Class Polynomials with the Chinese Remainder Theorem. Mathematics of Computation (2010)
24. Vélu, J.: Isogenies entre courbes elliptiques. Comptes Rendus De L'Academie Des Sciences Paris, Serie I-Mathematique, Serie A. 273, 238–241 (1971)
25. von zur Gathen, J., Shoup, V.: Computing Frobenius maps and factoring polynomials. Computational Complexity 2, 187–224 (1992)

A Subexponential Algorithm for Evaluating Large Degree Isogenies

David Jao and Vladimir Soukharev

Department of Combinatorics and Optimization
University of Waterloo, Waterloo, Ontario, N2L 3G1, Canada
{djao,vsoukhar}@math.uwaterloo.ca

Abstract. An isogeny between elliptic curves is an algebraic morphism which is a group homomorphism. Many applications in cryptography require evaluating large degree isogenies between elliptic curves efficiently. For ordinary curves of the same endomorphism ring, the previous best known algorithm has a worst case running time which is exponential in the length of the input. In this paper we show this problem can be solved in subexponential time under reasonable heuristics. Our approach is based on factoring the ideal corresponding to the kernel of the isogeny, modulo principal ideals, into a product of smaller prime ideals for which the isogenies can be computed directly. Combined with previous work of Bostan et al., our algorithm yields equations for large degree isogenies in quasi-optimal time given only the starting curve and the kernel.

1 Introduction

A well known theorem of Tate [29] states that two elliptic curves defined over the same finite field $\mathbb{F}_q$ are isogenous (i.e. admit an isogeny between them) if and only if they have the same number of points over $\mathbb{F}_q$. Using fast point counting algorithms such as Schoof's algorithm and others [9,25], it is very easy to check whether this condition holds, and thus whether or not the curves are isogenous. However, constructing the actual isogeny itself is believed to be a hard problem due to the nonconstructive nature of Tate's theorem. Indeed, given an ordinary curve $E/\mathbb{F}_q$ and an ideal of norm n in the endomorphism ring, the fastest previously known algorithm for constructing the unique (up to isomorphism) isogeny having this ideal as kernel has a running time of $O(n^{3+\varepsilon})$, except in a certain very small number of special cases [4,16,17]. In this paper, we present a new probabilistic algorithm for evaluating such isogenies, which in the vast majority of cases runs (heuristically) in subexponential time. Specifically, we show that for ordinary curves, one can evaluate isogenies of degree n between curves of nearly equal endomorphism ring over $\mathbb{F}_q$ in time less than $L_q(\frac{1}{2}, \frac{\sqrt{3}}{2})\log(n)$, provided n has no large prime divisors in common with the endomorphism ring discriminant. Although this running time is not polynomial in the input length, our algorithm is still much faster than the (exponential) previous best known algorithm, and in practice allows for the evaluation of isogenies of cryptographically sized degrees, some examples of which we present here. We emphasize that,

G. Hanrot, F. Morain, and E. Thomé (Eds.): ANTS-IX 2010, LNCS 6197, pp. 219–233, 2010.

in contrast with the previous results of Bröker et al. [4], our algorithm is not limited to special curves such as pairing friendly curves with small discriminant.

If an explicit equation for the isogeny as a rational function is desired, our approach in combination with the algorithm of Bostan et al. [3] can produce the equation in time $O(n^{1+\varepsilon})$ given E and an ideal of norm n, which is quasi-optimal in the sense that (up to log factors) it is equal to the size of the output. To our knowledge, this method is the only known algorithm for computing rational function expressions of large degree isogenies in quasi-optimal time in the general case, given only the starting curve and the kernel.

Apart from playing a central role in the implementation of the point counting algorithms mentioned above, isogenies have been used in cryptography to transfer the discrete logarithm problem from one elliptic curve to another [9,16,17,20,23,30]. In many of these applications, our algorithm cannot be used directly, since in cryptography one is usually given two isogenous curves, rather than one curve together with the isogeny degree. However, earlier results [16,17,20] have shown that the problem of computing isogenies between a given pair of curves can be reduced to the problem of computing isogenies of prime degree starting from a given curve. It is therefore likely that the previous best isogeny construction algorithms in the cryptographic setting can be improved or extended in light of the work that we present here.

2 Background

Let E and E' be elliptic curves defined over a finite field $\mathbb{F}_q$ of characteristic p. An isogeny $\phi\colon E \to E'$ defined over $\mathbb{F}_q$ is a non-constant rational map defined over $\mathbb{F}_q$ which is also a group homomorphism from $E(\mathbb{F}_q)$ to $E'(\mathbb{F}_q)$. This definition differs slightly from the standard definition in that it excludes constant maps [27, §III.4]. The degree of an isogeny is its degree as a rational map, and an isogeny of degree ℓ is called an ℓ-isogeny. Every isogeny of degree greater than 1 can be factored into a composition of isogenies of prime degree defined over $\bar{\mathbb{F}}_q$ [11].

For any elliptic curve $E\colon y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$ defined over $\mathbb{F}_q$, the Frobenius endomorphism is the isogeny $\pi_q\colon E \to E$ of degree q given by the equation $\pi_q(x, y) = (x^q, y^q)$. The characteristic polynomial of π_q is $X^2 - tX + q$ where $t = q + 1 - \#E(\mathbb{F}_q)$ is the trace of E.

An endomorphism of E is an isogeny $E \to E$ defined over the algebraic closure $\bar{\mathbb{F}}_q$ of $\mathbb{F}_q$. The set of endomorphisms of E together with the zero map forms a ring under the operations of pointwise addition and composition; this ring is called the endomorphism ring of E and denoted $\mathrm{End}(E)$. The ring $\mathrm{End}(E)$ is isomorphic either to an order in a quaternion algebra or to an order in an imaginary quadratic field [27, V.3.1]; in the first case we say E is supersingular and in the second case we say E is ordinary.

Two elliptic curves E and E' defined over $\mathbb{F}_q$ are said to be isogenous over $\mathbb{F}_q$ if there exists an isogeny $\phi\colon E \to E'$ defined over $\mathbb{F}_q$. A theorem of Tate states that two curves E and E' are isogenous over $\mathbb{F}_q$ if and only if $\#E(\mathbb{F}_q) = \#E'(\mathbb{F}_q)$ [29, §3]. Since every isogeny has a dual isogeny [27, III.6.1], the property of being

isogenous over $\mathbb{F}_q$ is an equivalence relation on the finite set of $\bar{\mathbb{F}}_q$-isomorphism classes of elliptic curves defined over $\mathbb{F}_q$. Moreover, isomorphisms between elliptic curves can be classified completely and computed efficiently in all cases [16]. Accordingly, we define an isogeny class to be an equivalence class of elliptic curves, taken up to $\bar{\mathbb{F}}_q$-isomorphism, under this equivalence relation.

Curves in the same isogeny class are either all supersingular or all ordinary. The vast majority of curves are ordinary, and indeed the number of isomorphism classes of supersingular curves is finite for each characteristic. Also, ordinary curves form the majority of the curves of interest in applications such as cryptography. Hence, we assume for the remainder of this paper that we are in the **ordinary case**.

Let K denote the imaginary quadratic field containing $\mathrm{End}(E)$, with maximal order $\mathcal{O}_K$. For any order $\mathcal{O} \subseteq \mathcal{O}_K$, the conductor of $\mathcal{O}$ is defined to be the integer $[\mathcal{O}_K : \mathcal{O}]$. The field K is called the CM field of E. We write c_E for the conductor of $\mathrm{End}(E)$ and c_π for the conductor of $\mathbb{Z}[\pi_q]$. It follows from [12, §7] that $\mathrm{End}(E) = \mathbb{Z} + c_E\mathcal{O}_K$ and $\Delta = c_E^2\Delta_K$, where Δ (respectively, Δ_K) is the discriminant of the imaginary quadratic order $\mathrm{End}(E)$ (respectively, $\mathcal{O}_K$). Furthermore, the characteristic polynomial has discriminant $\Delta_\pi = t^2 - 4q = \mathrm{disc}(\mathbb{Z}[\pi_q]) = c_\pi^2\Delta_K$, with $c_\pi = c_E \cdot [\mathrm{End}(E) : \mathbb{Z}[\pi_q]]$.

Following [14] and [16], we say that an isogeny $\phi\colon E \to E'$ of prime degree ℓ defined over $\mathbb{F}_q$ is "down" if $[\mathrm{End}(E) : \mathrm{End}(E')] = \ell$, "up" if $[\mathrm{End}(E') : \mathrm{End}(E)] = \ell$, and "horizontal" if $\mathrm{End}(E) = \mathrm{End}(E)$. Two curves in an isogeny class are said to "have the same level" if their endomorphism rings are equal. Within each isogeny class, the property of having the same level is an equivalence relation. A horizontal isogeny always goes between two curves of the same level; likewise, an up isogeny enlarges the endomorphism ring and a down isogeny reduces it. Since there are fewer elliptic curves at higher levels than at lower levels, the collection of elliptic curves in an isogeny class visually resembles a "pyramid" or a "volcano" [14], with up isogenies ascending the structure and down isogenies descending. If we restrict to the graph of ℓ-isogenies for a single ℓ, then in general the ℓ-isogeny graph is disconnected, having one ℓ-volcano for each intermediate order $\mathbb{Z}[\pi_q] \subset \mathcal{O} \subset \mathcal{O}_K$ such that $\mathcal{O}$ is maximal at ℓ (meaning $\ell \nmid [\mathcal{O}_K : \mathcal{O}]$). The "top level" of the class consists of curves E with $\mathrm{End}(E) = \mathcal{O}_K$, and the "bottom level" consists of curves with $\mathrm{End}(E) = \mathbb{Z}[\pi_q]$.

We say that ℓ is an *Elkies prime* [2, p. 119] if $\ell \nmid c_E$ and $\left(\frac{\Delta}{\ell}\right) \neq -1$, or equivalently if and only if E admits a horizontal isogeny of degree ℓ. The number of ℓ-isogenies of each type can easily be determined explicitly [14,16,21]. In particular, for all but the finitely many primes ℓ dividing $[\mathcal{O}_K : \mathbb{Z}[\pi_q]]$, we have that every rational ℓ-isogeny admitted by E is horizontal.

3 The Bröker-Charles-Lauter Algorithm

Our algorithm is an extension of the algorithm developed by Bröker, Charles, and Lauter [4] to evaluate large degree isogenies over ordinary elliptic curves with

endomorphism rings of small class number, such as pairing-friendly curves [15]. In this section we provide a summary of their results.

The following notation corresponds to that of [4]. Let $E/\mathbb{F}_q$ be an ordinary elliptic curve with endomorphism ring $\mathrm{End}(E)$ isomorphic to an imaginary quadratic order $\mathcal{O}_\Delta$ of discriminant $\Delta < 0$. Identify $\mathrm{End}(E)$ with $\mathcal{O}_\Delta$ via the unique isomorphism ι such that $\iota^*(x)\omega = x\omega$ for all invariant differentials ω and all $x \in \mathcal{O}_\Delta$. Then every horizontal separable isogeny on E of prime degree ℓ corresponds (up to isomorphism) to a unique prime ideal $\mathfrak{L} \subset \mathcal{O}_\Delta$ of norm ℓ for some Elkies prime ℓ. We denote the kernel of this isogeny by $E[\mathfrak{L}]$. Any two distinct isomorphic horizontal isogenies (i.e., pairs of isogenies where one is equal to the composition of the other with an isomorphism) induce different maps on the space of differentials of E, and a separable isogeny is uniquely determined by the combination of its kernel and the induced map on the space of differentials. A *normalized* isogeny is an isogeny $\phi\colon E \to E'$ for which $\phi^*(\omega_{E'}) = \omega_E$ where ω_E denotes the invariant differential of E. Algorithm 1 (identical to Algorithm 4.1 in [4]) evaluates, up to automorphisms of E, the unique normalized horizontal isogeny of degree ℓ corresponding to a given kernel ideal $\mathfrak{L} \subset \mathcal{O}_\Delta$.

The following theorem, taken verbatim from [4], shows that the running time of Algorithm 1 is polynomial in the quantities $\log(\ell)$, $\log(q)$, n, and $|\Delta|$.

Theorem 3.1. *Let $E/\mathbb{F}_q$ be an ordinary elliptic curve with Frobenius π_q, given by a Weierstrass equation, and let $P \in E(\mathbb{F}_{q^n})$ be a point on E. Let $\Delta = \mathrm{disc}(\mathrm{End}(E))$ be given. Assume that $[\mathrm{End}(E) : \mathbb{Z}[\pi_q]]$ and $\#E(\mathbb{F}_{q^n})$ are coprime, and let $\mathfrak{L} = (\ell, c + d\pi_q)$ be an $\mathrm{End}(E)$-ideal of prime norm $\ell \neq \mathrm{char}(\mathbb{F}_q)$ not dividing the index $[\mathrm{End}(E) : \mathbb{Z}[\pi_q]]$. Algorithm 7 computes the unique elliptic curve E' such that there exists a normalized isogeny $\phi\colon E \to E'$ with kernel $E[\mathfrak{L}]$. Furthermore, it computes the x-coordinate of $\phi(P)$ if $\mathrm{End}(E)$ does not equal $\mathbb{Z}[i]$ or $\mathbb{Z}[\zeta_3]$ and the square, respectively cube, of the x-coordinate of $\phi(P)$ otherwise. The running time of the algorithm is polynomial in $\log(\ell)$, $\log(q)$, n and $|\Delta|$.*

4 A Subexponential Algorithm for Evaluating Horizontal Isogenies

As was shown in Sections 2 and 3, any horizontal isogeny can be expressed as a composition of prime degree isogenies, one for each prime factor of the kernel, and any prime degree isogeny is a composition of a normalized isogeny and an isomorphism. Therefore, to evaluate a horizontal isogeny given its kernel, it suffices to treat the case of horizontal normalized prime degree isogenies.

Our objective is to evaluate the unique horizontal normalized isogeny on a given elliptic curve $E/\mathbb{F}_q$ whose kernel ideal is given as $\mathfrak{L} = (\ell, c + d\pi_q)$, at a given point $P \in E(\mathbb{F}_{q^n})$, where ℓ is an Elkies prime. As in [4], we must also impose the additional restriction that $\ell \nmid [\mathrm{End}(E) : \mathbb{Z}[\pi_q]]$; for Elkies primes, an equivalent restriction is that $\ell \nmid [\mathcal{O}_K : \mathbb{Z}[\pi_q]]$, but we retain the original formulation for consistency with [4].

Algorithm 1. The Bröker-Charles-Lauter algorithm

Input: A discriminant Δ, an elliptic curve $E/\mathbb{F}_q$ with $\mathrm{End}(E) = \mathcal{O}_\Delta$ and a point $P \in E(\mathbb{F}_{q^n})$ such that $[\mathrm{End}(E) : \mathbb{Z}[\pi_q]]$ and $\#E(\mathbb{F}_{q^n})$ are coprime, and an $\mathrm{End}(E)$-ideal $\mathfrak{L} = (\ell, c + d\pi_q)$ of prime norm $\ell \neq \mathrm{char}(\mathbb{F}_q)$ not dividing the index $[\mathrm{End}(E) : \mathbb{Z}[\pi_q]]$.

Output: The unique elliptic curve E' admitting a normalized isogeny $\phi\colon E \to E'$ with kernel $E[\mathfrak{L}]$, and the x-coordinate of $\phi(P)$ for $\Delta \neq -3, -4$ and the square (resp. cube) of the x-coordinate otherwise.

1: Compute the direct sum decomposition $\mathrm{Pic}(\mathcal{O}_\Delta) = \bigotimes \langle [I_i] \rangle$ of $\mathrm{Pic}(\mathcal{O}_\Delta)$ into cyclic groups generated by the degree 1 prime ideals I_i of smallest norm that are coprime to the product $p \cdot \#E(\mathbb{F}_{q^n}) \cdot [\mathrm{End}(E) : \mathbb{Z}[\pi_q]]$.
2: Using brute force[1], find $e_1, e_2, \ldots, e_k$ such that $[\mathfrak{L}] = [I_1^{e_1}] \cdot [I_2^{e_2}] \cdots [I_k^{e_k}]$.
3: Find α (using Cornacchia's algorithm) and express $\mathfrak{L} = I_1^{e_1} \cdot I_2^{e_2} \cdots I_k^{e_k} \cdot (\alpha)$.
4: Compute a sequence of isogenies $(\phi_1, \ldots, \phi_s)$ such that the composition $\phi_c : E \to E_c$ has kernel $E[I_1^{e_1} \cdot I_2^{e_2} \cdots I_k^{e_k}]$ using the method of [4, § 3].
5: Evaluate $\phi_c(P) \in E_c(\mathbb{F}_{q^n})$.
6: Write $\alpha = (u + v\pi_q)/(zm)$. Compute the isomorphism $\eta\colon E_c \xrightarrow{\sim} E'$ with $\eta^*(\omega_{E'}) = (u/zm)\omega_{E_c}$. Compute $Q = \eta(\phi_c(P))$.
7: Compute $(zm)^{-1} \bmod \#E(\mathbb{F}_{q^n})$, and compute $R = ((zm)^{-1}(u + v\pi_q))(Q)$.
8: Put $r = x(R)^{|\mathcal{O}_\Delta|^*/2}$ and return (E', r).

In practice, one is typically given ℓ instead of $\mathfrak{L}$, but since it is easy to calculate the list of (at most two) possible primes $\mathfrak{L}$ lying over ℓ (cf. [6]), these two interpretations are for all practical purposes equivalent, and we switch freely between them when convenient. When ℓ is small, one can use modular polynomial based techniques [4, §3.1], which have running time $O(\ell^3 \log(\ell)^{4+\varepsilon})$ [13]. However, for isogeny degrees of cryptographic size (e.g. 2^{160}), this approach is impractical. The Bröker-Charles-Lauter algorithm sidesteps this problem, by using an alternative factorization of $\mathfrak{L}$. However, the running time of Bröker-Charles-Lauter is polynomial in $|\Delta|$, and therefore even this method only works for small values of $|\Delta|$. In this section we present a modified version of the Bröker-Charles-Lauter algorithm which is suitable for large values of $|\Delta|$.

We begin by giving an overview of our approach. In order to handle large values of $|\Delta|$, there are two main problems to overcome. One problem is that we need a fast way to produce a factorization

$$\mathfrak{L} = I_1^{e_1} I_2^{e_2} \cdots I_k^{e_k} \cdot (\alpha) \tag{1}$$

as in lines 2 and 3 of Algorithm 1. The other problem is that the exponents e_i in Equation (1) need to be kept small, since the running times of lines 3 and 4 of Algorithm 1 are proportional to $\sum_i |e_i| \,\mathrm{Norm}(I_i)^2$. The first problem, that of finding a factorization of $\mathfrak{L}$, can be solved in subexponential time using the index calculus algorithm of Hafner and McCurley [18] (see also [6, Chap. 11]).

[1] Bröker, Charles, and Lauter mention that this computation can be done in "various ways" [4, p. 107], but the only explicit method given in [4] is brute force. The use of brute force limits the algorithm to elliptic curves for which $|\Delta|$ is small, such as pairing-friendly curves.

Algorithm 2. Computing a factor base

Input: A discriminant Δ, a bound N.

Output: The set $\mathcal{I}$ consisting of split prime ideals of norm less than N, together with the corresponding set $\mathcal{F}$ of quadratic forms.

1: Set $\mathcal{F} \leftarrow \emptyset$.
2: Set $\mathcal{I} \leftarrow \emptyset$.
3: Find all primes $p < N$ such that $(\frac{\Delta}{p}) = 1$. Call this set P. Let $k = |P|$.
4: For each prime $p_i \in P$, find an ideal $\mathfrak{p}_i$ of norm p_i (using Cornacchia's algorithm).
5: For each i, find a quadratic form $f_i = [(p_i, b_i, c_i)]$ corresponding to $\mathfrak{p}_i$ in $\mathrm{Cl}(\mathcal{O}_\Delta)$, using the technique of [26, §3].
6: Output $\mathcal{I} = \{\mathfrak{p}_1, \mathfrak{p}_2, \ldots, \mathfrak{p}_k\}$ and $\mathcal{F} = \{f_1, f_2, \ldots, f_k\}$.

To resolve the second problem, we turn to an idea which was first introduced by Galbraith et. al [17], and recently further refined by Bisson and Sutherland [1]. The idea is that, in the process of sieving for smooth norms, one can arbitrarily restrict the input exponent vectors to sparse vectors $(e_1, e_2, ..., e_k)$ such that $\sum_i |e_i| N(I_i)^2$ is kept small. This restriction is implemented in line 6 of Algorithm 3. As in [1], one then assumes heuristically that the imposition of this restriction does not affect the eventual probability of obtaining a smooth norm in the Hafner and McCurley algorithm. Note that, unlike the input exponents, the exponents appearing in the factorizations of the ensuing smooth norms (that is, the values of y_i in Algorithm 3) are always small, since the norm in question is derived from a reduced quadratic form.

We now describe the individual components of our algorithm in detail.

4.1 Finding a Factor Base

Let $\mathrm{Cl}(\mathcal{O}_\Delta)$ denote the ideal class group of $\mathcal{O}_\Delta$. Algorithm 2 produces a factor base consisting of split primes in $\mathcal{O}_\Delta$ of norm less than some bound N. The optimal value of N will be determined in Section 4.4.

4.2 "Factoring" Large Prime Degree Ideals

Algorithm 3, based on the algorithm of Hafner and McCurley, takes as input a discriminant Δ, a curve E, a prime ideal $\mathfrak{L}$ of prime norm ℓ in $\mathcal{O}_\Delta$, a smoothness bound N, and an extension degree n. It outputs a factorization

$$\mathfrak{L} = I_1^{e_1} I_2^{e_2} \cdots I_k^{e_k} \cdot (\alpha)$$

as in Equation 1, where the I_i's are as in Algorithm 1, the exponents e_i are positive, sparse, and small (i.e., polynomial in N), and the ideal (α) is a principal fractional ideal generated by α.

4.3 Algorithm for Evaluating Prime Degree Isogenies

The overall algorithm for evaluating prime degree isogenies is given in Algorithm 4. This algorithm is identical to Algorithm 1, except that the factorization of $\mathfrak{L}$ is performed using Algorithm 3. To maintain consistency with [4], we

Algorithm 3. "Factoring" a prime ideal

Input: A discriminant Δ, an elliptic curve $E/\mathbb{F}_q$ with $\mathrm{End}(E) = \mathcal{O}_\Delta$, a smoothness bound N, a prime ideal $\mathfrak{L}$ of norm ℓ in $\mathcal{O}_\Delta$, an extension degree n.

Output: Relation of the form $\mathfrak{L} = (\alpha) \cdot \prod_{i=1}^{k} I_i^{e_i}$, where (α) is a fractional ideal, I_i are as in Algorithm 1, and $e_i > 0$ are small and sparse.

1: Run Algorithm 2 on input Δ and N to obtain $\mathcal{I} = \{\mathfrak{p}_1, \mathfrak{p}_2, \ldots, \mathfrak{p}_k\}$ and $\mathcal{F} = \{f_1, f_2, \ldots, f_k\}$. Discard any primes dividing $p \cdot \#E(\mathbb{F}_{q^n}) \cdot [\mathrm{End}(E) : \mathbb{Z}[\pi_q]]$.
2: Set $p_i \leftarrow \mathrm{Norm}(\mathfrak{p}_i)$. (These values are also calculated in Algorithm 2.)
3: Obtain the reduced quadratic form $[\mathfrak{L}]$ corresponding to the ideal class of $\mathfrak{L}$.
4: **repeat**
5: **for** $i = 1, \ldots, k$ **do**
6: Pick exponents x_i in the range $[0, (N/p_i)^2]$ such that at most k_0 are nonzero, where k_0 is a global absolute constant (in practice, $k_0 = 3$ suffices).
7: **end for**
8: Compute the reduced quadratic form $\mathfrak{a} = (a, b, c)$ for which the ideal class $[\mathfrak{a}]$ is equivalent to $[\mathfrak{L}] \cdot \prod_{i=1}^{k} f_i^{x_i}$.
9: **until** The integer a factors completely into the primes p_i, and the relation derived from $[\mathfrak{a}] = [\mathfrak{L}] \cdot \prod_{i=1}^{k} f_i^{x_i}$ contains fewer than $\sqrt{\log(|\Delta|/3)}/z$ nonzero exponents.
10: Write $a = \prod_{i=1}^{k} p_i^{u_i}$.
11: **for** i=1, $\ldots, k$ **do**
12: Using the technique of Seysen ([26, Theorem 3.1]), determine the signs of the exponents $y_i = \pm u_i$ for which $\mathfrak{a} = \prod_{i=1}^{k} f_i^{y_i}$.
13: Let $e_i = y_i - x_i$. (These exponents satisfy $[\mathfrak{L}] = \prod_{i=1}^{k} f_i^{e_i}$.)
14: **if** $e_i \geq 0$ **then**
15: Set $I_i \leftarrow \bar{\mathfrak{p}}_i$
16: **else**
17: Set $I_i \leftarrow \mathfrak{p}_i$
18: **end if**
19: **end for**
20: Compute the principal ideal $I = \mathfrak{L} \cdot \prod_{i=1}^{k} I_i^{|e_i|}$.
21: Using Cornacchia's algorithm, find a generator $\beta \in \mathcal{O}_\Delta$ of I.
22: Set $m \leftarrow \prod_{i=1}^{k} p_i^{|e_i|}$ and $\alpha \leftarrow \frac{\beta}{m}$.
23: Output $\mathfrak{L} = (\alpha) \cdot \bar{I}_1^{|e_1|} \cdot \bar{I}_2^{|e_2|} \cdots \bar{I}_k^{|e_k|}$.

have included the quantities Δ and $\mathrm{End}(E)$ as part of the input to the algorithm. However, we remark that these quantities can be computed from $E/\mathbb{F}_q$ in $L_q(\frac{1}{2}, \frac{\sqrt{3}}{2})$ operations using the algorithm of Bisson and Sutherland [1], even if they are not provided as input.

4.4 Running Time Analysis

In this section, we determine the theoretical running time of Algorithm 4, as well as the optimal value of the smoothness bound N to use in line 1 of the algorithm. As is typical for subexponential time factorization algorithms involving a factor base, these two quantities depend on each other, and hence both are calculated simultaneously.

Algorithm 4. Evaluating prime degree isogenies

Input: A discriminant Δ, an elliptic curve $E/\mathbb{F}_q$ with $\mathrm{End}(E) = \mathcal{O}_\Delta$ and a point $P \in E(\mathbb{F}_{q^n})$ such that $[\mathrm{End}(E) : \mathbb{Z}[\pi_q]]$ and $\#E(\mathbb{F}_{q^n})$ are coprime, and an $\mathrm{End}(E)$-ideal $\mathfrak{L} = (\ell, c+d\pi_q)$ of prime norm $\ell \neq \mathrm{char}(\mathbb{F}_q)$ not dividing the index $[\mathrm{End}(E) : \mathbb{Z}[\pi_q]]$.

Output: The unique elliptic curve E' admitting a normalized isogeny $\phi\colon E \to E'$ with kernel $E[\mathfrak{L}]$, and the x-coordinate of $\phi(P)$ for $\Delta \neq -3, -4$ and the square (resp. cube) of the x-coordinate otherwise.

1: Choose a smoothness bound N (see Section 4.4).
2: Using Algorithm 3 on input $(\Delta, E, N, \mathfrak{L}, n)$, obtain a factorization of the form $\mathfrak{L} = I_1^{e_1} \cdot I_2^{e_2} \cdots I_k^{e_k} \cdot (\alpha)$.
3: Compute a sequence of isogenies $(\phi_1, \ldots, \phi_s)$ such that the composition $\phi_c : E \to E_c$ has kernel $E[I_1^{e_1} \cdot I_2^{e_2} \cdots I_k^{e_k}]$ using the method of [4, § 3].
4: Evaluate $\phi_c(P) \in E_c(\mathbb{F}_{q^n})$.
5: Write $\alpha = (u + v\pi_q)/(zm)$. Compute the isomorphism $\eta\colon E_c \xrightarrow{\sim} E'$ with $\eta^*(\omega_{E'}) = (u/zm)\omega_{E_c}$. Compute $Q = \eta(\phi_c(P))$.
6: Compute $(zm)^{-1} \bmod \#E(\mathbb{F}_{q^n})$, and compute $R = ((zm)^{-1}(u + v\pi_q))(Q)$.
7: Put $r = x(R)^{|\mathcal{O}_\Delta|^*/2}$ and return (E', r).

As in [9], we define[2] $L_n(\alpha, c)$ by

$$L_n(\alpha, c) = O(\exp((c + o(1))(\log(n))^\alpha(\log(\log(n)))^{1-\alpha})).$$

The quantity $L_n(\alpha, c)$ interpolates between polynomial and exponential size as α ranges from 0 to 1. We set $N = L_{|\Delta|}(\frac{1}{2}, z)$ for an unspecified value of z, and in the following paragraphs we determine the optimal value of z which minimizes the running time of Algorithm 4. (The fact that $\alpha = \frac{1}{2}$ is optimal is clear from the below analysis, as well as from prior experience with integer factorization algorithms.) For convenience, we will abbreviate $L_{|\Delta|}(\alpha, c)$ to $L(\alpha, c)$ throughout.

Line 2 of Algorithm 4 involves running Algorithm 3, which in turn calls Algorithm 2. As it turns out, Algorithm 2 is almost the same as Algorithm 11.1 from [6], which requires $L(\frac{1}{2}, z)$ time, as shown in [6]. The only difference is that we add an additional step where we obtain the quadratic form corresponding to each prime ideal in the factor base. This extra step requires $O(\log(\mathrm{Norm}(I))^{1+\varepsilon})$ time for a prime ideal I, using Cornacchia's Algorithm [19]. Thus, the overall running time for Algorithm 2 is bounded above by

$$L(\tfrac{1}{2}, z) \cdot \log(L(\tfrac{1}{2}, z))^{1+\varepsilon} = L(\tfrac{1}{2}, z).$$

Line 2 of Algorithm 3 takes $\log(\ell)$ time using standard algorithms [12]. The loop in lines 4–9 of Algorithm 3 is very similar to the FindRelation algorithm in [1], except that we only use one discriminant, and we omit the requirement that $\#R/D_1 > \#R/D_2$ (which in any case is meaningless when there is only one discriminant). Needless to say, this change can only speed up the algorithm.

[2] The definition of $L_n(\alpha, c)$ in [6] differs from that of [9] in the $o(1)$ term. We account for this discrepancy in our text.

Taking $\mu = \sqrt{2}z$ in [1, Prop. 6], we find that the (heuristic) expected running time of the loop in lines 4–9 of Algorithm 3 is $L(\frac{1}{2}, \frac{1}{4z})$.

The next step in Algorithm 3 having nontrivial running time is the computation of the ideal product in line 20. To exponentiate an element of an arbitrary semigroup to a power e requires $O(\log e)$ semigroup multiplication operations [10, §1.2]. To multiply two ideals I and J in an imaginary quadratic order (via composition of quadratic forms) requires $O(\max(\log(\mathrm{Norm}(I)), \log(\mathrm{Norm}(J)))^{1+\varepsilon})$ bit operations using fast multiplication [24, §6]. Each of the expressions $|I_i|^{|e_i|}$ therefore requires $O(\log |e_i|)$ ideal multiplication operations to compute, with each individual multiplication requiring

$$O((|e_i| \log(\mathrm{Norm}(I_i)))^{1+\varepsilon}) = O\left(\left(\left(\frac{N}{p_i}\right)^2 \log(p_i)\right)^{1+\varepsilon}\right) = O(N^{2+\varepsilon})$$

bit operations, for a total running time of $(\log e_i)O(N^{2+\varepsilon}) = L(\frac{1}{2}, 2z)$ for each i. This calculation must be performed once for each nonzero exponent e_i. By line 9, the number of nonzero exponents appearing in the relation is at most $\sqrt{\log(|\Delta|/3)}/z$, so the amount of time required to compute all of the $|I_i|^{|e_i|}$ for all i is $(\sqrt{\log(|\Delta|/3)}/z)L(\frac{1}{2}, 2z) = L(\frac{1}{2}, 2z)$. Afterward, the values $|I_i|^{|e_i|}$ must all be multiplied together, a calculation which entails at most $\sqrt{\log(|\Delta|/3)}/z$ ideal multiplications where the log-norms of the input multiplicands are bounded above by

$$\log \mathrm{Norm}(I_i^{|e_i|}) = |e_i| \log \mathrm{Norm}(I_i) \leq \left(\frac{N}{p_i}\right)^2 \log p_i \leq N^2 = L(\tfrac{1}{2}, 2z),$$

and thus each of the (at most) $\sqrt{\log(|\Delta|/3)}/z$ multiplications in the ensuing product can be completed in time at most $(\sqrt{\log(|\Delta|/3)}/z)L(\frac{1}{2}, 2z) = L(\frac{1}{2}, 2z)$. Finally, we must multiply this end result by $\mathfrak{L}$, an operation which requires $O(\max(\log \ell, L(\frac{1}{2}, 2z))^{1+\varepsilon})$ time. All together, the running time of step 20 is $L(\frac{1}{2}, 2z) + O(\max(\log \ell, L(\frac{1}{2}, 2z))^{1+\varepsilon}) = \max((\log \ell)^{1+\varepsilon}, L(\frac{1}{2}, 2z))$, and the norm of the resulting ideal I is bounded above by $\ell \cdot \exp(L(\frac{1}{2}, 2z))$.

Obtaining the generator β of I in line 21 of Algorithm 3 using Cornacchia's algorithm requires

$$O(\log(\mathrm{Norm}(I))^{1+\varepsilon}) = (\log \ell + L(\tfrac{1}{2}, 2z))^{1+\varepsilon}$$

time. We remark that finding β given I is substantially easier than the usual Cornacchia's algorithm, which entails finding β given only $\mathrm{Norm}(I)$. The usual algorithm requires finding *all* the square roots of Δ modulo $\mathrm{Norm}(I)$, which is very slow when $\mathrm{Norm}(I)$ has a large number of prime divisors. This time-consuming step is unnecessary when the ideal I itself is given, since the embedding of the ideal I in $\mathrm{End}(E)$ already provides (up to sign) the correct square root of Δ mod I. A detailed description of this portion of Cornacchia's algorithm in the context of the full algorithm, together with running time figures specific to each

sub-step, is given by Hardy et al. [19]; for our purposes, the running time of a single iteration of Step 6 in [19, §4] is the relevant figure. This concludes our analysis of Algorithm 3.

Returning to Algorithm 4, we find that (as in [4]) the computation of the individual isogenies ϕ_i in line 3 of Algorithm 4 is limited by the time required to compute the modular polynomials $\Phi_n(x, y)$. Using the Chinese remainder theorem-based method of Bröker et al. [5], these polynomials can be computed mod q in time $O(n^3 \log^{3+\varepsilon}(n))$, and the resulting polynomials require $O(n^2(\log^2 n + \log q))$ space. For each ideal I_i, the corresponding modular polynomial of level p_i only needs to be computed once, but the polynomial once computed must be evaluated, differentiated, and otherwise manipulated e_i times, at a cost of $O(p_i^{2+\varepsilon})$ field operations in $\mathbb{F}_q$ per manipulation, or $O(p_i^{2+\varepsilon})(\log q)^{1+\varepsilon}$ bit operations using fast multiplication. The total running time of line 3 is therefore

$$O(p_i^{3+\varepsilon}) + \sum_i |e_i| p_i^{2+\varepsilon} (\log q)^{1+\varepsilon} \leq O(N^{3+\varepsilon}) + \sum_i \left(\left(\frac{N}{p_i} \right)^2 \right) p_i^{2+\varepsilon} (\log q)^{1+\varepsilon}$$

$$\leq O(N^{3+\varepsilon}) + \frac{\sqrt{\log(|\Delta|/3)}}{z} N^{2+\varepsilon} (\log q)^{1+\varepsilon} = L(\tfrac{1}{2}, 3z) + L(\tfrac{1}{2}, 2z)(\log q)^{1+\varepsilon}.$$

Similarly, the evaluation of ϕ_c in line 4 requires

$$\sum_i |e_i| p_i^{2+\varepsilon} = L(\tfrac{1}{2}, 2z)$$

field operations in $\mathbb{F}_{q^n}$, which corresponds to $L(\frac{1}{2}, 2z)(\log q^n)^{1+\varepsilon}$ bit operations using fast multiplication.

Combining all the above quantities, we obtain a total running time of

$$\begin{aligned}
& L(\tfrac{1}{2}, z) && \text{(algorithm 2)} \\
& \quad + L(\tfrac{1}{2}, \tfrac{1}{4z}) && \text{(lines 4–9, algorithm 3)} \\
& \quad + \max((\log \ell)^{1+\varepsilon}, L(\tfrac{1}{2}, 2z)) && \text{(line 20, algorithm 3)} \\
& \quad + (\log \ell + L(\tfrac{1}{2}, 2z))^{1+\varepsilon} && \text{(line 21, algorithm 3)} \\
& \quad + L(\tfrac{1}{2}, 3z) + L(\tfrac{1}{2}, 2z)(\log q)^{1+\varepsilon} && \text{(line 3, algorithm 4)} \\
& \quad + L(\tfrac{1}{2}, 2z)(\log q^n)^{1+\varepsilon} && \text{(line 4, algorithm 4)}
\end{aligned}$$

$$= L(\tfrac{1}{2}, \tfrac{1}{4z}) + (\log \ell + L(\tfrac{1}{2}, 2z))^{1+\varepsilon} + L(\tfrac{1}{2}, 3z) + L(\tfrac{1}{2}, 2z)(\log q^n)^{1+\varepsilon}.$$

When $|\Delta|$ is large, we may impose the reasonable assumption that $\log(\ell) \ll L(\frac{1}{2}, z)$ and $\log(q^n) \ll L(\frac{1}{2}, z)$. In this case, the running time of Algorithm 4 is dominated by the expression $L(\frac{1}{2}, \frac{1}{4z}) + L(\frac{1}{2}, 3z)$, which attains a minimum at $z = \frac{1}{2\sqrt{3}}$. Taking this value of z, we find that the running time of Algorithm 4 is equal to $L_{|\Delta|}(\frac{1}{2}, \frac{\sqrt{3}}{2})$. Since the maximum value of $|\Delta| \leq |\Delta_\pi| = 4q - t^2$ is $4q$, we can alternatively express this running time as simply $L_q(\frac{1}{2}, \frac{\sqrt{3}}{2})$.

In the general case, $\log(\ell)$ and $\log(q^n)$ might be non-negligible compared to $L(\frac{1}{2}, z)$. This can happen in one of two ways: either $|\Delta|$ is small, or (less likely) ℓ is very large and/or n is large. When this happens, we can still bound the running time of Algorithm 4 by taking $z = \frac{1}{2\sqrt{3}}$ in the foregoing calculation, although such a choice may fail to be optimal. We then find that the running time of Algorithm 4 is bounded above by

$$(\log(\ell) + L(\tfrac{1}{2}, \tfrac{1}{\sqrt{3}}))^{1+\varepsilon} + L(\tfrac{1}{2}, \tfrac{\sqrt{3}}{2}) + L(\tfrac{1}{2}, \tfrac{1}{\sqrt{3}})(\log q^n)^{1+\varepsilon}.$$

We summarize our results in the following theorem.

Theorem 4.1. *Let $E/\mathbb{F}_q$ be an ordinary elliptic curve with Frobenius π_q, given by a Weierstrass equation, and let $P \in E(\mathbb{F}_{q^n})$ be a point on E. Let $\Delta = \mathrm{disc}(\mathrm{End}(E))$ be given. Assume that $[\mathrm{End}(E) : \mathbb{Z}[\pi_q]]$ and $\#E(\mathbb{F}_{q^n})$ are coprime, and let $\mathfrak{L} = (\ell, c + d\pi_q)$ be an $\mathrm{End}(E)$-ideal of prime norm $\ell \neq \mathrm{char}(\mathbb{F}_q)$ not dividing the index $[\mathrm{End}(E) : \mathbb{Z}[\pi_q]]$. Under the heuristics of [1, §4], Algorithm 4 computes the unique elliptic curve E' such that there exists a normalized isogeny $\phi\colon E \to E'$ with kernel $E[\mathfrak{L}]$. Furthermore, it computes the x-coordinate of $\phi(P)$ if $\mathrm{End}(E)$ does not equal $\mathbb{Z}[i]$ or $\mathbb{Z}[\zeta_3]$ and the square, respectively cube, of the x-coordinate of $\phi(P)$ otherwise. The running time of the algorithm is bounded above by*

$$(\log(\ell) + L(\tfrac{1}{2}, \tfrac{1}{\sqrt{3}}))^{1+\varepsilon} + L(\tfrac{1}{2}, \tfrac{\sqrt{3}}{2}) + L(\tfrac{1}{2}, \tfrac{1}{\sqrt{3}})(\log q^n)^{1+\varepsilon}.$$

The running time of the algorithm is subexponential in $\log|\Delta|$, and polynomial in $\log(\ell)$, $\log(q)$, and n.

5 Examples

5.1 Small Example

Let $p = 10^{10}+19$ and let $E/\mathbb{F}_p$ be the curve $y^2 = x^3+15x+129$. Then $E(\mathbb{F}_p)$ has cardinality $10000036491 = 3 \cdot 3333345497$ and trace $t = -36471$. To avoid any bias in the selection of the prime ℓ, we set ℓ to be the smallest Elkies prime of E larger than $p/2$, namely $\ell = 5000000029$. We will evaluate the x-coordinate of $\phi(P)$, where ϕ is an isogeny of degree ℓ, and P is chosen arbitrarily to be the point $(5940782169, 2162385016) \in E(\mathbb{F}_p)$. We remark that, although this example is designed to be artificially small for illustration purposes, the evaluation of this isogeny would already be infeasible if we were using prior techniques based on modular functions of level ℓ.

The discriminant Δ of E is $\Delta = t^2 - 4p = -38669866235$. Set $w = \frac{1+\sqrt{\Delta}}{2}$ and $\mathcal{O} = \mathcal{O}_\Delta$. The quadratic form $(5000000029, -2326859861, 270713841)$ represents a prime ideal $\mathfrak{L}$ of norm ℓ, and we show how to calculate the isogeny ϕ having kernel corresponding to $E[\mathfrak{L}]$. Using an implementation of Algorithm 3 in MAGMA [22], we find immediately the relation $\mathfrak{L} = (\frac{\beta}{m}) \cdot \mathfrak{p}_{19} \cdot \mathfrak{p}_{31}^{24}$

where $\beta = 588048307603210005w - 235788727470005542279904$, $m = 19 \cdot 31^{24}$, $\mathfrak{p}_{19} = (19, 2w+7)$, and $\mathfrak{p}_{31} = (31, 2w+5)$. Using this factorization, we can then evaluate $\phi\colon E \to E'$ using the latter portion of Algorithm 4. We find that E' is the curve with Weierstrass equation $y^2 = x^3 + 3565469415x + 7170659769$, and $\phi(P) = (7889337683, \pm 3662693258)$. We omit the details of these steps, since this portion of the algorithm is identical to the algorithm of Bröker, Charles and Lauter, and the necessary steps are already extensively detailed in their article [4].

We can check our computations for consistency by performing a second computation, starting from the curve $E' : y^2 = x^3 + 3565469415x + 7170659769$, the point $P' = (7889337683, 3662693258) \in E'(\mathbb{F}_p)$, and the conjugate ideal $\bar{\mathfrak{L}}$, which is represented by the quadratic form $(5000000029, 2326859861, 270713841)$. Let $\bar{\phi}\colon E' \to E''$ denote the unique normalized isogeny with kernel $E'[\bar{\mathfrak{L}}]$. Up to a normalization isomorphism $\iota\colon E \to E''$, the isogeny $\bar{\phi}$ should equal the dual isogeny $\hat{\phi}$ of ϕ, and the composition $\bar{\phi}(\phi(P))$ should yield $\iota(\ell P)$. Indeed, upon performing the computation, we find that E'' has equation

$$y^2 = x^3 + (15/\ell^4)x + (129/\ell^6),$$

which is isomorphic to E via the isomorphism $\iota\colon E \to E''$ defined by $\iota(x, y) = (x/\ell^2, y/\ell^3)$, and

$$\bar{\phi}(\phi(P)) = (3163843645, 8210361642) = (5551543736/\ell^2, 6305164567/\ell^3),$$

in agreement with the value of ℓP, which is $(5551543736, 6305164567)$.

5.2 Medium Example

Let E be the ECCp-109 curve [8] from the Certicom ECC Challenge [7], with equation $y^2 = x^3 + ax + b$ over $\mathbb{F}_p$ where

$$p = 564538252084441556247016902735257$$
$$a = 321094768129147601892514872825668$$
$$b = 430782315140218274262276694323197$$

As before, to avoid any bias in the choice of ℓ, we set ℓ to be the least Elkies prime greater than $p/2$, and we define $w = \frac{1+\sqrt{\Delta}}{2}$ where $\Delta = \mathrm{disc}(\mathrm{End}(E))$. Let $\mathfrak{L}$ be the prime ideal of norm ℓ in $\mathrm{End}(E)$ corresponding to the reduced quadratic form (ℓ, b, c) of discriminant Δ, where $b = -105137660734123120905310489472471$. For each Elkies prime p, let $\mathfrak{p}_p$ denote the unique prime ideal corresponding to the reduced quadratic form (p, b, c) where $b \geq 0$. Our smoothness bound in this case is $N = L(\frac{1}{2}, \frac{1}{2\sqrt{3}}) \approx 200$. Using Sutherland's `smoothrelation` package [28], which implements the FindRelation algorithm of [1], one finds in a few seconds (using an initial seed of 0) the relation $\mathfrak{L} = \left(\frac{\beta}{m}\right)\mathfrak{I}$, where

$$\mathfrak{I} = \bar{\mathfrak{p}}_7^{72}\bar{\mathfrak{p}}_{13}^{100}\bar{\mathfrak{p}}_{23}^{14}\bar{\mathfrak{p}}_{47}^{2}\bar{\mathfrak{p}}_{73}^{2}\bar{\mathfrak{p}}_{103}\mathfrak{p}_{179}\mathfrak{p}_{191}$$
$$m = 7^{72}13^{100}23^{14}47^{2}73^{2}103^{1}179^{1}191^{1}$$

and

$$\beta = 338394760102012126781530993189189355567744037461413704749298715122260417314622648471444260197118494483544222058008848371713152334033312180094376774440754045496152167352278262491589014097167238827239427644476075704890979685 \cdot w$$

We find that the codomain E' of the normalized isogeny $\phi\colon E \to E'$ of kernel $E[\mathfrak{L}]$ has equation $y^2 = x^3 + a'x + b'$ where

$$a' = 840812629621647700320334943079 76$$
$$b' = 506928585427238387307510041944828$$

and that the base point

$$P = (973390109870590665231561339089 35, 14967037284616928576068237197 8898)$$

of E given in the Certicom ECC challenge has image

$$(450689656718652268803536868496211, \pm 345608697871189839292674734567941).$$

under ϕ. As with the first example, we checked the computation for consistency by using the conjugate ideal.

5.3 Large Example

Let E be the ECCp-239 curve [8] from the Certicom ECC Challenge [7]. Then E has equation $y^2 = x^3 + ax + b$ over $\mathbb{F}_p$ where

$$p = 862591559561497151050143615844796924047865589835498401307522524859467869$$
$$a = 820125117492400602839381236756362453725976037283079104527317913759073622$$
$$b = 545482459632327583111433582031095022426858572446976004219654298705912499$$

Let $\mathfrak{L}$ be the prime ideal whose norm is the least Elkies prime greater than $p/2$ and whose ideal class is represented by the quadratic form (ℓ, b, c) with $b \geq 0$. We have $N = L(\frac{1}{2}, \frac{1}{2\sqrt{3}}) \approx 5000$, and one finds in a few hours using `smoothrelation` [28] that $\mathfrak{L}$ is equivalent to

$$\mathfrak{I} = \bar{\mathfrak{p}}_7^2 \mathfrak{p}_{11} \mathfrak{p}_{19} \mathfrak{p}_{37}^2 \bar{\mathfrak{p}}_{71}^2 \bar{\mathfrak{p}}_{131} \mathfrak{p}_{211} \bar{\mathfrak{p}}_{389} \bar{\mathfrak{p}}_{433} \bar{\mathfrak{p}}_{467} \bar{\mathfrak{p}}_{859}^{18} \mathfrak{p}_{863} \bar{\mathfrak{p}}_{1019} \bar{\mathfrak{p}}_{1151} \bar{\mathfrak{p}}_{1597} \bar{\mathfrak{p}}_{2143}^6 \bar{\mathfrak{p}}_{2207}^5 \bar{\mathfrak{p}}_{3359}$$

where each ideal $\mathfrak{p}_p$ is represented by the reduced quadratic form (p, b, c) having $b \geq 0$ (this computation can be reconstructed with [28] using the seed 7). The quotient $\mathfrak{L}/\mathfrak{I}$ is generated by β/m where $m = \mathrm{Norm}(\mathfrak{I})$ and β is

$$-92352598680305965222540607026543911791348859237474142895912091406705330 7\backslash 4585317 - 91755276862381815669553474208435929343264618996293547812922790 9w.$$

Given this relation, evaluating isogenies of degree ℓ is a tedious but routine computation using Elkies-Atkin techniques [4, §3.1]. Although we do not complete it here, the computation is well within the reach of present technology; indeed, Bröker et al. [5] have computed classical modular polynomials mod p of level up to 20000, well beyond the largest prime of 3389 appearing in our relation.

6 Related Work

Bisson and Sutherland [1] have developed an algorithm to compute the endomorphism ring of an elliptic curve in subexponential time, using relation-finding techniques which largely overlap with ours. Although our main results were obtained independently, we have incorporated their ideas into our algorithm in several places, resulting in a simpler presentation as well as a large speedup compared to the original version of our work.

Given two elliptic curves E and E' over $\mathbb{F}_q$ admitting a normalized isogeny $\phi\colon E \to E'$ of degree ℓ, the equation of ϕ as a rational function contains $O(\ell)$ coefficients. Bostan et al. [3] have published an algorithm which produces this equation, given E, E', and ℓ. Their algorithm has running time $O(\ell^{1+\varepsilon})$, which is quasi-optimal given the size of the output. Using our algorithm, it is possible to compute E' from E and ℓ in time $\log(\ell)L_{|\Delta|}(\frac{1}{2}, \frac{\sqrt{3}}{2})$ for large ℓ. Hence the combination of the two algorithms can produce the equation of ϕ within a quasi-optimal running time of $O(\ell^{1+\varepsilon})$, given only E and ℓ (or E and $\mathfrak{L}$), without the need to provide E' in the input.

Acknowledgments

We thank the anonymous referees for numerous suggestions which led to substantial improvements in our main result.

References

1. Bisson, G., Sutherland, A.: Computing the endomorphism ring of an ordinary elliptic curve over a finite field. Journal of Number Theory (to appear 2009)
2. Blake, I.F., Seroussi, G., Smart, N.P.: Elliptic curves in cryptography. London Mathematical Society Lecture Note Series, vol. 265. Cambridge University Press, Cambridge (2000); Reprint of the 1999 original (1999)
3. Bostan, A., Morain, F., Salvy, B., Schost, É.: Fast algorithms for computing isogenies between elliptic curves. Math. Comp. 77(263), 1755–1778 (2008)
4. Bröker, R., Charles, D., Lauter, K.: Evaluating large degree isogenies and applications to pairing based cryptography. In: Galbraith, S.D., Paterson, K.G. (eds.) Pairing 2008. LNCS, vol. 5209, pp. 100–112. Springer, Heidelberg (2008)
5. Bröker, R., Lauter, K., Sutherland, A.: Modular polynomials via isogeny volcanoes (2010)
6. Buchmann, J., Vollmer, U.: Binary quadratic forms. Algorithms and Computation in Mathematics, vol. 20. Springer, Berlin (2007); An algorithmic approach
7. Certicom ECC Challenge, http://www.certicom.com/images/pdfs/cert_ecc_challenge.pdf.

8. Certicom ECC Curves List, http://www.certicom.com/index.php/curves-list
9. Cohen, H., Frey, G., Avanzi, R., Doche, C., Lange, T., Nguyen, K., Vercauteren, F. (eds.): Handbook of elliptic and hyperelliptic curve cryptography. Discrete Mathematics and its Applications. Chapman & Hall/CRC (2006)
10. Cohen, H.: A course in computational algebraic number theory. Graduate Texts in Mathematics, vol. 138. Springer, Berlin (1993)
11. Couveignes, J.-M., Morain, F.: Schoof's algorithm and isogeny cycles. In: Huang, M.-D.A., Adleman, L.M. (eds.) ANTS 1994. LNCS, vol. 877, pp. 43–58. Springer, Heidelberg (1994)
12. Cox, D.A.: Primes of the form $x^2 + ny^2$. A Wiley-Interscience Publication, John Wiley & Sons Inc., New York (1989); Fermat, class field theory and complex multiplication
13. Enge, A.: Computing modular polynomials in quasi-linear time. Math. Comp. 78(267), 1809–1824 (2009)
14. Fouquet, M., Morain, F.: Isogeny volcanoes and the SEA algorithm. In: Fieker, C., Kohel, D.R. (eds.) ANTS 2002. LNCS, vol. 2369, pp. 276–291. Springer, Heidelberg (2002)
15. Freeman, D., Scott, M., Teske, E.: A taxonomy of pairing-friendly elliptic curves. J. Cryptology (to appear 2010)
16. Galbraith, S.D.: Constructing isogenies between elliptic curves over finite fields. LMS J. Comput. Math. 2, 118–138 (1999) (electronic)
17. Galbraith, S.D., Hess, F., Smart, N.P.: Extending the GHS Weil descent attack. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 29–44. Springer, Heidelberg (2002)
18. Hafner, J., McCurley, K.: A rigorous subexponential algorithm for computation of class groups. J. Amer. Math. Soc. 2(4), 837–850 (1989)
19. Hardy, K., Muskat, J.B., Williams, K.S.: A deterministic algorithm for solving $n = fu^2 + gv^2$ in coprime integers u and v. Math. Comp. 55(191), 327–343 (1990)
20. Jao, D., Miller, S.D., Venkatesan, R.: Do all elliptic curves of the same order have the same difficulty of discrete log? In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 21–40. Springer, Heidelberg (2005)
21. Kohel, D.: Endomorphism rings of elliptic curves over finite fields. PhD thesis, University of California, Berkeley (1996)
22. MAGMA Computational Algebra System, http://magma.maths.usyd.edu.au/
23. Menezes, A., Teske, E., Weng, A.: Weak fields for ECC. In: Okamoto, T. (ed.) CT-RSA 2004. LNCS, vol. 2964, pp. 366–386. Springer, Heidelberg (2004)
24. Schönhage, A.: Fast reduction and composition of binary quadratic forms. In: ISSAC 1991: Proceedings of the 1991 International Symposium on Symbolic and Algebraic Computation, pp. 128–133. ACM, New York (1991)
25. Schoof, R.: Counting points on elliptic curves over finite fields. J. Théor. Nombres Bordeaux 7(1), 219–254 (1995); Les Dix-huitièmes Journées Arithmétiques (Bordeaux, 1993)
26. Seysen, M.: A probabilistic factorization algorithm with quadratic forms of negative discriminant. Math. Comp. 48(178), 757–780 (1987)
27. Silverman, J.: The arithmetic of elliptic curves. Graduate Texts in Mathematics, vol. 106. Springer, New York (1992); Corrected reprint of the 1986 original (1986)
28. Sutherland, A.:Smoothrelation, http://math.mit.edu/~drew/smoothrelation_v1.tar
29. Tate, J.: Endomorphisms of abelian varieties over finite fields. Invent. Math. 2, 134–144 (1966)
30. Teske, E.: An elliptic curve trapdoor system. J. Cryptology 19(1), 115–133 (2006)

Huff's Model for Elliptic Curves

Marc Joye[1], Mehdi Tibouchi[2,⋆], and Damien Vergnaud[2]

[1] Technicolor, Security & Content Protection Labs
1 avenue de Belle Fontaine, 35576 Cesson-Sévigné Cedex, France
marc.joye@technicolor.com
[2] École Normale Supérieure – C.N.R.S. – I.N.R.I.A.
45, Rue d'Ulm – 75230 Paris CEDEX 05 – France
{mehdi.tibouchi,damien.vergnaud}@ens.fr

Abstract. This paper revisits a model for elliptic curves over $\mathbb{Q}$ introduced by Huff in 1948 to study a diophantine problem. Huff's model readily extends over fields of odd characteristic. Every elliptic curve over such a field and containing a copy of $\mathbb{Z}/4\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$ is birationally equivalent to a Huff curve over the original field.

This paper extends and generalizes Huff's model. It presents fast explicit formulæ for point addition and doubling on Huff curves. It also addresses the problem of the efficient evaluation of pairings over Huff curves. Remarkably, the so-obtained formulæ feature some useful properties, including completeness and independence of the curve parameters.

Keywords: Elliptic curves, Huff's model, unified addition law, complete addition law, explicit formulæ, scalar multiplication, Tate pairing, Miller's algorithm.

1 Introduction

Elliptic curves have been extensively studied in algebraic geometry and number theory since the middle of the nineteenth century. More recently, they have been used to devise efficient algorithms for factoring large integers [19,22] or for primality proving [2,13,23]. They also revealed useful in the construction of cryptosystems [18,20].

In this paper, we develop an elliptic curve model introduced by Huff in 1948 to study a diophantine problem. We present fast explicit formulæ for adding or doubling points on Huff curves. We also devise a couple of extensions and generalizations upon this model. We analyze the impact of these curves in cryptographic applications. Some of our addition formulæ are unified; *i.e.*, they remain valid for doubling a point. Even better, they achieve completeness (*i.e.*, are valid for all inputs) when restricted to a cyclic subgroup, as is customary in cryptographic settings. We also consider the problem of pairing computation over Huff curves.

⋆ This research was completed while the second author was visiting the Okamoto Research Laboratory at the NTT Information Sharing Platform (Tokyo, Japan).

G. Hanrot, F. Morain, and E. Thomé (Eds.): ANTS-IX 2010, LNCS 6197, pp. 234–250, 2010.

1.1 Background

Elliptic curves and cryptography. In 1985, Koblitz [18] and Miller [20] independently proposed the use of elliptic curves in public-key cryptography. The main advantage of elliptic curve systems stems from the absence of a subexponential-time algorithm to compute discrete logarithms on general elliptic curves over finite fields. Consequently, one can use an elliptic curve group that is smaller in size compared with systems based on either integer factorization or the discrete log problem in the multiplicative group of a finite field, while maintaining the same (heuristic) level of security (see [17] for a recent survey on elliptic curve cryptography).

The use of elliptic curves in cryptography makes the key sizes smaller but the arithmetic of the underlying group is more tedious (for example, with the widely-used Jacobian coordinates, the general addition of two points on an elliptic curve typically requires 16 field multiplications). Therefore a huge amount of research has been devoted to the analysis of the performance of various forms of elliptic curves proposed in the mathematical literature: Weierstraß cubics, Jacobi intersections, Hessian curves, Jacobi quartics, or the more recent forms of elliptic curves due to Montgomery, Doche-Icart-Kohel or Edwards (see [6] for an encyclopedic overview of these models). For instance, since 2007, there has been a rapid development of the curves introduced by Edwards in [12] and their use in cryptology. Bernstein and Lange proposed a more general version of these curves in [7] and the *inverted Edwards* coordinates in [8]. Bernstein, Birkner, Joye, Lange, and Peters studied *twisted Edwards* curves in [5]. Hisil, Wong, Carter and Dawson proposed *extended twisted Edwards* coordinates in [14]. Bernstein, Lange, and Farashahi covered the *binary case* in [9]. The first formulæ for computing *pairings* over Edwards curves were published by Das and Sarkar [11]. They were subsequently improved by Ionica and Joux [16]. The best implementation to date is due to Arène, Lange, Naehrig, and Ritzenhaler [1]. The present paper is aimed at providing a similar study for a forgotten model of elliptic curves hinted by Huff in 1948.

A diophantine problem. Huff [15] considered rational distance sets S (*i.e.*, subsets S of the plane $\mathbb{R}^2$ such that for all $s, t \in S$, the distance between s and t is a rational number) of the following form: given distinct $a, b \in \mathbb{Q}$, S contains the four points $(0, \pm a)$ and $(0, \pm b)$ on the y-axis, plus points $(x, 0)$ on the x-axis, for some $x \in \mathbb{Q}$. Such a point $(x, 0)$ must then satisfy the equations $x^2 + a^2 = u^2$ and $x^2 + b^2 = v^2$ with $u, v \in \mathbb{Q}$. The system of associated homogeneous equations $x^2 + a^2z^2 = u^2$ and $x^2 + b^2z^2 = v^2$ defines a curve of genus 1 in $\mathbb{P}^3$. Huff, and later his student Peeples [24], provided examples where this curve has positive rank over $\mathbb{Q}$, thus exhibiting examples of arbitrarily large rational distance sets of cardinality $k > 4$ such that exactly $k - 4$ points are on one line.

The above mentioned genus 1 curve is birationally equivalent to the curve

$$ax(y^2 - 1) = by(x^2 - 1) \tag{1}$$

for some parameters a and b in $\mathbb{Q}$. It is easily seen that, over any field $\mathbb{K}$ of odd characteristic, Equation (1) defines an elliptic curve if $a^2 \neq b^2$ and $a, b \neq 0$.

Indeed, if $ab \neq 0$, the gradient of the curve $F(X,Y,Z) = aX(Y^2-Z^2)-bY(X^2-Z^2)$ in the projective plane $\mathbb{P}^2(\mathbb{K})$ is

$$\left(\frac{\partial F}{\partial X}, \frac{\partial F}{\partial Y}, \frac{\partial F}{\partial Z}\right) = \left(a(Y^2-Z^2) - 2bXY, 2aXY - b(X^2-Z^2), 2(-aX+bY)Z\right),$$

which does not vanish at the three points at infinity $(1:0:0)$, $(0:1:0)$ and $(a:b:0)$ and vanishes at a finite point $(x:y:1)$ if and only if $ax = by$, which together with Eq. (1) implies that $x^2 = y^2$ and therefore $a^2 = b^2$. It is worth noting that in characteristic 2, the point $(1:1:1)$ is always singular and therefore the family of curves defined by (1) does not contain any smooth curve. As will be shown in Section 3, we can extend our study to even characteristic by considering a generalized model.

1.2 Contributions of the Paper

Our first contribution is a detailed study of Huff's form for elliptic curves over finite fields of odd characteristic and a statement of the addition law in these groups. We show in particular that all elliptic curves over non-binary finite fields with a subgroup isomorphic to $\mathbb{Z}/4\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$ can be transformed to Huff's form. We then analyze their arithmetic and investigate several generalizations and extensions. In particular, we present explicit formulæ (*i.e.*, as a series of field operations) that

- compute a *complete* addition $(X_1:Y_1:Z_1) \oplus (X_2:Y_2:Z_2)$ using $12\mathsf{m}$;
- compute a *unified* addition $(X_1:Y_1:Z_1) \oplus (X_2:Y_2:Z_2)$ using $11\mathsf{m}$;
- compute a mixed addition $(X_1:Y_1:Z_1) \oplus (X_2:Y_2:1)$ using $10\mathsf{m}$;
- compute a doubling $[2](X_1:Y_1:Z_1)$ using $6\mathsf{m}+5\mathsf{s}$

where m and s denote multiplications and squarings in the base field $\mathbb{K}$.

As a further contribution, since bilinear pairings have found numerous applications in cryptography, we also present formulæ for computing Tate pairings using Huff's form. Specifically, we present explicit formulæ that

- compute a *full* Miller addition using $1\mathsf{M}+(k+15)\mathsf{m}$;
- compute a *mixed* Miller addition using $1\mathsf{M}+(k+13)\mathsf{m}$;
- compute a Miller doubling using $1\mathsf{M}+1\mathsf{S}+(k+11)\mathsf{m}+6\mathsf{s}$

on a Huff curve over $\mathbb{K} = \mathbb{F}_q$ of embedding degree k. M and S denote multiplications and squarings in the larger field $\mathbb{F}_{q^k}$ while m and s are operations in $\mathbb{F}_q$ as before.

Outline. The rest of this paper is organized as follows. The next section introduces Huff's model. We develop efficient unified addition formulæ and discuss the applicability of the model. We explicit the class of elliptic curves covered by Huff's model. In Section 3, we present several generalizations and extensions. We offer dedicated addition formulæ. We generalize Huff's model to cover a larger class of elliptic curves. We also extend the model to the case of binary fields. Section 4 deals with pairings over Huff curves. We exploit the relative simplicity of the underlying group law to devise efficient formulæ for the evaluation of the Tate pairing. Finally, we conclude in Section 5.

2 Huff's Model

Let $\mathbb{K}$ denote a field of characteristic $\neq 2$. Consider the set of projective points $(X : Y : Z) \in \mathbb{P}^2(\mathbb{K})$ satisfying the equation

$$E_{/\mathbb{K}} : aX(Y^2 - Z^2) = bY(X^2 - Z^2) \tag{2}$$

where $a, b \in \mathbb{K}^\times$ and $a^2 \neq b^2$. This form is referred to as *Huff's model* of an elliptic curve.

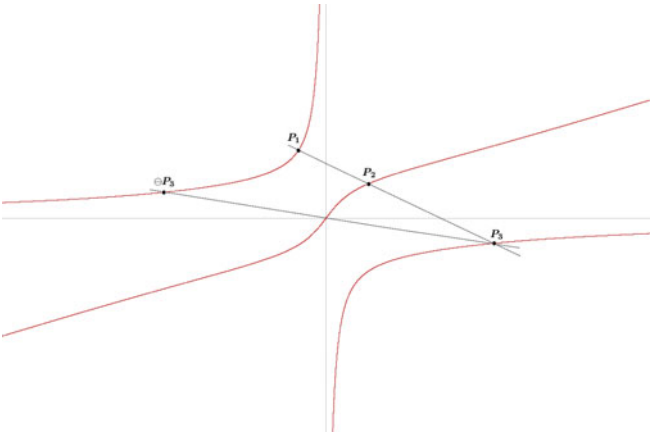


Fig. 1. Example of a Huff curve (over $\mathbb{R}$)

The tangent line at $(0 : 0 : 1)$ is $aX = bY$, which intersects the curve with multiplicity 3, so that $\boldsymbol{O} = (0 : 0 : 1)$ is an inflection point of E. $(E, \boldsymbol{O})$ is therefore an elliptic curve with $\boldsymbol{O}$ as neutral element and whose group law, denoted $\oplus$, has the following property: for any line intersecting the cubic curve E at the three points $\boldsymbol{P_1}$, $\boldsymbol{P_2}$ and $\boldsymbol{P_3}$ (counting multiplicities), we have $\boldsymbol{P_1} \oplus \boldsymbol{P_2} \oplus \boldsymbol{P_3} = \boldsymbol{O}$. In particular, the inverse of point $\boldsymbol{P_1} = (X_1 : Y_1 : Z_1)$ is $\ominus \boldsymbol{P_1} = (X_1 : Y_1 : -Z_1)$ and the sum of $\boldsymbol{P_1}$ and $\boldsymbol{P_2}$ is $\boldsymbol{P_1} \oplus \boldsymbol{P_2} = \ominus \boldsymbol{P_3}$. We note that a point at infinity is its own inverse. Hence, the three points at infinity (*i.e.*, on the line $Z = 0$ in $\mathbb{P}^2$) — namely, $(1 : 0 : 0)$, $(0 : 1 : 0)$ and $(a : b : 0)$, are exactly the three primitive 2-torsion points of E. The sum of any two of them is equal to the third one. More generally, $(X_1 : Y_1 : Z_1) \oplus (1 : 0 : 0)$ is the inverse of the point of intersection of the "horizontal" line passing through $(X_1 : Y_1 : Z_1)$ with E. When $Z_1 \neq 0$, we have

$$(X_1 : Y_1 : Z_1) \oplus (1 : 0 : 0) = ({Z_1}^2 : -X_1 Y_1 : X_1 Z_1)\,,$$

and analogously,

$$(X_1 : Y_1 : Z_1) \oplus (0 : 1 : 0) = (-X_1 Y_1 : {Z_1}^2 : Y_1 Z_1)\ .$$

From $(a:b:0) = (1:0:0) \oplus (0:1:0)$, when $Z_1 \neq 0$, we get $(X_1 : Y_1 : Z_1) + (a : b : 0) = ({Z_1}^2 : -X_1Y_1 : X_1Z_1) \oplus (0:1:0)$ and therefore

$$(X_1 : Y_1 : Z_1) \oplus (a : b : 0) = \begin{cases} (a:b:0) & \text{if } (X_1 : Y_1 : Z_1) = (0:0:1) \\ (Y_1Z_1 : X_1Z_1 : -X_1Y_1) & \text{otherwise} \end{cases} .$$

We remark that adding $(a:b:0)$ to any of the points $(\pm 1 : \pm 1 : 1)$ transforms it into its inverse. It follows that these four points are the four solutions to the equation $[2]\boldsymbol{P} = (a:b:0)$ and so are primitive 4-torsion points. The eight remarkable points we identified form a subgroup isomorphic to $\mathbb{Z}/4\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$. When $\mathbb{K} = \mathbb{Q}$, this must be the full torsion since, according to a theorem by Mazur, the torsion subgroup is of order at most 12 (and thus exactly 8 here).

Remark 1. In [15, p. 445], it is noted that the inverse projective transformations

$$\begin{aligned} &\Upsilon : \mathbb{P}^2(\mathbb{K}) \to \mathbb{P}^2(\mathbb{K}) : \\ &\quad (X:Y:Z) \mapsto (U:V:W) = \big(ab(bX - aY) : ab(b^2 - a^2)Z : -aX + bY\big) \end{aligned}$$

and

$$\begin{aligned} &\Upsilon^{-1} : \mathbb{P}^2(\mathbb{K}) \to \mathbb{P}^2(\mathbb{K}) : \\ &\qquad (U:V:W) \mapsto (X:Y:Z) = \big(b(U + a^2W) : a(U + b^2W) : V\big) \end{aligned}$$

induce a correspondence between Eq. (2) and the Weierstraß equation

$$V^2W = U(U + a^2W)(U + b^2W) \ .$$

Observe that point at infinity $(0:1:0)$ on the Weierstraß curve is mapped to $(0:0:1)$ on the Huff curve through Υ^{-1}. Observe also that map Υ^{-1} is a line-preserving transformation. This is another way to see that the group law on a Huff curve E follows the chord-and-tangent rule [25, § 2] with $\boldsymbol{O} = (0:0:1)$ as neutral element.

2.1 Affine Formulæ

We give explicit formulæ for the group law. Excluding the 2-torsion, we use the non-homogeneous form $ax(y^2 - 1) = by(x^2 - 1)$. Let $y = \lambda x + \mu$ denote the secant line passing through two different points $\boldsymbol{P_1} = (x_1, y_1)$ and $\boldsymbol{P_2} = (x_2, y_2)$. This line intersects the curve at a third point $\ominus \boldsymbol{P_3} = (-x_3, -y_3)$. Plugging the line equation into the curve equation, we get

$$ax\big((\lambda x + \mu)^2 - 1\big) = b(\lambda x + \mu)(x^2 - 1) \implies \lambda(a\lambda - b)x^3 + \mu(2a\lambda - b)x^2 + \cdots = 0 \ .$$

Whenever defined, we so obtain

$$\begin{cases} x_3 = x_1 + x_2 + \dfrac{\mu(2a\lambda - b)}{\lambda(a\lambda - b)} \\ y_3 = \lambda x_3 - \mu \end{cases}$$

with $\lambda = \dfrac{y_1 - y_2}{x_1 - x_2}$ and $\mu = y_1 - \lambda x_1$. After simplification, we have

$$\begin{aligned} x_3 &= x_1 + x_2 + \frac{(x_1y_2 - x_2y_1)\big(2a(y_1 - y_2) - b(x_1 - x_2)\big)}{(y_1 - y_2)\big(a(y_1 - y_2) - b(x_1 - x_2)\big)} \\ &= \frac{(x_1 - x_2)\big(a({y_1}^2 - {y_2}^2) - b(x_1y_1 - x_2y_2)\big)}{(y_1 - y_2)\big(a(y_1 - y_2) - b(x_1 - x_2)\big)} \end{aligned}$$

and

$$y_3 = -\frac{(y_1 - y_2)\big(b({x_1}^2 - {x_2}^2) - a(x_1y_1 - x_2y_2)\big)}{(x_1 - x_2)\big(a(y_1 - y_2) - b(x_1 - x_2)\big)} \ .$$

The above formulæ can be further simplified by reusing the curve equation. A simple calculation shows that

$$\big(a(y_1 - y_2) - b(x_1 - x_2)\big)(x_1 + x_2)y_1y_2 = a(x_2y_1 - x_1y_2)(y_1y_2 - 1) \ .$$

Hence, we can write

$$\begin{aligned} x_3 &= x_1 + x_2 - \frac{\big(2a(y_1 - y_2) - b(x_1 - x_2)\big)(x_1 + x_2)y_1y_2}{(y_1 - y_2)a(y_1y_2 - 1)} \\ &= x_1 + x_2 - \frac{x_2y_1 - x_1y_2}{y_1 - y_2} - \frac{(x_1 + x_2)y_1y_2}{y_1y_2 - 1} \\ &= \frac{x_1y_1 - x_2y_2}{y_1 - y_2} - \frac{(x_1 + x_2)y_1y_2}{y_1y_2 - 1} \ . \end{aligned}$$

Furthermore, as easily shown

$$b(x_1y_1 - x_2y_2)(x_1x_2 + 1) = (y_1 - y_2)\big(ax_1x_2(y_1 + y_2) + b(x_1 + x_2)\big) \, ,$$

it thus follows that

$$\begin{aligned} x_3 &= \frac{ax_1x_2(y_1 + y_2) + b(x_1 + x_2)}{b(x_1x_2 + 1)} - \frac{(x_1 + x_2)y_1y_2}{y_1y_2 - 1} \\ &= \frac{(x_1 + x_2)(1 + y_1y_2)}{(1 + x_1x_2)(1 - y_1y_2)} \ , \end{aligned} \tag{3}$$

since $ax_1x_2(y_1 + y_2)(1 - y_1y_2) = by_1y_2(x_1 + x_2)(1 - x_1x_2)$.

Likewise, by symmetry, we have

$$y_3 = \frac{(y_1 + y_2)(1 + x_1x_2)}{(1 - x_1x_2)(1 + y_1y_2)} \ . \tag{4}$$

Equations (3) and (4) are defined whenever $x_1x_2 \neq \pm 1$ and $y_1y_2 \neq \pm 1$. Advantageously, curve parameters are not involved. Moreover, this addition law is *unified*: it can be used to double a point (*i.e.*, when $\boldsymbol{P_2} = \boldsymbol{P_1}$).

2.2 Projective Formulæ

Previous affine formulæ involve inversions in $\mathbb{K}$. To avoid these operations and get faster arithmetic, projective coordinates may be preferred.

We let m and s represent the cost of a multiplication and of a squaring in $\mathbb{K}$, respectively. The projective form of Eqs (3) and (4) is

$$\begin{cases} X_3 = (X_1Z_2 + X_2Z_1)(Y_1Y_2 + Z_1Z_2)^2(Z_1Z_2 - X_1X_2) \\ Y_3 = (Y_1Z_2 + Y_2Z_1)(X_1X_2 + Z_1Z_2)^2(Z_1Z_2 - Y_1Y_2) \\ Z_3 = ({Z_1}^2{Z_2}^2 - {X_1}^2{X_2}^2)({Z_1}^2{Z_2}^2 - {Y_1}^2{Y_2}^2) \end{cases} . \tag{5}$$

In more detail, this can be evaluated as

$$\begin{gathered} m_1 = X_1X_2,\ \ m_2 = Y_1Y_2,\ \ m_3 = Z_1Z_2, \\ m_4 = (X_1 + Z_1)(X_2 + Z_2) - m_1 - m_3,\ \ m_5 = (Y_1 + Z_1)(Y_2 + Z_2) - m_2 - m_3, \\ m_6 = (m_2 + m_3)(m_3 - m_1),\ \ m_7 = (m_1 + m_3)(m_3 - m_2), \\ m_8 = m_4(m_2 + m_3),\ \ m_9 = m_5(m_1 + m_3), \\ X_3 = m_8m_6,\ \ Y_3 = m_9m_7,\ \ Z_3 = m_6m_7, \end{gathered}$$

that is, with $\underline{12\mathsf{m}}$.

2.3 Applicability

If $(x_1, y_1) \neq (0, 0)$ then $(x_1, y_1) \oplus (a : b : 0) = -(\frac{1}{x_1}, \frac{1}{y_1})$. Observe that Equation (5) remains valid for doubling point $(a : b : 0)$ or for adding point $(a : b : 0)$ to another finite point (*i.e.*, which is not at infinity) different from $\boldsymbol{O}$; we get $(X_1 : Y_1 : Z_1) \oplus (a : b : 0) = (-Y_1Z_1 : -X_1Z_1 : X_1Y_1)$ as expected. The addition formula is however not valid for adding $(0 : 1 : 0)$ or $(1 : 0 : 0)$. More generally, we have:

Theorem 1. *Let $\mathbb{K}$ be a field of characteristic $\neq 2$. Let $\boldsymbol{P_1} = (X_1 : Y_1 : Z_1)$ and $\boldsymbol{P_2} = (X_2 : Y_2 : Z_2)$ be two points on a Huff curve over $\mathbb{K}$. Then the addition formula given by Eq. (5) is valid provided that $X_1X_2 \neq \pm Z_1Z_2$ and $Y_1Y_2 \neq \pm Z_1Z_2$.*

Proof. If $\boldsymbol{P_1}$ and $\boldsymbol{P_2}$ are finite, we can write $\boldsymbol{P_1} = (x_1, y_1)$ and $\boldsymbol{P_2} = (x_2, y_2)$. The above affine formula for (x_3, y_3) as given by Eqs (3) and (4) is defined whenever $x_1x_2 \neq \pm 1$ and $y_1y_2 \neq \pm 1$. This translates into $X_1X_2 \neq \pm Z_1Z_2$ and $Y_1Y_2 \neq \pm Z_1Z_2$ for their projective coordinates.

It remains to analyze points at infinity. The points with their Z-coordinate equal to 0 are $(1 : 0 : 0)$, $(0 : 1 : 0)$ and $(a : b : 0)$. If $\boldsymbol{P_1}$ or $\boldsymbol{P_2} \in \{(1 : 0 : 0), (0 : 1 : 0)\}$, the condition $X_1X_2 \neq \pm Z_1Z_2$ and $Y_1Y_2 \neq \pm Z_1Z_2$ is not satisfied. Suppose now $\boldsymbol{P_2} = (a : b : 0)$. The condition becomes $X_1 \neq 0$ and $Y_1 \neq 0$, which corresponds to $\boldsymbol{P_1} \notin \{\boldsymbol{O}, (1 : 0 : 0), (0 : 1 : 0)\}$. As aforementioned, the addition law is then valid for adding $\boldsymbol{P_1}$ to $(a : b : 0)$. □

The previous theorem says that the addition on a Huff curve is almost complete. However, the exceptional inputs are easily prevented in practice. Cryptographic applications typically involve (large) prime-order subgroups. More specifically, we state:

Corollary 1. *Let E be a Huff curve over a field $\mathbb{K}$ of odd characteristic. Let also $\boldsymbol{P} \in E(\mathbb{K})$ be a point of odd order. Then the addition law in the subgroup generated by $\boldsymbol{P}$ is complete.*

Proof. All points in $\langle \boldsymbol{P} \rangle$ are of odd order and thus are finite (remember that points at infinity are of order 2). It remains to show that for any points $\boldsymbol{P_1} = (x_1, y_1), \boldsymbol{P_2} = (x_2, y_2) \in \langle \boldsymbol{P} \rangle$, we have $x_1 x_2 \neq \pm 1$ and $y_1 y_2 \neq \pm 1$. Note that $x_1, y_1, x_2, y_2 \neq \pm 1$ since this corresponds to points of order 4 (and thus not in $\langle \boldsymbol{P} \rangle$). Suppose that $x_1 x_2 = \pm 1$. Then $a x_1 ({y_1}^2 - 1) = b y_1 ({x_1}^2 - 1) \implies a \frac{1}{x_1} ({y_1}^2 - 1) = b y_1 (1 - \frac{1}{{x_1}^2}) \implies \pm a x_2 ({y_1}^2 - 1) = -b y_1 ({x_2}^2 - 1)$. Hence, since $a x_2 ({y_2}^2 - 1) = b y_2 ({x_2}^2 - 1)$, it follows that $\mp y_2 ({y_1}^2 - 1) = y_1 ({y_2}^2 - 1) \implies (y_1 \pm y_2)(1 \mp y_1 y_2) = 0 \implies y_2 = \mp y_1$ or $y_1 y_2 = \pm 1$. As a result, when $x_1 x_2 = \pm 1$, we have $(x_2, y_2) \in \left\{ (\frac{1}{x_1}, -y_1), (\frac{1}{x_1}, \frac{1}{y_1}), (-\frac{1}{x_1}, y_1), (-\frac{1}{x_1}, -\frac{1}{y_1}) \right\}$. In all cases, one of $(x_1, y_1) \oplus (x_2, y_2)$ or $(x_1, y_1) \ominus (x_2, y_2)$ is a 2-torsion point, a contradiction. Likewise, it can be verified that the case $y_1 y_2 = \pm 1$ leads to a contradiction, which concludes the proof. □

The *completeness* of the addition law is very useful as it yields a natural protection against certain side-channel attacks (*e.g.*, see [10]). Another useful feature is that the addition law is *independent* of the curve parameters.

2.4 Universality of the Model

The next theorem states that every elliptic curve over a field of characteristic $\neq 2$ containing a copy of $\mathbb{Z}/4\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$ can be put in Huff's form. Generalizations and extensions are discussed in Section 3.

Theorem 2. *Any elliptic curve $(E, \boldsymbol{O})$ over a perfect field $\mathbb{K}$ of characteristic $\neq 2$ such that $E(\mathbb{K})$ contains a subgroup $\mathbb{G}$ isomorphic to $\mathbb{Z}/4\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$ is birationally equivalent over $\mathbb{K}$ to a Huff curve.*

Proof. The Riemann-Roch theorem implies that if $\boldsymbol{D} = a_1 \boldsymbol{P_1} + \cdots + a_r \boldsymbol{P_r}$ is a divisor of degree 0 on E then the dimension of the vector space

$$\mathscr{L}(\boldsymbol{D}) = \{ f \in \mathbb{K}(E)^\times \mid \mathrm{div}(f) \geqslant -\boldsymbol{D} \} \cup \{0\}$$

is equal to 1 when $a_1 \boldsymbol{P_1} \oplus \cdots \oplus a_r \boldsymbol{P_r} = \boldsymbol{O}$, and to 0 otherwise.

Let $\boldsymbol{H_{++}}, \boldsymbol{H_{+-}}, \boldsymbol{H_{-+}}$ and $\boldsymbol{H_{--}}$ denote the four points of $\mathbb{G}$ of order exactly 4 (with the convention $\boldsymbol{H_{++}} \oplus \boldsymbol{H_{--}} = \boldsymbol{O}$). Doubling these points produces a unique primitive 2-torsion point that we denote $\boldsymbol{R}$. We further let $\boldsymbol{P}$ and $\boldsymbol{Q}$ denote the other two 2-torsion points; say, $\boldsymbol{P} = \ominus \boldsymbol{H_{++}} \oplus \boldsymbol{H_{+-}}$ and $\boldsymbol{Q} = \boldsymbol{H_{++}} \oplus \boldsymbol{H_{+-}}$. We have $\boldsymbol{P} \oplus \boldsymbol{R} \ominus \boldsymbol{Q} \ominus \boldsymbol{O} = \boldsymbol{O}$; so there exists a nonzero

rational function x with divisor exactly $\boldsymbol{Q} + \boldsymbol{O} - \boldsymbol{P} - \boldsymbol{R}$. In particular, x is well-defined and nonzero at $\boldsymbol{H}_{++}$ and thus without loss of generality we may assume that $x(\boldsymbol{H}_{++}) = 1$. Similarly, there exists a rational function y with divisor $\boldsymbol{P} + \boldsymbol{O} - \boldsymbol{Q} - \boldsymbol{R}$ such that $y(\boldsymbol{H}_{++}) = 1$.

The rational function $x - 1$ has the same poles as x and vanishes at $\boldsymbol{H}_{++}$. Its divisor $\mathrm{div}(x-1)$ is thus given by $\boldsymbol{H}_{++} + \boldsymbol{X} - \boldsymbol{P} - \boldsymbol{R}$ for some point $\boldsymbol{X}$. Since this divisor is principal, we have $\boldsymbol{H}_{++} \oplus \boldsymbol{X} \ominus \boldsymbol{P} \ominus \boldsymbol{R} = \boldsymbol{O}$. Hence, it follows that $\boldsymbol{X} = \boldsymbol{P} \oplus \boldsymbol{R} \ominus \boldsymbol{H}_{++} = \ominus\boldsymbol{H}_{++} \oplus \boldsymbol{H}_{+-} \oplus \boldsymbol{R} \ominus \boldsymbol{H}_{++} = \boldsymbol{H}_{+-}$. Consequently, we have $x(\boldsymbol{H}_{+-}) = 1$. Likewise, it is verified that $y(\boldsymbol{H}_{-+}) = 1$.

Now, consider the map ι taking a rational function f to $\iota f : \boldsymbol{M} \mapsto f(\ominus\boldsymbol{M})$. This is an endomorphism of the vector space $\mathscr{L}(\boldsymbol{P} + \boldsymbol{R} - \boldsymbol{Q} - \boldsymbol{O})$. Indeed, the poles of ιf are $\ominus\boldsymbol{P} = \boldsymbol{P}$ and $\ominus\boldsymbol{R} = \boldsymbol{R}$ and its zeros are $\ominus\boldsymbol{Q} = \boldsymbol{Q}$ and $\ominus\boldsymbol{O} = \boldsymbol{O}$. Moreover, since $\iota^2 = \mathrm{id}$ and since $\mathscr{L}(\boldsymbol{P} + \boldsymbol{R} - \boldsymbol{Q} - \boldsymbol{O})$ is a one-dimensional vector space, ι is the multiplication map by 1 or -1. The equality $\iota x = x$ would imply $x(\boldsymbol{H}_{--}) = x(\boldsymbol{H}_{++}) = 1$, which contradicts the previous calculation of $\mathrm{div}(x-1)$. As a result, we must have $\iota x = -x$. In particular, noting that $\boldsymbol{H}_{-+} = \ominus\boldsymbol{H}_{+-}$, we obtain

$$x(\boldsymbol{H}_{-+}) = \iota x(\boldsymbol{H}_{+-}) = -x(\boldsymbol{H}_{+-}) = -1\,,$$

and similarly for $\boldsymbol{H}_{--}$. Since $x+1$ has the same poles as x, its divisor is then given by $\mathrm{div}(x+1) = \boldsymbol{H}_{-+} + \boldsymbol{H}_{--} - \boldsymbol{P} - \boldsymbol{R}$. Analogously, we obtain $\mathrm{div}(y+1) = \boldsymbol{H}_{+-} + \boldsymbol{H}_{--} - \boldsymbol{Q} - \boldsymbol{R}$.

Finally, consider the rational functions $u = x(y^2-1)$ and $v = y(x^2-1)$. We have:

$$\begin{aligned}
\mathrm{div}(u) &= \mathrm{div}(x) + \mathrm{div}(y-1) + \mathrm{div}(y+1) \\
&= (\boldsymbol{Q} + \boldsymbol{O} - \boldsymbol{P} - \boldsymbol{R}) + (\boldsymbol{H}_{++} + \boldsymbol{H}_{-+} - \boldsymbol{Q} - \boldsymbol{R}) + \\
&\qquad\qquad (\boldsymbol{H}_{+-} + \boldsymbol{H}_{--} - \boldsymbol{Q} - \boldsymbol{R}) \\
&= \boldsymbol{H}_{++} + \boldsymbol{H}_{+-} + \boldsymbol{H}_{-+} + \boldsymbol{H}_{--} + \boldsymbol{O} - \boldsymbol{P} - \boldsymbol{Q} - 3\boldsymbol{R}
\end{aligned}$$

and

$$\begin{aligned}
\mathrm{div}(v) &= \mathrm{div}(y) + \mathrm{div}(x-1) + \mathrm{div}(x+1) \\
&= (\boldsymbol{P} + \boldsymbol{O} - \boldsymbol{Q} - \boldsymbol{R}) + (\boldsymbol{H}_{++} + \boldsymbol{H}_{+-} - \boldsymbol{P} - \boldsymbol{R}) + \\
&\qquad\qquad (\boldsymbol{H}_{-+} + \boldsymbol{H}_{--} - \boldsymbol{P} - \boldsymbol{R}) \\
&= \boldsymbol{H}_{++} + \boldsymbol{H}_{+-} + \boldsymbol{H}_{-+} + \boldsymbol{H}_{--} + \boldsymbol{O} - \boldsymbol{P} - \boldsymbol{Q} - 3\boldsymbol{R}\ .
\end{aligned}$$

But the vector space $\mathscr{L}(\boldsymbol{P} + \boldsymbol{Q} + 3\boldsymbol{R} - \boldsymbol{O} - \boldsymbol{H}_{++} - \boldsymbol{H}_{+-} - \boldsymbol{H}_{-+} - \boldsymbol{H}_{--})$ is of dimension 1, so there exists a linear relation between u and v. In other words, there exist $a, b \in \mathbb{K}^\times$ such that $au = bv$; *i.e.*, such that $ax(y^2-1) = by(x^2-1)$.

The rational map $E \to \mathbb{P}^2(\mathbb{K})$ given by $\boldsymbol{M} \mapsto (x(\boldsymbol{M}) : y(\boldsymbol{M}) : 1)$ extends to a morphism defined on all of E, and its image is contained in $E_{a,b}$ in view of the previous relation (and $E_{a,b}$ itself is a smooth irreducible curve as seen in §1.1). We therefore have a non-constant — and hence surjective — morphism of curves $E \to E_{a,b}$. Moreover, its degree is at most 1: indeed, if a point $(x_0 : y_0 : 1) \in E_{a,b}(\overline{\mathbb{K}})$ has two distinct pre-images $\boldsymbol{M} \neq \boldsymbol{M'} \in E(\overline{\mathbb{K}})$, the functions $x - x_0$ and

$y - y_0$ vanish at $\boldsymbol{M}$ and $\boldsymbol{M'}$. Since they have the same poles as x and y, their divisors are respectively $\boldsymbol{M} + \boldsymbol{M'} - \boldsymbol{P} - \boldsymbol{R}$ and $\boldsymbol{M} + \boldsymbol{M'} - \boldsymbol{Q} - \boldsymbol{R}$, which yields $\boldsymbol{P} \oplus \boldsymbol{R} = \boldsymbol{M} \oplus \boldsymbol{M'} = \boldsymbol{Q} \oplus \boldsymbol{R}$, a contradiction. As a surjective morphism of degree 1, the map $E \to E_{a,b}$ is thus an isomorphism. □

3 Generalizations and Extensions

This section presents dedicated addition formulæ. It also presents a generalization of the model as originally introduced by Huff so that it covers more curves and extends to binary fields.

3.1 Faster Computations

Dedicated doubling. The doubling formula can be sped up by evaluating squarings in $\mathbb{K}$ with a specialized implementation. The cost of a point doubling then becomes $\underline{7\mathsf{m} + 5\mathsf{s}}$. When $\mathsf{s} > \frac{3}{4}\mathsf{m}$, an even faster way for doubling a point is given by

$$\begin{gathered}
m_1 = X_1 Y_1,\ \ m_2 = X_1 Z_1,\ \ m_3 = Y_1 Z_1,\ \ s_1 = {Z_1}^2, \\
m_4 = (m_2 - m_3)(m_2 + m_3),\ \ m_5 = (m_1 - s_1)(m_1 + s_1), \\
m_6 = (m_1 - s_1)(m_2 - m_3),\ \ m_7 = (m_1 + s_1)(m_2 + m_3), \\
X([2]\boldsymbol{P_1}) = (m_6 - m_7)(m_4 + m_5),\ \ Y([2]\boldsymbol{P_1}) = (m_6 + m_7)(m_4 - m_5), \\
Z([2]\boldsymbol{P_1}) = (m_4 + m_5)(m_4 - m_5),
\end{gathered}$$

that is, with $\underline{10\mathsf{m} + 1\mathsf{s}}$.

Moving the origin. Choosing $\boldsymbol{O'} = (0 : 1 : 0)$ as the neutral element results in translating the group law. If we let $\oplus'$ denote the corresponding point addition, we have $\boldsymbol{P_1} \oplus' \boldsymbol{P_2} = (\boldsymbol{P_1} \ominus \boldsymbol{O'}) \oplus (\boldsymbol{P_2} \ominus \boldsymbol{O'}) \oplus \boldsymbol{O'} = \boldsymbol{P_1} \oplus \boldsymbol{P_2} \oplus \boldsymbol{O'}$. Hence, we get

$$\begin{cases}
X_3 = (X_1 Z_2 + X_2 Z_1)(Y_1 Y_2 + Z_1 Z_2)(Y_1 Z_2 + Y_2 Z_1) \\
Y_3 = (X_1 X_2 - Z_1 Z_2)({Z_1}^2 {Z_2}^2 - {Y_1}^2 {Y_2}^2) \\
Z_3 = (Y_1 Z_2 + Y_2 Z_1)(X_1 X_2 + Z_1 Z_2)(Y_1 Y_2 - Z_1 Z_2)
\end{cases}.$$

This can be evaluated with $\underline{11\mathsf{m}}$ as

$$\begin{gathered}
m_1 = X_1 X_2,\ \ m_2 = Y_1 Y_2,\ \ m_3 = Z_1 Z_2, \\
m_4 = (X_1 + Z_1)(X_2 + Z_2) - m_1 - m_3,\ \ m_5 = (Y_1 + Z_1)(Y_2 + Z_2) - m_2 - m_3, \\
X_3 = m_4(m_2 + m_3)m_5,\ \ Y_3 = (m_1 - m_3)(m_3 - m_2)(m_3 + m_2), \\
Z_3 = m_5(m_1 + m_3)(m_2 - m_3)\ .
\end{gathered} \tag{6}$$

This addition formula is unified: it can be used for doubling as well.

For a mixed point addition (*i.e.*, when $Z_2 = 1$), we have $m_3 = Z_1$ and the number of required multiplications drops to $\underline{10\mathsf{m}}$. When used for dedicated doubling, the above addition formula requires $\underline{6\mathsf{m} + 5\mathsf{s}}$, which can equivalently be obtained as

$$\begin{gathered} s_1 = {X_1}^2,\ \ s_2 = {Y_1}^2,\ \ s_3 = {Z_1}^2, \\ s_4 = (X_1 + Y_1)^2 - s_1 - s_2,\ \ s_5 = (Y_1 + Z_1)^2 - s_2 - s_3, \\ X([2]\boldsymbol{P_1}) = 2s_3 s_4(s_2 + s_3),\ \ Y([2]\boldsymbol{P_1}) = (s_1 - s_3)(s_3 - s_2)(s_3 + s_2), \\ Z([2]\boldsymbol{P_1}) = s_5(s_1 + s_3)(s_2 - s_3)\ . \end{gathered} \tag{7}$$

Note that the expression for the inverse of point $\boldsymbol{P_1}$ is unchanged: $\ominus'\boldsymbol{P_1} = \ominus(\boldsymbol{P_1} \ominus \boldsymbol{O'}) \oplus \boldsymbol{O'} = \ominus\boldsymbol{P_1} = (X_1 : Y_1 : -Z_1)$.

3.2 More Formulæ

Alternative addition formulæ can be derived using the curve equation. For example, whenever defined, we can write $(x_3, y_3) = (x_1, y_1) \oplus (x_2, y_2)$ with

$$x_3 = \frac{(x_1 - x_2)(y_1 + y_2)}{(y_1 - y_2)(1 - x_1 x_2)} \quad \text{and} \quad y_3 = \frac{(y_1 - y_2)(x_1 + x_2)}{(x_1 - x_2)(1 - y_1 y_2)}\ .$$

In projective coordinates, this gives

$$\begin{cases} X_3 = (X_1 Z_2 - X_2 Z_1)^2 (Y_1 Z_2 + Y_2 Z_1)(Z_1 Z_2 - Y_1 Y_2) \\ Y_3 = (Y_1 Z_2 - Y_2 Z_1)^2 (X_1 Z_2 + X_2 Z_1)(Z_1 Z_2 - X_1 X_2) \\ Z_3 = (X_1 Z_2 - X_2 Z_1)(Y_1 Z_2 - Y_2 Z_1)(Z_1 Z_2 - X_1 X_2)(Z_1 Z_2 - Y_1 Y_2) \end{cases},$$

which can be evaluated with $13\mathsf{m}$ as

$$\begin{gathered} m_1 = X_1 Z_2,\ \ m_2 = X_2 Z_1,\ \ m_3 = Y_1 Z_2,\ \ m_4 = Y_2 Z_1, \\ m_5 = (Z_1 - X_1)(Z_2 + X_2) + m_1 - m_2,\ \ m_6 = (Z_1 - Y_1)(Z_2 + Y_2) + m_3 - m_4, \\ m_7 = (m_1 - m_2)m_6,\ \ m_8 = (m_3 - m_4)m_5, \\ X_3 = (m_1 - m_2)(m_3 + m_4)m_7,\ \ Y_3 = (m_1 + m_2)(m_3 - m_4)m_8,\ \ Z_3 = m_7 m_8\ . \end{gathered}$$

Although not as efficient as the usual addition, this alternative formula is useful in some pairing computations (see Section 4.2).

3.3 Twisted Curves

As shown in Theorem 1, the group of points of a Huff elliptic curve contains a copy of $\mathbb{Z}/4\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$. This implies that the curve order is a multiple of 8. Several cryptographic standards, however, require elliptic curves with group order of the form $h\,n$ where $h \in \{1, 2, 3, 4\}$ and n is a prime.

We can generalize Huff's model to accommodate the case $h = 4$. Let $\mathcal{P} \in \mathbb{K}[t]$ denote a monic polynomial of degree 2, with non-zero discriminant, and such that $\mathcal{P}(0) \neq 0$. We can then introduce the cubic curve

$$ax\mathcal{P}(y) = by\mathcal{P}(x)$$

where $a, b \in \mathbb{K}^\times$. The set of points $\{(0:0:1),(0:1:0),(1:0:0),(a:b:0)\} \cong \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$ belongs to the curve. Moreover, when $\mathcal{P}$ factors in $\mathbb{K}$ — *i.e.*, when $\mathcal{P}(t) = (t-\omega_1)(t-\omega_2)$ with $\omega_1, \omega_2 \in \mathbb{K}^\times$, the four points $(\pm\omega_1 : \pm\omega_2 : 1)$ are also on the curve.

When $\mathrm{Char}\,\mathbb{K} \neq 2$, we consider $\mathcal{P}(t) = t^2 - d$ for some $d \in \mathbb{K}^\times$. So we deal with the set of projective points $(X:Y:Z) \in \mathbb{P}^2(\mathbb{K})$ satisfying the non-singular cubic equation

$$\hat{E}_d : aX(Y^2 - dZ^2) = bY(X^2 - dZ^2) \tag{8}$$

where $a, b, d \in \mathbb{K}^\times$ and $a^2 \neq b^2$. This equation corresponds to Weierstraß equation $V^2W = U(U + \frac{a^2}{d}W)(U + \frac{b^2}{d}W)$ under the inverse transformations $(X:Y:Z) = \big(b(dU + a^2W) : a(dU + b^2W) : dV\big)$ and $(U:V:W) = \big(ab(bX - aY) : ab(b^2 - a^2)Z : d(-aX + bY)\big)$. The transformation $(X:Y:Z) \leftarrow (X:Y:Z\sqrt{d})$ induces an isomorphism from $E = \hat{E}_1$ to $\hat{E}_d$ over $\mathbb{K}(\sqrt{d})$. Curves $\hat{E}_d$ are therefore *quadratic twists* of Huff curves.

In affine coordinates, we consider the curve equation $ax(y^2 - d) = by(x^2 - d)$. The sum of two finite points $\boldsymbol{P_1} = (x_1, y_1)$ and $\boldsymbol{P_2} = (x_2, y_2)$ such that $x_1x_2 \neq \pm d$ and $y_1y_2 \neq \pm d$ is given by (x_3, y_3) where

$$x_3 = \frac{d(x_1 + x_2)(d + y_1y_2)}{(d + x_1x_2)(d - y_1y_2)} \quad \text{and} \quad y_3 = \frac{d(y_1 + y_2)(d + x_1x_2)}{(d - x_1x_2)(d + y_1y_2)} \ . \tag{9}$$

Extending the computations of §2.2, it is readily verified that the sum of two points can be evaluated with 12m (plus a couple of multiplications by constant d) using projective coordinates. The faster computations of the previous section also generalize to twisted curves.

3.4 Binary Fields

Huff's form can be extended to a binary field as

$$ax(y^2 + y + 1) = by(x^2 + x + 1) \ .$$

This curve is birationally equivalent to Weierstraß curve

$$v(v + (a+b)u) = u(u + a^2)(u + b^2)$$

under the inverse maps

$$(x, y) = \left(\frac{b(u + a^2)}{v}, \frac{a(u + b^2)}{v + (a+b)u}\right) \quad \text{and} \quad (u, v) = \left(\frac{ab}{xy}, \frac{ab(axy + b)}{x^2y}\right) \ .$$

The neutral element is $\boldsymbol{O} = (0, 0)$.

4 Pairings

4.1 Preliminaries

Let $(E, \boldsymbol{O})$ be an elliptic curve over $\mathbb{K} = \mathbb{F}_q$, with q odd. Suppose that $\#E(\mathbb{F}_q) = hn$ where n is a prime such that $\gcd(n, q) = 1$. Let further k denote the

embedding degree with respect to n, namely the smallest extension $\mathbb{F}_{q^k}$ of $\mathbb{F}_q$ containing all n-th roots of unity. In other words, k is the smallest positive integer k such that $n \mid q^k - 1$. For better efficiency, we further assume that $k > 1$ is even.

For any point $\boldsymbol{P} \in E(\mathbb{F}_q)[n]$, we let $f_{\boldsymbol{P}}$ denote a rational function on E defined over $\mathbb{F}_q$ such that $\mathrm{div}(f_{\boldsymbol{P}}) = n\boldsymbol{P} - n\boldsymbol{O}$; it exists and is unique up to a multiplicative constant, according to the Riemann-Roch theorem. The group of n-th roots of unity in $\mathbb{F}_{q^k}$ is denoted by μ_n. The *(reduced) Tate pairing* is then defined as

$$T_n : E(\mathbb{F}_q)[n] \times E(\mathbb{F}_{q^k})/[n]E(\mathbb{F}_{q^k}) \to \mu_n : (\boldsymbol{P}, \boldsymbol{Q}) \mapsto f_{\boldsymbol{P}}(\boldsymbol{Q})^{(q^k-1)/n} \ .$$

This definition does not depend on the choice of $f_{\boldsymbol{P}}$ with the appropriate divisor, nor on the class of $\boldsymbol{Q}$ mod $[n]E(\mathbb{F}_{q^k})$.

In practice, T_n can be computed using a technique due to Miller [21], in terms of rational functions $g_{\boldsymbol{R},\boldsymbol{P}}$ depending on $\boldsymbol{P}$ and on a variable point $\boldsymbol{R}$. Function $g_{\boldsymbol{R},\boldsymbol{P}}$ is the so-called *line function* with divisor $\boldsymbol{R} + \boldsymbol{P} - \boldsymbol{O} - (\boldsymbol{R} \oplus \boldsymbol{P})$, which arises in addition formulæ when E is represented as a plane cubic. The core idea is to derive function $f_{\boldsymbol{P}}$ iteratively. Letting $f_{i,\boldsymbol{P}}$ be the function with divisor $\mathrm{div}(f_{i,\boldsymbol{P}}) = i\boldsymbol{P} - ([i]\boldsymbol{P}) - (i-1)\boldsymbol{O}$, it is easily verified that

$$f_{i+j,\boldsymbol{P}} = f_{i,\boldsymbol{P}} \cdot f_{j,\boldsymbol{P}} \cdot g_{[i]\boldsymbol{P},[j]\boldsymbol{P}} \ .$$

Observe that $f_{1,\boldsymbol{P}} = 1$ and $f_{n,\boldsymbol{P}} = f_{\boldsymbol{P}}$. Hence, if $n = \overline{n_{\ell-1}n_{\ell-1}\cdots n_0}_2$ is the binary representation of n, the Tate pairing can be computed as follows.

Algorithm 1. Miller's algorithm

1: $f \leftarrow 1$; $\boldsymbol{R} \leftarrow \boldsymbol{P}$
2: **for** $i = \ell - 2$ down to 0 **do**
3: $\quad f \leftarrow f^2 \cdot g_{\boldsymbol{R},\boldsymbol{R}}(\boldsymbol{Q})$; $\boldsymbol{R} \leftarrow [2]\boldsymbol{R}$
4: $\quad$ **if** $(n_i = 1)$ **then**
5: $\quad\quad f \leftarrow f \cdot g_{\boldsymbol{R},\boldsymbol{P}}(\boldsymbol{Q})$; $\boldsymbol{R} \leftarrow \boldsymbol{R} \oplus \boldsymbol{P}$
6: $\quad$ **end if**
7: **end for**
8: **return** $f^{(q^k-1)/n}$

Contrary to Edwards curves or Jacobi quartics, Huff curves are represented as plane cubics. This makes Miller's algorithm, along with a number of improvements proposed for Weierstraß curves (*e.g.*, as presented in [3]), directly applicable to the computation of pairings over Huff curves.

4.2 Pairing Formulæ for Huff Curves

Throughout the for-loop of Algorithm 1, the line function is always evaluated at the same point $\boldsymbol{Q} \in E(\mathbb{F}_{q^k}) \setminus E(\mathbb{F}_q)$. It is therefore customary to represent

this point in affine coordinates. In our case, it is most convenient to choose the coordinates of $\boldsymbol{Q}$ as $\boldsymbol{Q} = (y, z) = (1 : y : z)$. Indeed, since the embedding degree k is even, the field $\mathbb{F}_{q^k}$ can be represented as $\mathbb{F}_{q^{k/2}}(\alpha)$, where α is any quadratic non-residue in $\mathbb{F}_{q^{k/2}}$. As a result, $\boldsymbol{Q}$ can be chosen of the form $\boldsymbol{Q} = (y_{\boldsymbol{Q}}, z_{\boldsymbol{Q}}\alpha)$ with $y_{\boldsymbol{Q}}, z_{\boldsymbol{Q}} \in \mathbb{F}_{q^{k/2}}$ [4]. To do so, it suffices to pick a point on a quadratic twist of E over $\mathbb{F}_{q^{k/2}}$ and take its image under the isomorphism over $\mathbb{F}_{q^k}$.

Now, for any two points $\boldsymbol{R}$, $\boldsymbol{P}$ in $E(\mathbb{F}_q)$, let $\ell_{\boldsymbol{R},\boldsymbol{P}}$ denote the rational function vanishing on the line through $\boldsymbol{R}$ and $\boldsymbol{P}$. In general, we have

$$\ell_{\boldsymbol{R},\boldsymbol{P}}(\boldsymbol{Q}) = \frac{(zX_{\boldsymbol{P}} - Z_{\boldsymbol{P}}) - \lambda(yX_{\boldsymbol{P}} - Y_{\boldsymbol{P}})}{Y_{\boldsymbol{P}}}$$

where λ is the "(y, z)-slope" of the line through $\boldsymbol{R}$ and $\boldsymbol{P}$. Then, the divisor of $\ell_{\boldsymbol{R},\boldsymbol{P}}$ is

$$\mathrm{div}(\ell_{\boldsymbol{R},\boldsymbol{P}}) = \boldsymbol{R} + \boldsymbol{P} + \boldsymbol{T} - (1 : 0 : 0) - (0 : 1 : 0) - (a : b : 0)$$

where $\boldsymbol{T}$ is the third point of intersection (counting multiplicities) of the line through $\boldsymbol{R}$ and $\boldsymbol{P}$ with the elliptic curve. In particular, if the neutral element of the group law $\oplus$ is denoted by $\boldsymbol{U}$, the line function $g_{\boldsymbol{R},\boldsymbol{P}}$ can be written as

$$g_{\boldsymbol{R},\boldsymbol{P}} = \frac{\ell_{\boldsymbol{R},\boldsymbol{P}}}{\ell_{\boldsymbol{R}\oplus\boldsymbol{P},\boldsymbol{U}}} \ .$$

We concentrate on the case when $\boldsymbol{U} = \boldsymbol{O} = (0 : 0 : 1)$. Then for any $\boldsymbol{Q} = (y_{\boldsymbol{Q}}, z_{\boldsymbol{Q}}\alpha)$, we have

$$\ell_{\boldsymbol{R}\oplus\boldsymbol{P},\boldsymbol{O}}(\boldsymbol{Q}) = y_{\boldsymbol{Q}} - \frac{Y_{\boldsymbol{R}\oplus\boldsymbol{P}}}{X_{\boldsymbol{R}\oplus\boldsymbol{P}}} \in \mathbb{F}_{q^{k/2}} \ .$$

Since this quantity lies in a proper subfield of $\mathbb{F}_{q^k}$, it goes to 1 after the final exponentiation in Miller's algorithm, which means that it can be discarded altogether. Similarly, divisions by $X_{\boldsymbol{P}}$ can be omitted, and denominators in the expression of λ can be canceled. In other words, if $\lambda = A/B$, we can compute the line function as

$$g_{\boldsymbol{R},\boldsymbol{P}}(\boldsymbol{Q}) = (zX_{\boldsymbol{P}} - Z_{\boldsymbol{P}}) \cdot B - (yX_{\boldsymbol{P}} - Y_{\boldsymbol{P}}) \cdot A$$

and get the required result.

We can now detail precise formulæ for the addition and doubling steps in the so-called Miller loop (*i.e.*, the main for-loop in Algorithm 1). We let M and S represent the cost of a multiplication and of a squaring in $\mathbb{F}_{q^k}$ while m and s are operations in $\mathbb{F}_q$ as before.

Addition step. In the case of addition, the (y, z)-slope of the line through $\boldsymbol{R} = (X_R : Y_R : Z_R)$ and $\boldsymbol{P} = (X_P : Y_P : Z_P)$ is

$$\lambda = \frac{Z_{\boldsymbol{R}}X_{\boldsymbol{P}} - Z_{\boldsymbol{P}}X_{\boldsymbol{R}}}{Y_{\boldsymbol{R}}X_{\boldsymbol{P}} - Y_{\boldsymbol{P}}X_{\boldsymbol{R}}} \ .$$

Therefore, the line function to be evaluated is of the form

$$g_{\boldsymbol{R},\boldsymbol{P}}(\boldsymbol{Q}) = (z_{\boldsymbol{Q}}\alpha \cdot X_{\boldsymbol{P}} - Z_{\boldsymbol{P}})(Y_{\boldsymbol{R}}X_{\boldsymbol{P}} - Y_{\boldsymbol{P}}X_{\boldsymbol{R}}) - (y_{\boldsymbol{Q}} \cdot X_{\boldsymbol{P}} - Y_{\boldsymbol{P}})(Z_{\boldsymbol{R}}X_{\boldsymbol{P}} - Z_{\boldsymbol{P}}X_{\boldsymbol{R}}) \ .$$

Since $\boldsymbol{P}$ and $\boldsymbol{Q}$ are constant throughout the loop, the values depending only on $\boldsymbol{P}$ and $\boldsymbol{Q}$ — in this case $y'_{\boldsymbol{Q}} = y_{\boldsymbol{Q}} \cdot X_{\boldsymbol{P}} - Y_{\boldsymbol{P}}$ and $z'_{\boldsymbol{Q}} = z_{\boldsymbol{Q}}\alpha \cdot X_{\boldsymbol{P}}$, can be precomputed.

Then, each Miller addition step requires computing $\boldsymbol{R} \oplus \boldsymbol{P}$ (one addition on the curve over $\mathbb{F}_q$), evaluating $g_{\boldsymbol{R},\boldsymbol{P}}(\boldsymbol{Q})$, and computing $f \cdot g_{\boldsymbol{R},\boldsymbol{P}}(\boldsymbol{Q})$ (one multiplication in the field $\mathbb{F}_{q^k}$).

We consider two types of Miller addition steps: full addition, for which no assumption is made on the representation of $\boldsymbol{P}$, and mixed addition, for which we further assume that $\boldsymbol{P}$ is given in affine coordinates (*i.e.*, $X_{\boldsymbol{P}} = 1$). Both steps start with computing $\boldsymbol{R} \oplus \boldsymbol{P}$, including all intermediate results.

Full addition. Computing $\boldsymbol{R} \oplus \boldsymbol{P}$ requires $13\mathsf{m}$ using the dedicated addition formula from §3.1, including all intermediate results $m_1, \ldots, m_8$. Compute further $m_9 = (X_{\boldsymbol{R}} + Y_{\boldsymbol{R}})(X_{\boldsymbol{P}} - Y_{\boldsymbol{P}})$. We then have

$$g_{\boldsymbol{R},\boldsymbol{P}}(\boldsymbol{Q}) = (z'_{\boldsymbol{Q}} - Z_{\boldsymbol{P}})(m_9 + m_5 - m_6) - y'_{\boldsymbol{Q}}(m_1 - m_2)$$

where the first term requires $(\frac{k}{2} + 1)\mathsf{m}$ and the second term $\frac{k}{2}\mathsf{m}$. With the final multiplication over $\mathbb{F}_{q^k}$, the total cost of full addition is thus of $\underline{1\mathsf{M} + (k + 15)\mathsf{m}}$.

Mixed addition. Now that $X_{\boldsymbol{P}} = 1$, computing $\boldsymbol{R} \oplus \boldsymbol{P}$ using the formula from §2.2, including all the intermediate results $m_1, \ldots, m_9$, only requires $11\mathsf{m}$, since the computation of m_1 is free. We then have

$$g_{\boldsymbol{R},\boldsymbol{P}}(\boldsymbol{Q}) = (z'_{\boldsymbol{Q}} - Z_{\boldsymbol{P}})(Y_{\boldsymbol{R}} - Y_{\boldsymbol{P}}X_{\boldsymbol{R}}) - y'_{\boldsymbol{Q}}(2Z_{\boldsymbol{R}} - m_4)$$

where both terms require the same number of multiplications as before, plus one for $Y_{\boldsymbol{P}}X_{\boldsymbol{R}}$. The total cost of mixed addition is thus of $\underline{1\mathsf{M} + (k + 13)\mathsf{m}}$.

Doubling step. In the case of doubling, the (y, z)-slope of the tangent line at $\boldsymbol{R} = (X_R : Y_R : Z_R)$ is

$$\lambda = \frac{a(Z_{\boldsymbol{R}})^2 - 2bY_{\boldsymbol{R}}Z_{\boldsymbol{R}} - a(X_{\boldsymbol{R}})^2}{b(Y_{\boldsymbol{R}})^2 - 2aY_{\boldsymbol{R}}Z_{\boldsymbol{R}} - b(X_{\boldsymbol{R}})^2} = \frac{A}{B} \ .$$

Thus, the line function is of the form

$$g_{\boldsymbol{R},\boldsymbol{R}}(\boldsymbol{Q}) = z_{\boldsymbol{Q}}\alpha \cdot X_{\boldsymbol{R}}B - Z_{\boldsymbol{R}}B - y_{\boldsymbol{Q}} \cdot X_{\boldsymbol{R}}A + Y_{\boldsymbol{R}}A \ .$$

Miller's doubling involves computing the point $[2]\boldsymbol{R}$, which we do using the formulæ from §2.2 in $7\mathsf{m} + 5\mathsf{s}$. Then the quantities A and B are obtained by computing the additional product $m_{10} = 2Y_{\boldsymbol{R}}Z_{\boldsymbol{R}} = (Y_{\boldsymbol{R}} + Z_{\boldsymbol{R}})^2 - m_2 - m_3$ using a single squaring. Computing $g_{\boldsymbol{R},\boldsymbol{R}}(\boldsymbol{Q})$ requires multiplying those two values by $X_{\boldsymbol{R}}$ and $Y_{\boldsymbol{R}}$ (resp. $X_{\boldsymbol{R}}$ and $Z_{\boldsymbol{R}}$), hence an additional $4\mathsf{m}$. And finally, multiplications by $y_{\boldsymbol{Q}}$ and $z_{\boldsymbol{Q}}\alpha$ both require $\frac{k}{2}\mathsf{m}$. Taking into account the multiplication and the squaring in $\mathbb{F}_{q^k}$ needed to complete the doubling step, the total cost of Miller doubling is thus of $\underline{1\mathsf{M} + 1\mathsf{S} + (k + 11)\mathsf{m} + 6\mathsf{s}}$.

5 Conclusion

This paper introduced and studied Huff's model, a new representation of elliptic curves to be considered alongside previous models such as Montgomery, Doche-Icart-Kohel and Edwards. This new model provides efficient arithmetic, competitive with some of the fastest known implementations (although not quite as fast as "inverted Edwards" for now). Moreover, it has a number of additional desirable properties, including unified/complete addition laws and formulæ that do not depend on curve parameters (both properties are useful in cryptographic applications to thwart certain implementation attacks). It is also suitable to other computations on elliptic curves, such as the evaluation of pairings.

We believe that this model is worthy of consideration by the community, and hope our contribution might spark further research into efficient implementations of elliptic curve arithmetic.

Acknowledgments. We are grateful to an anonymous referee for useful comments. This work was partly supported by the French ANR-07-TCOM-013-04 PACE Project and by the European Commission through the IST Program under Contract ICT-2007-216646 ECRYPT II.

References

1. Arène, C., Lange, T., Naehrig, M., Ritzenthaler, C.: Faster computation of the Tate pairing. In: Cryptology ePrint Archive, Report 2009/155 (2009), http://eprint.iacr.org/
2. Atkin, A.O.L., Morain, F.: Elliptic curves and primality proving. Math. Comp. 61(203), 29–68 (1993)
3. Barreto, P.S.L.M., Lynn, B., Scott, M.: Efficient implementation of pairing-based cryptosystems. J. Cryptology 17(4), 321–334 (2004)
4. Barreto, P.S.L.M., Lynn, B., Scott, M.: On the selection of pairing-friendly groups. In: Matsui, M., Zuccherato, R.J. (eds.) SAC 2003. LNCS, vol. 3006, pp. 17–25. Springer, Heidelberg (2004)
5. Bernstein, D.J., Birkner, P., Joye, M., Lange, T., Peters, C.: Twisted Edwards curves. In: Vaudenay, S. (ed.) AFRICACRYPT 2008. LNCS, vol. 5023, pp. 389–405. Springer, Heidelberg (2008)
6. Bernstein, D.J., Lange, T.: Explicit-formulas database, http://www.hyperelliptic.org/EFD/
7. Bernstein, D.J., Lange, T.: Faster addition and doubling on elliptic curves. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 29–50. Springer, Heidelberg (2007)
8. Bernstein, D.J., Lange, T.: Inverted Edwards coordinates. In: Boztaş, S., Lu, H.-F(F.) (eds.) AAECC 2007. LNCS, vol. 4851, pp. 20–27. Springer, Heidelberg (2007)
9. Bernstein, D.J., Lange, T., Farashahi, R.R.: Binary Edwards curves. In: Oswald, E., Rohatgi, P. (eds.) CHES 2008. LNCS, vol. 5154, pp. 244–265. Springer, Heidelberg (2008)
10. Blake, I.F., Seroussi, G., Smart, N.P.: Advances in Elliptic Curve Cryptography, ch. V. London Mathematical Society Lecture Note Series, vol. 317. Cambridge University Press, Cambridge (2005)

11. Das, M.P.L., Sarkar, P.: Pairing computation on twisted Edwards form elliptic curves. In: Galbraith, S.D., Paterson, K.G. (eds.) Pairing 2008. LNCS, vol. 5209, pp. 192–210. Springer, Heidelberg (2008)
12. Edwards, H.M.: A normal form for elliptic curves. Bull. Am. Math. Soc., New Ser. 44(3), 393–422 (2007)
13. Goldwasser, S., Kilian, J.: Primality testing using elliptic curves. J. ACM 46(4), 450–472 (1999)
14. Hisil, H., Wong, K.K.-H., Carter, G., Dawson, E.: Twisted Edwards curves revisited. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 326–343. Springer, Heidelberg (2008)
15. Huff, G.B.: Diophantine problems in geometry and elliptic ternary forms. Duke Math. J. 15, 443–453 (1948)
16. Ionica, S., Joux, A.: Another approach to pairing computation in Edwards coordinates. In: Chowdhury, D.R., Rijmen, V., Das, A. (eds.) INDOCRYPT 2008. LNCS, vol. 5365, pp. 400–413. Springer, Heidelberg (2008)
17. Koblitz, A.H., Koblitz, N., Menezes, A.: Elliptic curve cryptography: The serpentine course of a paradigm shift. J. Number Theory (to appear)
18. Koblitz, N.: Elliptic curve cryptosystems. Math. Comp. 48, 203–209 (1987)
19. Lenstra Jr., H.W.: Factoring integers with elliptic curves. Ann. Math. 126(2), 649–673 (1987)
20. Miller, V.S.: Use of elliptic curves in cryptography. In: Williams, H.C. (ed.) CRYPTO 1985. LNCS, vol. 218, pp. 417–426. Springer, Heidelberg (1986)
21. Miller, V.S.: The Weil paring, and its efficient implementation. J. Cryptology 17(1), 235–261 (2004)
22. Montgomery, P.L.: Speeding up the Pollard and elliptic curve methods of factorization. Mathematics of Computation 48(177), 243–264 (1987)
23. Morain, F.: Primality proving using elliptic curves: An update. In: Buhler, J.P. (ed.) ANTS 1998. LNCS, vol. 1423, pp. 111–127. Springer, Heidelberg (1998)
24. Peeples Jr., W.D.: Elliptic curves and rational distance sets. Proc. Am. Math. Soc. 5, 29–33 (1954)
25. Silverman, J.H.: The Arithmetic of Elliptic Curves, ch III. Graduate Texts in Mathematics, vol. 106. Springer, Heidelberg (1986)

Efficient Pairing Computation with Theta Functions

David Lubicz[1,2] and Damien Robert[3]

[1] DGA-MI, BP 7419, F-35174 Bruz
[2] IRMAR, Universté de Rennes 1, Campus de Beaulieu, F-35042 Rennes
[3] LORIA, CARAMEL Project,
Campus Scientifique, BP 239, 54506 Vandoeuvre-lès-Nancy Cedex

Abstract. In this paper, we present a new approach based on theta functions to compute Weil and Tate pairings. A benefit of our method, which does not rely on the classical Miller's algorithm, is its generality since it extends to all abelian varieties the classical Weil and Tate pairing formulas. In the case of dimension 1 and 2 abelian varieties our algorithms lead to implementations which are efficient and naturally deterministic. We also introduce symmetric Weil and Tate pairings on Kummer varieties and explain how to compute them efficiently. We exhibit a nice algorithmic compatibility between some algebraic groups quotiented by the action of the automorphism -1, where the $\mathbb{Z}$-action can be computed efficiently with a Montgomery ladder type algorithm.

1 Introduction

In recent years, many new and interesting cryptographic protocols have been proposed which use the existence of pairings on abelian varieties. In order to obtain efficient and secure implementations of these protocols it is important to be able to compute quickly these pairings. Miller has proposed a method (see for instance [2]) to compute the function on an algebraic curve given up to a constant factor by the data of a principal divisor. This method is a key ingredient of all known algorithms to compute pairings. In this paper, we propose a different approach based on theta functions. We first make explicit the link between Weil and Tate pairings and the intersection pairing on the degree 1 homology of an abelian variety. Our method appears to be a very natural and straightforward way to compute the pairing associated to the Riemann form (or its arithmetic counterpart the commutator pairing) of an abelian variety. It is then easy to deduce practical formulas to compute Weil and Tate pairings. A first benefit of our approach is its generality: where Miller's algorithm rely on the representation of an abelian variety as the Jacobian of an algebraic curve, our method works with any abelian varieties. The case of the Tate pairing is noticeable: while the original definition of Tate [8] deals with any abelian varieties, the formula of Lichtenbaum [9] used in cryptographic applications is restricted to Jacobian of curves. This restriction does not appear in our formulas. Our algorithm also

G. Hanrot, F. Morain, and E. Thomé (Eds.): ANTS-IX 2010, LNCS 6197, pp. 251–269, 2010.

expand the algorithmic toolbox based on theta functions to compute with abelian varieties.

For the complexity analysis of our algorithm we focus on the case of level 2 and 4 theta functions in order to obtain the best running time and memory consumption. The only difference between the two cases lies in the initialisation phase of the algorithm: in level 4 one can recover enough information from the data of two points to compute the pairings. This is not possible with the level 2 embedding since it does not distinguish a point and its opposite. Nonetheless it is possible to define a "symmetric pairing" on the quotient of an abelian variety by the action of the automorphism -1. These notions extend the definition of the trace pairing proposed in [3].

We have chosen to present all the formulas of this paper using the classical analytic theory of theta functions. In order to consider also rationality problems which are essential to the definition of the Tate pairing, we make the assumption that all the abelian varieties that we consider are defined over a number field K and we suppose given a fixed embedding of K in its algebraic closure $\mathbb{C}$. Nonetheless, it should be understood that all our algorithms apply to the case of abelian varieties defined over any field of characteristic not equal to 2. To see this one can invoke the Lefschetz's principle or use Mumford's theory of algebraic theta functions. We refer to [10] for proofs of the main formulas of this paper in the theory of Mumford.

Our paper in organized as follows: in Section 2 we recall some basic definitions about theta functions. The Section 3 we give a method to compute the usual pairings by using a double and add algorithm based a theta addition formula. In Section 5 we make a precise assessment about the complexity of our algorithm. We also introduce symmetric pairings on Kummer varieties and explain how to adapt our algorithms to compute them efficiently. We end the paper with an example of computation in Section 6.

2 Some Notations and Basic Facts

In this section, in order to fix the notations, we recall some well known facts on analytic theta functions (see for instance [14,6]). Let $\mathbb{H}_g$ be the g dimensional Siegel upper-half space which is the set of $g \times g$ symmetric matrices Ω whose imaginary part is positive definite. For $\Omega \in \mathbb{H}_g$, we denote by $\Lambda_\Omega = \Omega\mathbb{Z}^g + \mathbb{Z}^g$ the lattice of $\mathbb{C}^g$ defined by Ω. If A is an abelian variety of dimension g over the number field K with a principal polarisation then A is analytically isomorphic to $\mathbb{C}^g/\Lambda_\Omega$ for a certain $\Omega \in \mathbb{H}_g$. In the rest of this paper, we denote by $\pi : \mathbb{C}^g \to \mathbb{C}^g/\Lambda_\Omega = A$ the canonical projection. The classical theory of theta functions gives a lot of functions on $\mathbb{C}^g$ that are pseudo-periodic with respect to Λ_Ω and can be used as a projective coordinate system for A. More precisely, for $a, b \in \mathbb{Q}^g$, the theta function with rational characteristics (a, b) is an analytic function on $\mathbb{C}^g \times \mathbb{H}_g$ given by:

$$\theta\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right](z, \Omega) = \sum_{n \in \mathbb{Z}^g} \exp\left[\pi i^t (n+a).\Omega.(n+a) + 2\pi i^t (n+a).(z+b)\right]. \quad (1)$$

In order to write the pseudo-periodicity relations verified by the theta functions it is convenient to introduce a certain pairing on $\mathbb{C}^g$. First we identify $\mathbb{C}^g$ to $\mathbb{R}^{2g}$ via the isomorphism $\mathbb{R}^{2g} \to \mathbb{C}^g$, $(x_1, x_2) \mapsto \Omega x_1 + x_2$. Then for $\alpha, \beta \in \mathbb{R}^{2g}$ with $\alpha = (\alpha_1, \alpha_2)$ and $\beta = (\beta_1, \beta_2)$, we put $e_\Omega(\alpha, \beta) = \exp(2\pi i(\alpha_1\beta_2 - \alpha_2\beta_1))$. The pseudo-periodicity of $\theta\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right]$ is given by

$$\theta\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right](z+\Omega.m+n, \Omega) = e_\Omega(\Omega.a+b, \Omega.m+n)e^{-\pi i {}^t m.\Omega.m - 2\pi i {}^t m.z}\theta\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right](z, \Omega). \quad (2)$$

We say that a function f on $\mathbb{C}^g$ is Λ_Ω-quasi-periodic of level $\ell \in \mathbb{N}$ if for all $z \in \mathbb{C}^g$ and $m \in \mathbb{Z}^g$, we have: $f(z+m) = f(z)$, $f(z+\Omega.m) = \exp(-\pi i \ell^t m.\Omega.m - 2\pi i \ell^t z.m) f(z)$. For any $\ell \in \mathbb{N}^*$, the set $H_{\Omega,\ell}$ of Λ_Ω-quasi-periodic functions of level ℓ is a finite dimensional $\mathbb{C}$-vector space whose basis can be given by the theta functions with characteristics: $(\theta\left[\begin{smallmatrix} 0 \\ b/\ell \end{smallmatrix}\right](z, \ell^{-1}.\Omega))_{b \in [0,\ldots,\ell-1]^g}$. If $\ell = k^2$, then an alternative basis of $H_{\Omega,\ell}$ is $(\theta\left[\begin{smallmatrix} a/k \\ b/k \end{smallmatrix}\right](kz, \Omega))_{a,b \in [0,\ldots,k-1]^g}$. A theorem of Lefschetz tells that if $\ell \geq 3$, the functions in $H_{\Omega,\ell}$ give a projective embedding of A in $\mathbb{P}^{\ell^g - 1}$, the projective space over $\mathbb{C}$ of dimension $\ell^g - 1$. For $\ell = 2$, the functions in $H_{\Omega,2}$ do not give a projective embedding of A. It is easy to check that for all $f \in H_{\Omega,2}$, we have $f(-z) = f(z)$. Under some well known general conditions [7, cor 4.5.2], the image of the embedding defined by $H_{\Omega,2}$ in $\mathbb{P}^{\ell^2-1}$ is the Kummer variety associated to A, which is the quotient of A by the automorphism -1.

Once we have chosen a level $\ell \in \mathbb{N}$, for the rest of this paper, we adopt the following conventions: we let $Z(\overline{\ell}) = (\mathbb{Z}/\ell\mathbb{Z})^g$ and for a point $z_P \in \mathbb{C}^g$ and $i \in Z(\overline{\ell})$ we put $\theta_i(z_P) = \theta\left[\begin{smallmatrix} 0 \\ i/\ell \end{smallmatrix}\right](z_P, \Omega/\ell)$. If $\ell = k^2$, for $i, j \in Z(\overline{k})$, we let $\theta_{i,j}(z_P) = \theta\left[\begin{smallmatrix} i/k \\ j/k \end{smallmatrix}\right](k.z_P, \Omega)$. We denote by $\widetilde{P}$ the element of $\mathbb{A}^{\ell^g}(\mathbb{C})$ with coordinates $\widetilde{P}_i = \theta_i(z_P)$ and let P be the associated point of A that we consider depending on the situation as embedded in $\mathbb{P}^{\ell^g-1}$ or as a point on the analytic variety $\mathbb{C}^g/\Lambda_\Omega$. In this paper, for $n, \ell \in \mathbb{N}$, such that n divides ℓ we will implicitly consider $Z(\overline{n})$ as a subgroup of $Z(\overline{\ell})$ via the morphism $x \mapsto (\ell/n).x$.

We denote by Ξ_ℓ the theta divisor of level ℓ on A which is the divisor of zero of $\theta\left[\begin{smallmatrix} 0 \\ 0 \end{smallmatrix}\right](z, \ell^{-1}.\Omega)$. There is an isogeny $\varphi_\ell : A \to \hat{A} = \mathrm{Pic}^0_A$, defined by $x \mapsto \tau_x^* \Xi_\ell - \Xi_\ell$ where τ_x is the translation by x morphism on A. The kernel of φ_ℓ is $A[\ell]$. For $\ell = 1$ we let $\Xi_1 = \Xi$. We denote by $K(A)$ the function field of A and if $f \in K(A)$, we denote (f) the divisor of the function f. Let $Z^0(A)$ be the group of 0-cycles of A that is the free commutative group over the set of closed points of A. If $D = \sum n_i P_i$ is an element of $Z^0(A)$ and $f \in K(A)$ then we put $f(D) = \prod_i f(P_i)^{n_i}$.

3 Weil and Tate Pairings and Theta Functions

In this section, we present formulas to compute Weil and Tate pairings from the knowledge of the theta coordinates of some points.

3.1 The Weil Pairing

For $\Omega \in \mathbb{H}_g$, let $A = \mathbb{C}^g/\Lambda_\Omega$ be the associated complex abelian variety and denote by $\pi : \mathbb{C}^g \to A$ the natural projection. Let ℓ be a positive integer, we denote by μ_ℓ the subgroup of $\mathbb{C}^*$ of ℓ^{th} roots of unity. For $z_P, z_Q \in \mathbb{C}^g$, let P, Q be the associated points of A, we consider the pairing: $e_W : A[\ell] \times A[\ell] \to \mu_\ell$, $(P, Q) \mapsto e_\Omega(z_P, z_Q)^\ell$. It is clear that e_W does not depend on the choice of z_P and z_Q representing P and Q respectively and that e_W is a non-degenerate skew linear form. The following proposition gives an expression of this pairing in term of the values of certain theta functions.

Lemma 1. *Let $\Omega \in \mathbb{H}_g$. Let $a, b \in \mathbb{Q}^g$, let ℓ be a positive integer and let $z_P, z_Q \in \mathbb{C}^g$ be such that $\ell.z_P = \ell.z_Q = 0 \mod \Lambda_\Omega$. Set $z_P = \Omega.z_{P1} + z_{P2}$ and $z_Q = \Omega.z_{Q1} + z_{Q2}$ with for $i = 1, 2$, $z_{Pi}, z_{Qi} \in \mathbb{R}^g$. Let $P = \pi(z_P)$ and $Q = \pi(z_Q)$. For all $z \in \mathbb{C}^g$, we have:*

$$e_W(P, Q) = \frac{\theta \begin{bmatrix} a+z_{Q1} \\ b+z_{Q2} \end{bmatrix}(z, \Omega)}{\theta \begin{bmatrix} a+z_{Q1} \\ b+z_{Q2} \end{bmatrix}(z + \ell.z_P, \Omega)} \frac{\theta \begin{bmatrix} a \\ b \end{bmatrix}(z + \ell.z_P, \Omega)}{\theta \begin{bmatrix} a \\ b \end{bmatrix}(z, \Omega)}. \tag{3}$$

Proof. By (2), we have:

$$\theta \begin{bmatrix} a+z_{Q1} \\ b+z_{Q2} \end{bmatrix}(z + \ell.z_P, \Omega) = e_\Omega(\Omega.(a + z_{Q1}) + (b + z_{Q2}), \Omega.\ell z_{P1} + \ell z_{P2})$$
$$\exp[(\pi i \ell^2({}^t z_{P1}.\Omega.z_{P1}) - 2\pi i {}^t z_{P1}.z]\theta \begin{bmatrix} a+z_{Q1} \\ b+z_{Q2} \end{bmatrix}(z, \Omega),$$

$$\theta \begin{bmatrix} a \\ b \end{bmatrix}(z + \ell.z_P, \Omega) = e_\Omega(\Omega.a + b, \Omega.\ell z_{P1} + \ell z_{P2})$$
$$\exp[-\pi i \ell^2({}^t z_{P1}.\Omega.z_{P1}) - 2\pi i {}^t z_{P1}.z]\theta \begin{bmatrix} a \\ b \end{bmatrix}(z, \Omega).$$

The lemma follows immediately.

Let $e'_W : A[\ell] \times A[\ell] \to \mu_\ell$ be the usual Weil pairing. We recall a possible definition for e'_W [13, p. 184]. Let $P, Q \in A[\ell]$. Let $D = \tau_Q^* \Xi - \Xi$, then D represents a point of $\hat{A}[\ell] = \mathrm{Pic}^0_A[\ell]$. As a consequence, there exists a function $f_Q \in K(A)$ such that $(f_Q) = \ell.D$. In the same way, there exists a function $g_Q \in K(A)$ such that $(g_Q) = [\ell]^*(D)$. As $[\ell]^*(f_Q) = \ell.[\ell]^* D = (g_Q^\ell)$ there exists a constant $c \in \mathbb{C}^*$ such that $[\ell]^* f_Q = c.g_Q^\ell$. Thus for X a general point of A, $\frac{g_Q(X)}{g_Q(X+P)}$ is an element of μ_ℓ which is equal to $e'_W(P, Q)$.

Proposition 1. *Keeping the notations from above, let $z_P = \Omega.z_{P1} + z_{P2}$ and $z_Q = \Omega.z_{Q1} + z_{Q2}$ be elements of $\mathbb{C}^g$ such that $P = \pi(z_P)$ and $Q = \pi(z_Q)$. For $z \in \mathbb{C}^g$, we have the following equalities, up to a multiplication by a constant:*

$$g_Q(z) = \frac{\theta \begin{bmatrix} z_{Q1} \\ z_{Q2} \end{bmatrix}(\ell.z, \Omega)}{\theta \begin{bmatrix} 0 \\ 0 \end{bmatrix}(\ell.z, \Omega)}, f_Q(z) = \mu_Q(z)^{-1} \left(\frac{\theta \begin{bmatrix} 0 \\ 0 \end{bmatrix}(z + z_Q)}{\theta \begin{bmatrix} 0 \\ 0 \end{bmatrix}(z)} \right)^\ell, \tag{4}$$

where $\mu_Q(z) : \mathbb{C}^g \to \mathbb{C}$ is given by $\mu_Q(z) = \frac{\theta \begin{bmatrix} 0 \\ 0 \end{bmatrix}(z + \ell z_Q)}{\theta \begin{bmatrix} 0 \\ 0 \end{bmatrix}(z)}$.

Remark 1. In the preceding equations, the domain of the functions g_Q and f_Q is $\mathbb{C}^g$ but we will see in the course of the proof that g_Q and f_Q are periodic with respect to Λ_Ω and are in fact well defined functions on A.

Proof. As $\pi^* \Xi$ is the divisor of zero of $\theta\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right](z,\Omega)$, $\pi^* D$ is the divisor of zero of $g'(z) = \theta\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right](z+z_Q,\Omega)/\theta\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right](z,\Omega)$. But $g(z) = \exp[\pi i^t z_{Q1}\Omega z_{Q1} + 2\pi i^t z_{Q1}(z+z_{Q2})]g'(z)$ has the same zero divisor as $g'(z)$ and $g(z) = \theta\left[\begin{smallmatrix}z_{Q1}\\z_{Q2}\end{smallmatrix}\right](z,\Omega)/\theta\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right](z,\Omega)$. Let $\widetilde{[l]} : \mathbb{C}^g \to \mathbb{C}^g$, $z \mapsto \ell z$. It is clear from its definition that up to a multiplication by a constant $g_Q = g \circ \widetilde{[l]}$ which gives the left hand of (4). It is easily seen using (2) that $g_Q(z)$ is periodic with respect to Λ_Ω and as a consequence descends to a function on A.

We turn to the proof of the second equality. As $\mu_Q(z)$ is a non vanishing function, the zero divisor of the function $\mu_Q(z)^{-1}\left(\theta\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right](z+z_Q)/\theta\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right](z)\right)^\ell$ is $\pi^*(\ell D)$. Moreover, it is easily seen using (2) that this function is periodic with respect to Λ_Ω, and descends to a function on A which up to a multiplication by a constant is $f_Q(z)$.

Corollary 1. *The pairing e_W is the Weil pairing.*

Proof. This is an immediate consequence of Lemma 1 with $a = b = 0$, Proposition 1 and the definition of the Weil pairing as $e'_W(P,Q) = \frac{g_Q(X)}{g_Q(X+P)}$.

Corollary 2. *Let $\Omega \in \mathbb{H}_g$. Let $a, b \in \mathbb{Q}^g$, let ℓ be a positive integer and let $z_P, z_Q \in \mathbb{C}^g$ be such that $\ell.z_P = \ell.z_Q = 0 \mod \Lambda_\Omega$. Let $P, Q \in A$ be such that $P = \pi(z_P)$ and $Q = \pi(z_Q)$ and let:*

$$\begin{aligned} L(z_P, z_Q) &= \frac{\theta\left[\begin{smallmatrix}a\\b\end{smallmatrix}\right](\ell.z_P + z_Q, \Omega)}{\theta\left[\begin{smallmatrix}a\\b\end{smallmatrix}\right](z_Q, \Omega)} \frac{\theta\left[\begin{smallmatrix}a\\b\end{smallmatrix}\right](0, \Omega)}{\theta\left[\begin{smallmatrix}a\\b\end{smallmatrix}\right](\ell.z_P, \Omega)}, \\ R(z_P, z_Q) &= \frac{\theta\left[\begin{smallmatrix}a\\b\end{smallmatrix}\right](\ell.z_Q + z_P, \Omega)}{\theta\left[\begin{smallmatrix}a\\b\end{smallmatrix}\right](z_P, \Omega)} \frac{\theta\left[\begin{smallmatrix}a\\b\end{smallmatrix}\right](0, \Omega)}{\theta\left[\begin{smallmatrix}a\\b\end{smallmatrix}\right](\ell.z_Q, \Omega)}. \end{aligned} \tag{5}$$

If $L(z_P, z_Q)$ and $R(z_P, z_Q)$ are well defined and non null, we have:

$$e_\Omega(z_P, z_Q)^\ell = e_W(P, Q) = L(z_P, z_Q)^{-1}.R(z_P, z_Q). \tag{6}$$

Proof. Since $Q + \ell P = Q$ and $\ell P = 0$, $L(z_P, z_Q)$ does not depend on $\left[\begin{smallmatrix}a\\b\end{smallmatrix}\right]$ so we can assume that $a = b = 0$. The corollary can then be proved by a direct computation.

But it also follows immediately from Proposition 1 and the formula $e_W(P,Q) = f_P(Q-0)/f_Q(P-0)$. In fact, using the notations of Proposition 1, we have

$$\frac{f_P(Q-0)}{f_Q(P-0)} = \frac{\mu_P(z_Q)\mu_Q(0)}{\mu_P(0)\mu_Q(z_P)}.$$

The result follows an immediate computation.

Remark 2. One can recognize in (6) a classical formula to compute the first Chern class of a line bundle from the knowledge of its factors of automorphy, see for instance [1, Th. 2.1.2].

3.2 The Tate Pairing

Let K be a number field and we suppose that A is defined over K. In this section, we suppose that $\mu_\ell \subset K$ and that $A[\ell]$ is rational over K. Let $\overline{K}$ be the algebraic closure of K and let $G = \mathrm{Gal}(\overline{K}/K)$. Let $\delta_1 : K^*/K^{*\ell} \to \mathrm{Hom}(G, \mu_\ell)$ (resp. $\delta_2 : A(K)/[\ell]A(K) \to \mathrm{Hom}(G, A[\ell])$) be the connecting morphism of the Galois cohomology long exact sequence associated to the Kummer exact sequence (resp. to the exact sequence $0 \to A[\ell] \to A(\overline{K}) \to A(\overline{K}) \to 0$). There exists a bilinear application often referred to as the Tate pairing $e_T : A(K)/[\ell]A(K) \times A[\ell] \to K^*/K^{*\ell}$ such that for $(P, Q) \in A(K)/[\ell]A(K) \times A[\ell]$, $e_W(\delta_2(P), Q) = \delta_1(e_T(P, Q))$. In the statement of the next proposition, we suppose that the principal polarization $\mathscr{L}$ of A defined by the matrix period is defined over K. Thus for any $X \in A(K)$ there exits $z_X \in \mathbb{C}^g$ such that $\pi(z_X) = X$ and $\theta(z_X)/\theta(0) \in K$. In general this rationality condition on $\mathscr{L}$ is not verified but we will see later on in Remark 4 how to adapt the formulas of the next proposition to cover the general case.

Proposition 2. *Let K be a number field and let A be a dimension g abelian variety over K. Let $\Omega \in \mathbb{H}_g$ be such that A is analytically isomorphic to $\mathbb{C}^g/\Lambda_\Omega$. Let $a, b \in \mathbb{Q}^g$, and let ℓ be a positive integer. Let $P \in A(K)/[\ell]A(K)$ and $Q \in A[\ell](K)$ and let $z_P, z_Q \in \mathbb{C}^g$ be such that $\pi(z_P) = P$ and $\pi(z_Q) = Q$ where $\pi : \mathbb{C}^g \to A$ is the natural projection (by abuse of notation we use P, Q to denote the corresponding points of an algebraic and analytic model of A). Suppose that we have chosen z_P, z_Q and z_{P+Q} such that*

$$\frac{\theta\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right](z_P + z_Q)}{\theta\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right](z_P)} \frac{\theta\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right](0)}{\theta\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right](z_Q)} \in K^*, \tag{7}$$

then we have

$$e_T(P, Q) = \frac{\theta\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right](\ell.z_Q + z_P)}{\theta\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right](z_P)} \frac{\theta\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right](0)}{\theta\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right](\ell.z_Q)}. \tag{8}$$

Proof. By Proposition 1, we have

$$f_Q(P - 0) = \frac{\theta\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right](z_P)}{\theta\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right](\ell.z_Q + z_P)} \frac{\theta\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right](\ell.z_Q)}{\theta\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right](0)} \left(\frac{\theta\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right](z_P + z_Q)}{\theta\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right](z_P)} \frac{\theta\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right](0)}{\theta\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right](z_Q)} \right)^\ell. \tag{9}$$

Taking care of the fact that $e_T(P, Q)$ has value in $K^*/K^{*\ell}$ we just have to prove that $e_T(P, Q) = f_Q(0 - P)$. The proof follows exactly the same computations as [16, p. 280].

4 Pairing Computations

In this section, we describe a general method to compute Weil or Tate pairings which does not rely on the usual Miller's loop and prove its correctness. We postpone to the next section the analysis of the running time of these algorithms.

Let $n, \ell \in \mathbb{N}$. We suppose that 2 divides n and that ℓ and n are relatively prime. Let A be an abelian variety over $\mathbb{C}$ with period matrix Ω. We represent A as a closed subvariety of $\mathbb{P}^{n^g-1}$ by the way of level n theta functions and we suppose that this embedding is defined over K. Denote by $\widetilde{A}$ the pullback of A via the natural projection $\kappa : \mathbb{A}^{n^g} \to \mathbb{P}^{n^g-1}$. In the following, we adopt the following convention: if P is a point of A, we denote by $\widetilde{P}$ an affine lift of P that is a point $\widetilde{P}$ of $\mathbb{A}^{n^g}$ such that $\kappa(\widetilde{P}) = P$.

An important ingredient of our algorithm is the Riemann addition formulas. The usual form of these formulas works for theta functions of level divisible by 4 (see for instance [6, p. 139]). In this paper we need a slight generalisation of these formulas for working also with level 2 theta functions. We recall that following the convention for the notation of theta functions described at the end of the introduction, we let for all $i \in Z(\overline{n})$, $z \in \mathbb{C}^g$, $\theta_i(z) = \theta \left[\begin{smallmatrix} 0 \\ i/n \end{smallmatrix} \right] (z, \Omega/n)$. Moreover, we recall that in the following we consider $Z(\overline{n})$ (resp. $Z(\overline{2})$) as a subgroup of $Z(\overline{2n})$ via the map $x \mapsto 2x$ (resp. $x \mapsto nx$).

Theorem 1. *Let $i, j, k, l \in Z(\overline{2n})$. We suppose that $i+j$, $i+k$ and $i+l \in Z(\overline{n})$. Let $\hat{Z}(\overline{2})$ be the dual group of $Z(\overline{2})$. For all $\chi \in \hat{Z}(\overline{2})$ and $z_1, z_2 \in \mathbb{C}^g$ we have*

$$\left(\sum_{\eta \in Z(\overline{2})} \chi(\eta) \theta_{i+j+\eta}(z_1+z_2) \theta_{i-j+\eta}(z_1-z_2) \right) \left(\sum_{\eta \in Z(\overline{2})} \chi(\eta) \theta_{k+l+\eta}(0) \theta_{k-l+\eta}(0) \right)$$
$$= \left(\sum_{\eta \in Z(\overline{2})} \chi(\eta) \theta_{i+k+\eta}(z_1) \theta_{i-k+\eta}(z_1) \right) \left(\sum_{\eta \in Z(\overline{2})} \chi(\eta) \theta_{j+l+\eta}(z_2) \theta_{j-l+\eta}(z_2) \right)$$

Proof. For $i \in Z(\overline{2n})$ and $z \in \mathbb{C}^g$, we let $\theta'_i(z) = \theta \left[\begin{smallmatrix} 0 \\ i/(2n) \end{smallmatrix} \right] (z, \Omega/(2n))$. Let $i, j \in Z(\overline{2n})$ be such that $i+j \in Z(\overline{n})$ and let $z_1, z_2 \in \mathbb{C}^g$. The usual duplication formula [6, p. 139] gives $\theta_{i+j}(z_1+z_2)\theta_{i-j}(z_1-z_2) = \frac{1}{2^g} \sum_{\eta \in Z(\overline{2})} \theta'_{i+\eta}(z_1)\theta'_{j+\eta}(z_2)$. For $\chi \in \hat{Z}(\overline{2})$, using this formula, we compute

$$\sum_{\eta \in Z(\overline{2})} \chi(\eta) \theta_{i+j+\eta}(z_1+z_2) \theta_{i-j+\eta}(z_1-z_2) = \frac{1}{2^g} \sum_{\eta_1, \eta_2 \in Z(\overline{2})} \chi(\eta_1+\eta_2) \theta'_{i+\eta_1}(z_1) \theta'_{j+\eta_2}(z_2)$$
$$= \frac{1}{2^g} \left(\sum_{\eta \in Z(\overline{2})} \chi(\eta) \theta'_{i+\eta}(z_1) \right) \left(\sum_{\eta \in Z(\overline{2})} \chi(\eta) \theta'_{j+\eta}(z_2) \right). \quad (10)$$

Using this last equation to compute the left and right hand sides of the preceding equation we obtain the result.

We suppose that the theta null point $\widetilde{0} = (\theta_i(0))_{i \in Z(\overline{n})}$ is known. We deduce immediately from Theorem 1 an algorithm that takes as inputs $\widetilde{P} = (\widetilde{P}_i)_{i \in Z(\overline{n})}$, $\widetilde{Q} = (\widetilde{Q}_i)_{i \in Z(\overline{n})}$ and $\widetilde{P-Q} = ((\widetilde{P-Q})_i)_{i \in Z(\overline{n})}$ and outputs $\widetilde{P+Q} = ((\widetilde{P+Q})_i)_{i \in Z(\overline{n})}$. We write $\widetilde{P+Q} = \mathrm{PseudoAdd}(\widetilde{P}, \widetilde{Q}, \widetilde{P-Q})$. Indeed we will see later (Proposition 3) that if $n = 4$, we can recover the projective point $P+Q$ from P and Q

using the Riemann addition formulas. It is then easy to see that if we moreover know $\widetilde{P}, \widetilde{Q}$ and $\widetilde{P-Q}$, then there is a unique affine point $\widetilde{P+Q}$ above $P+Q$ that satisfy the addition formulas from Theorem 1. If $n=2$, the point $\widetilde{P+Q}$ is also unique provided the abelian variety satisfies the generic condition from Theorem 3.

Chaining the algorithm PseudoAdd in a classical Montgomery ladder [2, alg. 9.5 p. 148] yields an algorithm that takes as inputs $\widetilde{Q} = (\widetilde{Q}_i)_{i\in Z(\overline{n})}$, $\widetilde{P+Q} = ((\widetilde{P+Q})_i)_{i\in Z(\overline{n})}$, $\widetilde{P} = (\widetilde{P}_i)_{i\in Z(\overline{n})}$, $\widetilde{0} = (\widetilde{0}_i)_{i\in Z(\overline{n})}$ and an integer ℓ and outputs $\widetilde{P+\ell Q}$. We write $\widetilde{P+\ell Q} = \text{ScalarMult}(\widetilde{P+Q}, \widetilde{Q}, \widetilde{P}, \widetilde{0}, \ell)$. In particular, we have $\ell\widetilde{P} = \text{ScalarMult}(\widetilde{P}, \widetilde{P}, \widetilde{0}, \widetilde{0}, \ell)$. The following lemma tells that the output of ScalarMult does not depend on the particular chain of PseudoAdd calls it uses.

Lemma 2. *Let $L = \{0, 1, \ldots, \ell\}$ be a Lucas sequence. Let $A_0 = \widetilde{P}$, $B_0 = \widetilde{0}$, $A_1 = \widetilde{P+Q}$ and $B_1 = \widetilde{Q}$. For $m \in L, m \geqslant 2$, write $m = j+k$ with $j, k, j-k \in L$. Let $B_m = \text{PseudoAdd}(B_j, B_k, B_{j-k})$ and $A_m = \text{PseudoAdd}(A_j, B_k, A_{j-k})$. Then $A_\ell = \widetilde{P+\ell Q}$. In other words $\widetilde{P+\ell Q}$ does not depend on the Lucas sequence used to compute it.*

Proof. If there exist $z_P, z_Q \in \mathbb{C}^g$ such that $\widetilde{P} = (\theta_i(z_P))_{i\in Z(\overline{n})}$, $\widetilde{Q} = (\theta_i(z_Q))_{i\in Z(\overline{n})}$ and $\widetilde{P+Q} = (\theta_i(z_P+z_Q))_{i\in Z(\overline{n})}$ then by Theorem 1 and a recursion we see that $A_j = (\theta_i(z_P + jz_Q))_{i\in Z(\overline{n})}$ and $B_j = (\theta_i(jz_Q))_{i\in Z(\overline{n})}$. Hence $A_\ell = (\theta_i(z_P + \ell z_Q)) = \widetilde{P+\ell Q}$.

Otherwise there exist λ_P, λ_Q and λ_{P+Q} in $\mathbb{C}^*$ such that $\widetilde{P} = \lambda_P(\theta_i(z_P))_{i\in Z(\overline{n})}$, $\widetilde{Q} = \lambda_Q(\theta_i(z_Q))_{i\in Z(\overline{n})}$ and $\widetilde{P+Q} = \lambda_{P+Q}(\theta_i(z_P+z_Q))_{i\in Z(\overline{n})}$. Since we have $\text{PseudoAdd}(\lambda_{P+Q}\widetilde{P+Q}, \lambda_Q\widetilde{Q}, \lambda_P\widetilde{P}) = \frac{\lambda_{P+Q}^2\lambda_Q^2}{\lambda_P}\text{PseudoAdd}(\widetilde{P+Q}, \widetilde{Q}, \widetilde{P})$, an easy recursion shows that $B_j = \lambda_Q^{j^2}(\theta_i(jz_Q))_{i\in Z(\overline{n})}$ and $A_j = \lambda_{P+Q}^j\lambda_Q^{j(j-1)}/\lambda_P^{j-1} \cdot (\theta_i(z_P + jz_Q))_{i\in Z(\overline{n})}$. Hence $A_\ell = \lambda_{P+Q}^\ell\lambda_Q^{\ell(\ell-1)}/\lambda_P^{\ell-1} \cdot (\theta_j(z_P+\ell z_Q))_{j\in Z(\overline{n})} = \widetilde{P+\ell Q}$.

Remark 3. There is a natural action of $\overline{K}^*$ on $\mathbb{A}^{n^g} - \{0\}$ by multiplication of the coordinates of a point that we denote by $\alpha * \widetilde{P}$ for $\alpha \in \overline{K}^*$ and $\widetilde{P} \in \mathbb{A}^{n^g}(\overline{K})$. In the proof of the preceding lemma we have seen the effect of this action on the output of the algorithm ScalarMult: let $P, Q \in A(\overline{K})$ and let $\widetilde{P}, \widetilde{Q}, \widetilde{P+Q}$ be affine lifts of P, Q and $P+Q$. Let $\widetilde{R} = \text{ScalarMult}(\widetilde{P+Q}, \widetilde{Q}, \widetilde{P}, \widetilde{0}, \ell)$. Let $\alpha, \beta, \gamma, \delta \in \overline{K}$, we have

$$\text{ScalarMult}(\alpha * \widetilde{P+Q}, \beta * \widetilde{Q}, \gamma * \widetilde{P}, \delta * \widetilde{0}, \ell) = (\alpha^\ell\beta^{\ell(\ell-1)}/\gamma^{\ell-1}\delta^{\ell(\ell-1)}) * \widetilde{R}, \quad (11)$$

$$\text{ScalarMult}(\alpha * \widetilde{P}, \alpha * \widetilde{P}, \delta * \widetilde{0}, \delta * \widetilde{0}, \ell) = \frac{\alpha^{\ell^2}}{\delta^{\ell^2-1}} * \text{ScalarMult}(\widetilde{P}, \widetilde{P}, \widetilde{0}, \widetilde{0}, \ell). \quad (12)$$

Given P and Q with projective coordinates $(\theta_i(z_P))_{i\in Z(\overline{n})}$ and $(\theta_i(z_Q))_{i\in Z(\overline{n})}$ for $z_P, z_Q \in \mathbb{C}^g$, we would like to compute $e_W(P,Q)$ and $e_T(P,Q)$.

We can state the main theorem of this section

Theorem 2. *We suppose that n and ℓ are relatively prime. For $X, Y \in A(\overline{K})$, denote by $\widetilde{X}$, $\widetilde{Y}$, $\widetilde{X+Y}$ any affine lifts of X, Y and $X+Y$. Recall that for $i \in Z(\overline{n})$, we denote by $\widetilde{X}_i$ the coordinate i of the point $\widetilde{X}$. For $\ell \in \mathbb{N}$ and $i \in Z(\overline{n})$, let $f_T(\widetilde{X},\widetilde{Y},\widetilde{X+Y},\widetilde{0},\ell,i) = \frac{\mathrm{ScalarMult}(\widetilde{X+Y},\widetilde{X},\widetilde{Y},\widetilde{0},\ell)_i}{\mathrm{ScalarMult}(\widetilde{X},\widetilde{X},\widetilde{0},\widetilde{0},\ell)_i}\frac{\widetilde{0}_i}{\widetilde{Y}_i}$. Then for $P, Q \in A[\ell]$ and $i \in Z(\overline{n})$, we have:*

$$e_W(P,Q)^n = f_T(\widetilde{P},\widetilde{Q},\widetilde{P+Q},\widetilde{0},\ell,i)^{-1} f_T(\widetilde{Q},\widetilde{P},\widetilde{P+Q},\widetilde{0},\ell,i), \tag{13}$$

whenever the right hand side is well defined.

Moreover, for $P \in A(K)/[\ell]A(K)$, $Q \in A[\ell]$, if we suppose that $\widetilde{0}$, $\widetilde{P}$, $\widetilde{Q}$ and $\widetilde{P+Q}$ are affine lifts of 0, P, Q and $P+Q$ with coordinates in K, then we have for $i \in Z(\overline{n})$,

$$e_T(P,Q)^n = f_T(\widetilde{Q},\widetilde{P},\widetilde{P+Q},\widetilde{0},\ell,i), \tag{14}$$

whenever the right hand side is well defined.

Proof. Let $z_P, z_Q \in \mathbb{C}^g$ such that $\pi(z_P) = P$ and $\pi(z_Q) = Q$ (recall that $\pi : \mathbb{C}^g \to A = \mathbb{C}^g/\Lambda_\Omega$ is the natural projection). Let $\widetilde{P} = (\theta_i(z_P))_{i\in Z(\overline{n})}$, $\widetilde{Q} = (\theta_i(z_Q))_{i\in Z(\overline{n})}$ and $\widetilde{P+Q} = (\theta_i(z_P+z_Q))_{i\in Z(\overline{n})}$. Then applying Corollary 2, if $P, Q \in A[\ell]$, we obtain that

$$e_{\Omega/n}(z_P,z_Q)^\ell = e_W(P,Q)^n = f_T(\widetilde{P},\widetilde{Q},\widetilde{P+Q},\widetilde{0},\ell,i)^{-1} f_T(\widetilde{Q},\widetilde{P},\widetilde{P+Q},\widetilde{0},\ell,i).$$

In the same way, by Proposition 2 (which apply for $i = 0$, but it is easy to see that the same result is true for any $i \in Z(\overline{n})$), we have for $P \in A(K)/[\ell]A(K)$ and $Q \in A[\ell]$, $e_T(P,Q)^n = f_T(\widetilde{Q},\widetilde{P},\widetilde{P+Q},\widetilde{0},\ell,i)$. Next, let $\alpha,\beta,\gamma,\delta \in \overline{K}$. By Remark 3, we have

$$f_T(\alpha*\widetilde{X},\beta*\widetilde{Y},\gamma*\widetilde{X+Y},\delta*\widetilde{0},\ell,i) = \frac{\gamma^\ell\delta^\ell}{\alpha^\ell\beta^\ell} f_T(\widetilde{X},\widetilde{Y},\widetilde{X+Y},\widetilde{0},\ell,i). \tag{15}$$

This shows that the expressions (13) and (14) for the Weil and Tate pairing do not depend on the choice of affine liftings (rational over K in the case of the Tate pairing) of P, Q and $P+Q$.

Remark 4. In this remark we keep the notations of the previous theorem. Let $\mathscr{L}$ be a polarization of A associated to Ξ_n for $n \in \mathbb{N}^*$ which is rational over K. Let $(\theta'_i)_{i\in Z(\overline{n})}$ be a basis of global sections of a trivialisation of $\pi^*(\mathscr{L})$ (and we rigidify this basis by setting $\theta'_0(0) = 1$). In general, it is not true that the polarization defined by the level n classical theta functions is rational over K. Nonetheless we know that there exits a non vanishing function ζ of $\mathbb{C}^g$ such that $\theta_i = \zeta\theta'_i$ for $i \in Z(\overline{n})$ (up to a renumbering of the basis θ'_i).

Let $0, z_P, z_Q, z_{P+Q} \in \mathbb{C}^g$. For $z_X \in \{0, z_P, z_Q, z_{P+Q}\}$, if we denote by $\widetilde{X}^{\mathrm{alg}} = (\theta'_i(z_X))_{i\in Z(\overline{n})}$, then there exist constant factors $c_X \in \mathbb{C}^*$ such that for $X \in \{0, P, Q, P+Q\}$ we have $c_X * \widetilde{X}^{\mathrm{alg}} = \widetilde{X}$.

As we can suppose that the coordinates of the points $\widetilde{X}^{\mathrm{alg}}$ for $X \in \{0, P, Q, P+Q\}$ are defined over K, we can rewrite (9) as:

$$e_T(P,Q) = \left(\frac{c_{P+Q}c_0}{c_P c_Q}\right)^{-\ell} \frac{\theta_i(\ell.z_Q + z_P)}{\theta_i(z_P)} \frac{\theta_i(0)}{\theta_i(\ell.z_Q)},$$

for $i \in Z(\overline{n})$. But by (15) we have the equation: $f_T(\widetilde{Q}^{\mathrm{alg}}, \widetilde{P}^{\mathrm{alg}}, \widetilde{P+Q}^{\mathrm{alg}}, \widetilde{0}^{\mathrm{alg}}, \ell, i) = \left(\frac{c_{P+Q}c_0}{c_P c_Q}\right)^{-\ell} . f_T(\widetilde{Q}, \widetilde{P}, \widetilde{P+Q}, \widetilde{0}, \ell, i) = \left(\frac{c_{P+Q}c_0}{c_P c_Q}\right)^{-\ell} \frac{\theta_i(\ell.z_Q+z_P)}{\theta_i(z_P)} \frac{\theta_i(0)}{\theta_i(\ell.z_Q)}$. Comparing these formulas, we obtain that we can compute the Tate pairing by taking affine lifts of 0, P, Q and $P+Q$ provided by the coordinates θ'_i. Now using (15) again, we obtain that to compute the Tate pairing we only have to choose affine lifts of 0, P, Q, and $P+Q$ which are rational over K.

As we have shown that the formulas of Theorem 2 do not depend on a choice of the affine lifts of the input points of the algorithm (as long as the choices are the same for the computation of the two functions f_T in the case of the Weil pairing), from now on we only consider projective points.

In order to have a working algorithm to compute Weil and Tate pairings, it remains to explain how to compute $P+Q$ from the knowledge of P and Q. As the formulas to compute the pairings only involve one of the level n theta functions, and since the number of the coordinates used in the computation of ScalarMult is n^g, for the sake of efficiency it is important to have a small n. As 2 divides n, from now on, we focus on the two interesting cases: $n = 2$ and $n = 4$.

We first treat the case $n = 4$. Let $z_P, z_Q \in \mathbb{C}^g$ and let $P = (P_i)_{i\in Z(\overline{n})} = (\theta_i(z_P))_{i\in Z(\overline{n})}$ and $Q = (Q_i)_{i\in Z(\overline{n})} = (\theta_i(z_Q))_{i\in Z(\overline{n})}$. From the knowledge of P and Q, with the addition formula (Theorem 1), one can compute the products:

$$\Big(\sum_{\eta\in Z(\overline{2})} \chi(\eta)\theta_{i+j+\eta}(z_P+z_Q)\theta_{i-j+\eta}(z_P-z_Q)\Big)\Big(\sum_{\eta\in Z(\overline{2})} \chi(\eta)\theta_{k+l+\eta}(0)\theta_{k-l+\eta}(0)\Big), \quad (16)$$

for $\chi \in \hat{Z}(\overline{2})$ and $i, j, k, l \in Z(\overline{2n})$ such that $i+j$, $i+k$, and $i+l \in Z(\overline{n})$. If we can prove that for any such choice of $i, j, k, l \in Z(\overline{2n})$ and $\chi \in \hat{Z}(\overline{2})$ there exist $k' \in k + Z(\overline{n})$ and $l' \in l + Z(\overline{n})$ such that $\sum_{\eta\in Z(\overline{2})} \chi(\eta)\theta_{k'+l'+\eta}(0)\theta_{k'-l'+\eta}(0) \neq 0$, then by summing over the characters the left bracket of (16) one can compute all the products $\theta_i(z_P+z_Q)\theta_j(z_P-z_Q)$, for $i, j \in Z(\overline{n})$ from which it is easy to recover by taking quotients the projective point $(\theta_i(z_P+z_Q))_{i\in Z(\overline{n})}$.

Now, using equation (10), we have

$$\sum_{\eta\in Z(\overline{2})} \chi(\eta)\theta_{k+l+\eta}(0)\theta_{k-l+\eta}(0) = \frac{1}{2^g}\Big(\sum_{\eta\in Z(\overline{2})} \chi(\eta)\theta'_{k+\eta}(0)\Big)\Big(\sum_{\eta\in Z(\overline{2})} \chi(\eta)\theta'_{l+\eta}(0)\Big), \quad (17)$$

where for $k \in Z(\overline{8})$, $\theta'_k(z) = \theta\left[\begin{smallmatrix}0\\ k/8\end{smallmatrix}\right](z, \Omega/8)$. We have the

Proposition 3. *Let $\delta \in \mathbb{N}$ be such that 4 divides δ. For any $a \in K(\overline{2\delta})$ there exists an element $b \in a + K(\delta)$ such that for all $\chi \in \hat{Z}(\overline{2})$ we have that $\sum_{\eta \in Z(\overline{2})} \chi(\eta)\theta \left[{}^{0}_{(b+\eta)/(2\delta)} \right](0, 1/(2\delta).\Omega) \neq 0$.*

Proof. This is just a rephrasing of [11, equation (*) p. 339].

Applying the preceding proposition to the factors of the right hand of equation (17), we obtain that there exists $k' \in k + Z(\overline{n})$ and $l' \in l + Z(\overline{n})$ such that $\sum_{\eta \in Z(\overline{2})} \chi(\eta)\theta_{k'+l'+\eta}(0)\theta_{k'-l'+\eta}(0) \neq 0$ and we are done.

In the case $n = 2$, as usual, for all $i \in Z(\overline{2})$, we put $\theta_i(z) = \theta \left[{}^{0}_{i/2} \right](z, 1/2.\Omega)$. Then by Theorem 1, we have for any $\chi \in \hat{Z}(\overline{2})$ and for well chosen pairs of quadruples $(i, j, k, l), (i', j', k', l') \in Z(\overline{2})^4$ an equation

$$\begin{aligned} &\Big(\sum_{\eta \in Z(\overline{2})} \chi(\eta)\theta_{i+\eta}(z_P + z_Q)\theta_{j+\eta}(z_P - z_Q)\Big)\Big(\sum_{\eta \in Z(\overline{2})} \chi(\eta)\theta_{k+\eta}(0)\theta_{l+\eta}(0)\Big) \\ &= \Big(\sum_{\eta \in Z(\overline{2})} \chi(\eta)\theta_{i'+\eta}(z_P)\theta_{j'+\eta}(z_P)\Big)\Big(\sum_{\eta \in Z(\overline{2})} \chi(\eta)\theta_{k'+\eta}(z_Q)\theta_{l'+\eta}(z_Q)\Big). \end{aligned} \tag{18}$$

If the kernel of χ does not contain the subgroup of $Z(\overline{2})$ generated by $k + l$ then we have $\sum_{\eta \in Z(\overline{2})} \chi(\eta)\theta_{k+\eta}(0)\theta_{l+\eta}(0) = 0$, so it is not possible to recover $\theta_{i+\eta}(z_P + z_Q)$ as before. This is consistent with the fact that for $i \in Z(\overline{2})$ and $z \in \mathbb{C}^g$, $\theta_i(z) = \theta_i(-z)$, the right hand side of (18) is invariant for the transformation $z_Q \mapsto -z_Q$ while it is not the case of the left hand side. The best we can hope is that for almost all period matrices $\Omega \in \mathbb{H}_g$ there exists a $k \in Z(\overline{2})$ such that for all $l \in Z(\overline{2})$ and $\chi \in \hat{Z}(\overline{2})$ such that $k + l$ is in the kernel of χ, we have $\sum_{\eta \in Z(\overline{2})} \chi(\eta)\theta_{k+\eta}(0)\theta_{l+\eta}(0) \neq 0$. This is exactly the content of Theorem 3. In order to prove this theorem, we let $T_{k,l,\chi} = \sum_{\eta \in Z(\overline{2})} \chi(\eta)\theta_{k+\eta}(0)\theta_{l+\eta}(0)$ and we state the following lemma:

Lemma 3. *For $\Omega \in \mathbb{H}_g$, the two following properties are equivalent:*

1. *There exists a $k \in Z(\overline{2})$ such that for all $\ell \in Z(\overline{2})$ and $\chi \in \hat{Z}(\overline{2})$ such that $k + l$ is in the kernel of χ, we have $T_{k,l,\chi} \neq 0$.*
2. *For all $i, j \in Z(\overline{2})$ such that ${}^t i.j = 0$, $\theta_{i,j}(0) \neq 0$.*

Proof. For $\chi \in \hat{Z}(\overline{2})$, let $\mu \in Z(\overline{2})$ be such that $\chi(\eta) = (-1)^{{}^t \eta.\mu}$. Let $\rho : Z(\overline{4}) \to Z(\overline{2})$, $x \mapsto x \mod Z(\overline{2})$ be the canonical projection. Then we have (see [14, prop 1.3 p. 124]), for all $i \in Z(\overline{4})$ $\sum_{\eta \in Z(\overline{2})} \chi(\eta)\theta'_{i+\eta}(0) = 2^g.\theta_{\mu,\rho(i)}(0)$, where $\theta'_k(z) = \theta \left[{}^{0}_{k/4} \right](z, 1/4.\Omega)$. Combining this relation together with (17), for all $i, j \in Z(\overline{4})$ such that $i + j \in Z(\overline{2})$, let $k = i + j$, $l = i - j$, we obtain the equality

$$T_{k,l,\chi} = T_{i+j,i-j,\chi} = 2^g.\theta_{\mu,\rho(i)}(0)\theta_{\mu,\rho(j)}(0) = 2^g.\theta_{\mu,k+l}(0)^2. \tag{19}$$

Since $\chi(k + l) = (-1)^{{}^t(k+l).\mu}$ the lemma follows immediately from (19).

It is well known that for $z \in \mathbb{C}^g$, and $k, l \in Z(\overline{2})$, we have $\theta_{k,l}(-z) = (-1)^{{}^t k.l}\theta_{k,l}(z)$. As a consequence, for all $k, l \in Z(\overline{2})$ such that ${}^t k.l = 1$ (the odd characteristics), we have $\theta_{k,l}(0) = 0$. Denote by $\mathcal{M}_4$ the quasi-projective variety over $\mathbb{C}$ defined as the locus of zeros of $\theta_{i,j}(0)$ considered as functions of Ω. It is clear that $\mathcal{M}_4$ parametrizes the set of principally polarized abelian varieties together with a level 4 structure since from the knowledge of a point in $\mathcal{M}_4$ one can recover the projective embedding of the corresponding abelian variety provided by the Riemann equations.

Theorem 3. *For all $k, l \in Z(\overline{2})$ such that ${}^t k.l = 0$, the function $\theta_{k,l}(0)$ on $\mathcal{M}_4$ is non-trivial and as consequence, its zero locus is a proper subvariety of $\mathcal{M}_4$ of codimension* 1.

Proof. We sketch the proof of the theorem. Suppose on the contrary that for $k, l \in Z(\overline{2})$ such that ${}^t k.l = 0$, $\theta_{k,l}(0)$ is a constant function of Ω. This is a degree 1 relation for level 4 theta constants, call it $R_{k,l}$. We have for all $k \in Z(\overline{4})$, $\theta_k(0) = \theta \left[\begin{smallmatrix} 0 \\ (2k)/8 \end{smallmatrix} \right] (0, (2\Omega)/8)$. Thus, the level 4 degree 1 relations $R_{k,l}$ induce degree 1 relations for level 8 theta constants. The hypothesis ${}^t k.l = 0$ means that these level 8 relations are not a linear combination of the symmetry relations $\theta_k(0) = \theta_{-k}(0)$ for all $k \in Z(\overline{8})$. This is a contradiction with the description of $\mathcal{M}_8$ the modular space of level 8 marked abelian varieties given by Mumford in [12, main th. p. 83] as an open subset of the reduced projective variety given by the symmetry relations and the Riemann relations.

Remark 5. The preceding theorem shows that the symmetric pairing computation algorithms that we describe in the next section works for a general abelian variety. However, one can ask if the closed proper subset of $\mathcal{M}_4$, given by the cancellation of some even level 4 theta constants contains noticeable abelian varieties. Actually, this is the case since a theorem of Frobenius [15, cor. 6.7 p. 3.102] tells us that the locus of Jacobian of hyperelliptic curves inside $\mathcal{M}_4$ can be given by equations of the form $\theta_{k,l}(0) = 0$ where (k, l) is an even characteristic. As a consequence, the algorithms of Section 5.2 to compute symmetric pairings don't apply to Jacobian of hyperelliptic of genus g when $g \geqslant 3$. It should be noted however that following [7, cor 4.5.2 and remark (2)], the condition that for all $k, l \in Z(\overline{2})$ such that ${}^t k.l = 0$, $\theta_{k,l}(0) \neq 0$ is equivalent to the fact the level 2 theta functions give a projectively normal embedding. Considering this result, the condition of Theorem 3 should be considered as natural.

5 Complexity Analysis

In this section, we explain how to use the results of the preceding section to compute efficiently pairings on abelian and Kummer varieties with a special focus on dimension 1 and 2 since these cases are particularly interesting for cryptographic applications.

5.1 Abelian Varieties

We begin with the case of abelian varieties since the main loop of the algorithm can also be used for the computation of symmetric pairings on Kummer varieties.

Initialisation phase. The initialisation phase depends on the representation of the points P and Q on the abelian variety A. If P and Q are given by theta coordinates of level 4 we can apply the procedure described in Section 4 to compute the homogeneous coordinates of $(\theta_i(P+Q))_{i\in Z(\overline{4})}$.

Suppose that another coordinate system is used to represent P and Q that we denote by $(X_i)_{i\in I}$ where X_i are rational functions on a Zariski open subset of A. Then by definition there exist formulas to compute $\theta_i(P)$ and $\theta_i(Q)$ from the knowledge of $X_i(P)$ and $X_i(Q)$. In practise, the dictionary between some useful coordinate system and the theta coordinates can easily be deduced from well known properties of theta functions. It should be remarked that in order to carry out these computations we might have to do a base field extension since in the projective embedding of A provided by the level 4 theta functions the 4-torsion of A is rational over the base field, whereas this may not the case with other models of A. The advantage of the level 4 is that no square root extraction is needed for the computation of $P+Q$, contrarily to the level 2 case as we will see.

From the knowledge of $\theta\left[\begin{smallmatrix}0\\ i/4\end{smallmatrix}\right](z_X, 1/4.\Omega)$, $i\in\mathbb{Z}(\overline{4})$ for $X=P,Q,P+Q$ we can then compute the level 2 coordinates given by $(\sum_{j\in Z(\overline{2})}\theta\left[\begin{smallmatrix}0\\ \frac{i+2j}{4}\end{smallmatrix}\right](z_X,\frac{\Omega}{4}))_{i\in Z(\overline{2})}$ for the coordinates of the (isogeneous) points $X=P,Q,P+Q$.

Pairing computation phase. As we have seen before, we can carry out the computations of the main loop of the algorithm with level 2 theta functions since at the end we only need one theta coordinate to compute the pairings. This is more efficient because we only need 2^g coordinates to represent a point and we can do the computation on the field of definition of the 2-torsion of A.

We suppose that we are given the level 2 coordinates of P, Q, $P+Q$. Rather than considering the formulas of Theorem 1 for the double and add algorithm, we use the level 2 formulas given in [4] for the genus 2 case, and in [5] for the genus 1 case. For instance, let E be an elliptic curve defined by $\Omega\in\mathbb{H}_1$, let $\Omega'=\Omega/2$ and put

$$a=\vartheta\left[\begin{smallmatrix}0\\ 0\end{smallmatrix}\right](0,\Omega');\ b=\vartheta\left[\begin{smallmatrix}0\\ 1/2\end{smallmatrix}\right](0,\Omega');\ A=\vartheta\left[\begin{smallmatrix}0\\ 0\end{smallmatrix}\right](0,2\Omega');\ B=\vartheta\left[\begin{smallmatrix}1/2\\ 0\end{smallmatrix}\right](0,2\Omega').$$

The duplication formulas are given by the equalities:

$$\begin{cases} a\vartheta\left[\begin{smallmatrix}0\\ 0\end{smallmatrix}\right](z,\Omega')=\vartheta\left[\begin{smallmatrix}0\\ 0\end{smallmatrix}\right](z,2\Omega')^2+\vartheta\left[\begin{smallmatrix}1/2\\ 0\end{smallmatrix}\right](z,2\Omega')^2,\\ b\vartheta\left[\begin{smallmatrix}0\\ 1/2\end{smallmatrix}\right](z,\Omega')=\vartheta\left[\begin{smallmatrix}0\\ 0\end{smallmatrix}\right](z,2\Omega')^2-\vartheta\left[\begin{smallmatrix}1/2\\ 0\end{smallmatrix}\right](z,2\Omega')^2.\end{cases}$$

$$\begin{cases} 2A\vartheta\left[\begin{smallmatrix}0\\ 0\end{smallmatrix}\right](2z,2\Omega')=\vartheta\left[\begin{smallmatrix}0\\ 0\end{smallmatrix}\right](z,\Omega')^2+\vartheta\left[\begin{smallmatrix}0\\ 1/2\end{smallmatrix}\right](z,\Omega')^2,\\ 2B\vartheta\left[\begin{smallmatrix}1/2\\ 0\end{smallmatrix}\right](2z,2\Omega')=\vartheta\left[\begin{smallmatrix}0\\ 0\end{smallmatrix}\right](z,\Omega')^2-\vartheta\left[\begin{smallmatrix}0\\ 1/2\end{smallmatrix}\right](z,\Omega')^2.\end{cases}$$

Let $x=\theta\left[\begin{smallmatrix}0\\ 0\end{smallmatrix}\right](z,\Omega')$ and $z=\theta\left[\begin{smallmatrix}0\\ 1/2\end{smallmatrix}\right](z,\Omega')$ using the above formulas yield the following algorithms:

Doubling Algorithm:
Input: A point $P = (x : z)$.
Output: The double $2.P = (x' : z')$.

1. $x_0 = (x^2 + z^2)^2$;
2. $z_0 = \frac{A^2}{B^2}(x^2 - z^2)^2$;
3. $x' = (x_0 + z_0)$;
4. $z' = \frac{a}{b}(x_0 - z_0)$;
5. Return $(x' : z')$.

Differential Addition Algorithm:
Input: Two points $P = (x : z)$ and $Q = (\widetilde{x} : \widetilde{z})$ on E, and $R = (\overline{x} : \overline{z}) = P - Q$, with $\overline{xz} \neq 0$.
Output: The point $P + Q = (x' : z')$.

1. $x_0 = (x^2 + z^2)(\widetilde{x}^2 + \widetilde{z}^2)$;
2. $z_0 = \frac{A^2}{B^2}(x^2 - z^2)(\widetilde{x}^2 - \widetilde{z}^2)$;
3. $x' = (x_0 + z_0)/\overline{x}$;
4. $z' = (x_0 - z_0)/\overline{z}$;
5. Return $(x' : z')$.

Recall that in order to compute the pairing $e_T(P, Q)$, we have to compute $\widetilde{P + \ell Q} = \text{ScalarMult}(\widetilde{P + Q}, \widetilde{Q}, \widetilde{P}, \widetilde{0}, \ell)$ and $\ell\widetilde{Q} = \text{ScalarMult}(\widetilde{Q}, \widetilde{Q}, \widetilde{0}, \widetilde{0}, \ell)$. It should be remarked that in the computation of $\widetilde{P + \ell Q}$, we need exactly the same values of $j.Q$ for some $j \in \{1, \ldots, \ell\}$ as those required to obtain $\ell\widetilde{Q}$. Since we want to avoid a division in each step, we use a Montgomery ladder so that the differences in the adding step are always the same points. To speed up the differential additions, we have renormalised the theta null point (a, b) to $(1, b/a)$. It is easy to see by doing the same computation as in Remark 3 that this does not change the value of the Tate pairing $e_T(P, Q)$. Moreover we also have renormalised the theta null point (A, B). Looking back at the proof of 1, we see that this change each affine addition by the constant factor B^{-2}. This also does not affect the final value of the Tate pairing $e_T(P, Q)$, since we use the same Lucas sequence for computing $\ell\widetilde{Q}$ and $\widetilde{P + \ell Q}$.

This give the following steps for the pairing: from $(j-1)Q$, jQ and $P + jQ$ we compute $2(j-1)Q$, $(2j-1)Q$, $P + (2j-1)Q$ or $(2j-1)Q$, $2jQ$ and $P + 2jQ$ depending on the binary decomposition of ℓ. We remark that at each step we do a doubling and two adding, and that we add the same point to the triple $(j-1)Q, jQ, P+jQ$. For instance in genus 1, we only have to compute $\frac{A^2}{B^2}(x^2-z^2)$ once, where $(x : z)$ are the coordinates of the doubled point.

The figure below summarises the cost per bit of computation of the Tate pairing with our algorithm in genus 1 and 2 with the following notations: S is for squaring, M is for general multiplication, m is for multiplication by a constant.

Tate pairing	First pairing $e(P, Q)$	Following pairings $e(P', Q)$
Dimension 1	8S+4m+4M	2S+1m+2M
Dimension 2	13S+12m+11M	4S+3m+4M

The algorithms that we have presented in this section are deterministic and generalize immediately to the higher dimension case. Usually when computing a pairing, the field of definition of Q has a smaller degree than the field of definition of P, so that at each step one adding and one doubling is done with points in the smaller field. We also remark that if we have to compute several pairings $e(P_1, Q)$, $e(P_2, Q)$, ... with the same Q, it makes sense to store the

results of the computations of the jQ so that for the next pairings we only have to compute the $P_i + jQ$. For instance when $g = 1$ if we store the $\log_2(\ell)$ coordinates $(x^2 + z^2, \frac{A^2}{B^2}(x^2 - z^2))$ of each doubling step, we can compute the subsequent pairings with only five multiplications at each step.

5.2 Kummer Varieties

Let A be a principally polarized abelian variety of dimension g defined by $\Omega \in \mathbb{H}_g$. As we have seen in the introduction, the level 2 theta functions defined by Ω give a projective embedding of the Kummer variety associated to a A. We recall that the Kummer variety $\mathscr{K}_A$ of A is the quotient of A by the action of the automorphism -1 of A. Let $\zeta : A \to \mathscr{K}_A$ be the natural projection. In the following, if $P \in A(\overline{K})$ we denote by $\overline{P}$ its image by ζ. The construction of $\mathscr{K}_A$ does not preserve the group structure of A. Nonetheless, we remark that from the data of $\overline{P} \in \mathscr{K}_A(K)$ one can compute $2\overline{P}$ without ambiguity, and from the data of $\overline{P}$, $\overline{Q}$ and $\overline{P - Q}$ one can compute $\overline{P + Q}$. As a consequence, $\mathscr{K}_A$ inherits from A of an action of $\mathbb{Z}$ on its points which can be computed by a Montgomery ladder like algorithm.

Let e be a pairing on A, and let $\overline{K}_0^*$ be the quotient of $\overline{K}^*$ by the action of the automorphism -1. Let $\zeta_0 : \overline{K}^* \to \overline{K}_0^*$ be the natural projection. The pairing e gives a well defined application $\overline{e} : \mathscr{K}_A(\overline{K}) \times \mathscr{K}_A(\overline{K}) \to \overline{K}_0^*$, $(\overline{P}, \overline{Q}) \mapsto \zeta_0(e(P, Q))$. It is easily seen that the elements of $\overline{K}_0^*$ are in bijection with the set $S = \{x + 1/x, x \in \overline{K}^*\}$. Identifying $\overline{K}_0^*$ with S, the application ζ_0 is given by $\zeta_0(x) = x + 1/x$, $x \in \overline{K}^*$ from which we deduce the expression of $\overline{e} : (\overline{P}, \overline{Q}) \mapsto e(P, Q) + e(-P, Q)$. This pairing has been introduced in [3]. In the following, if e is a pairing, we say that $\overline{e}$ is the symmetric pairing associated to e. The symmetric pairing $\overline{e}$ can be seen as a version of e for compressed coordinates as it takes as input points with 2^g coordinates rather than 4^g.

Its cryptographic relevance comes from the compatibility of $\overline{e}$ with the $\mathbb{Z}$-set structures of $\mathscr{K}_A$ and $\overline{K}_0^*$: for all $\lambda, \mu \in \mathbb{Z}$, $\overline{P}, \overline{Q} \in \mathscr{K}_A$, we have $\overline{e}(\lambda.\overline{P}, \mu.\overline{Q}) = (\lambda\mu).\overline{e}(\overline{P}, \overline{Q})$. In [3], the authors give an algorithm based on Lucas sequences to compute the action of $\mathbb{Z}$ on $\overline{K}_0$ for certain finite fields. Here we would like to emphasize that the compatibility of the $\mathbb{Z}$-structure of $\mathscr{K}_A$ and $\overline{K}_0$ is also algorithmic. It comes from the fact and on any quotient of an algebraic group by the automorphism -1 there exists a natural Montgomery ladder algorithm to compute the resulting $\mathbb{Z}$-action. In the case of $\overline{K}_0$ we obtain very simple and general formulas. For $x \in \overline{K}$, and $i, j \in \mathbb{Z}$, we have

$$(x^i + \frac{1}{x^i})^2 = (x^{2i} + \frac{1}{x^{2i}} + 2); \quad (x^i + \frac{1}{x^i})(x^j + \frac{1}{x^j}) = (x^{i+j} + \frac{1}{x^{i+j}}) + (x^{i-j} + \frac{1}{x^{i-j}}).$$

We have seen that the codomain of the Tate pairing e_T is the multiplicative group $K^*/K^{*\ell}$. Again, we can take the quotient of this group by the action of (-1) on it, denote it by $(K^*/K^{*\ell})_0$. It is clear that there is a bijection between the set $(K^*/K^{*\ell})_0$ and the set $S_T = \{x + 1/x, x \in K_T\}$ where K_T is a set of representatives of $K^*/K^{*\ell}$. Moreover, one can compute the $\mathbb{Z}$-action on such representatives using the preceding algorithm.

Initialisation phase. We suppose that we know the level 2 coordinates $\theta_i(z_P)$ and $\theta_i(z_Q)$, $i \in Z(\overline{2})$ of P and Q. We may assume (by multiplying by a projective factor) that the values of the projective coordinates $(\theta_i(z_P))_{i\in Z(\overline{2})}$ and $(\theta_i(z_Q))_{i\in Z(\overline{2})}$ are in K. Using Theorem 1 and Theorem 3, we obtain that for a general choice of $\mathscr{K}_A$, it is possible to compute for all $i,j \in Z(\overline{2})$ and $\chi \in \hat{Z}(\overline{2})$ such that $\chi(i-j)=1$, $\sum_{\eta\in Z(\overline{2})}\chi(\eta)\theta_{i+\eta}(z_P+z_Q)\theta_{j+\eta}(z_P+z_Q)$ from the inputs. By summing over the characters, we obtain for all $i,j \in Z(\overline{2})$

$$\kappa_{ij} = \theta_i(z_P+z_Q)\theta_j(z_P-z_Q) + \theta_j(z_P+z_Q)\theta_i(z_P-z_Q). \tag{20}$$

We suppose that $\theta_0(z_P+z_Q)\theta_0(z_P-z_Q)\neq 0$, if necessary by replacing the index 0 by another one. By rescaling the projective coordinates, we do our computations as if $\theta_0(z_P-z_Q)=1$ hence we know $\theta_0(z_P+z_Q)$.

For $i \in Z(\overline{2})$, let $\mathfrak{P}_i(X) = X^2 - 2\frac{\kappa_{i0}}{\kappa_{00}}X + \frac{\kappa_{ii}}{\kappa_{00}}$. The roots of $\mathfrak{P}_i(X)$ are $\frac{\theta_i(z_P+z_Q)}{\theta_0(z_P+z_Q)}$, $\frac{\theta_i(z_P-z_Q)}{\theta_0(z_P-z_Q)}$. If P or Q is a point of 2-torsion, $\overline{P+Q}=\overline{P-Q}\in\mathscr{K}_A$ so each $\mathfrak{P}_i(X)$ has a double root. Otherwise, we may suppose that there exist $\alpha\in Z(\overline{2})$, $\alpha\neq 0$ such that the matrix $M=\begin{pmatrix}\theta_0(z_P+z_Q) & \theta_0(z_P-z_Q)\\ \theta_\alpha(z_P+z_Q) & \theta_\alpha(z_P-z_Q)\end{pmatrix}$ is invertible.

We can compute $\{\theta_\alpha(z_P+z_Q),\theta_\alpha(z_P-z_Q)\}$ by finding the roots of $\mathfrak{P}_\alpha(X)$. As by hypothesis, $P+Q, P-Q\in A(K)$, we deduce that these roots are in K. We fix an arbitrary ordering $(\theta_\alpha(z_P+z_Q),\theta_\alpha(z_P-z_Q))$ of these roots (depending on the ordering, we will compute $\overline{P-Q}$ or $\overline{P+Q}$).

We can then find $\{\theta_i(z_P+z_Q),\theta_i(z_P-z_Q)\}$ by solving the system

$$\begin{pmatrix}\theta_0(z_P+z_Q) & \theta_0(z_P-z_Q)\\ \theta_\alpha(z_P+z_Q) & \theta_\alpha(z_P-z_Q)\end{pmatrix}\begin{pmatrix}\theta_i(z_P-z_Q)\\ \theta_i(z_P+z_Q)\end{pmatrix}=\begin{pmatrix}\kappa_{i0}\\ \kappa_{i\alpha}\end{pmatrix}. \tag{21}$$

This method requires one square root.

Pairing computation phase. Let $P\in A(K)/[\ell]A(K)$ and $Q\in A[\ell]$ and denote by $\overline{P},\overline{Q}$ the corresponding points on $\mathscr{K}_A$. Denote by $\theta_i(z)$, $i\in Z(\overline{2})$, the level 2 theta functions associated to Ω. We present two methods to compute the symmetric Tate pairing.

A first method is to consider the formula $\overline{e}_T(\overline{P},\overline{Q}) = e_T(P,Q)+e_T(P,-Q)$. We have explained in the last paragraph how to compute the set $S=\{\overline{P+Q},\overline{P-Q}\}$ at the expence of a square root extraction. By choosing a point in S, we can use the algorithm from Section 5.1 to compute $e(P,Q)$ (resp $e(P,-Q)$). We can then compute $\overline{e}_T(P,Q)=e(P,Q)+e(P,-Q)$ with a simple division.

Another approach is to work in the algebra $\mathscr{A}=K[X]/(\mathfrak{P}_\alpha(X))$ for $\alpha\in Z(\overline{2})$ as before. We denote by g the unique automorphism of the algebra of $\mathscr{A}$ leaving K invariant and different from the identity. For each $i\in Z(\overline{2})$ by using equation (21) we can express $\theta_i(z_P+z_Q)=\gamma_i X+\delta_i$. (We can always compute an inverse of $\gamma X+\delta$ except when $-\delta/\gamma$ is a root of $\mathfrak{P}_\alpha$. But in this case we have found a root of $\mathfrak{P}_\alpha$ and we can use the first method.) Now, consider the vector $(T_j)_{j\in Z(\overline{2})}$ where $T_0=1$, $T_\alpha=X$ and $T_j=\gamma_j X+\delta_j$. We compute $R=\mathrm{ScalarMult}(T,Q,P,\widetilde{0},\ell)_i$. Then it is easily seen that

$$R + g.R = \text{ScalarMult}(P+Q, Q, P, \widetilde{0}, \ell)_i + \text{ScalarMult}(P-Q, Q, P, \widetilde{0}, \ell)_i.$$

By Proposition 2, and using the fact that $\theta_i(-z_Q) = \theta_i(z_Q)$ we have for $i \in Z(\overline{2})$ $\overline{e}_T(\overline{P}, \overline{Q}) = \frac{[\theta_i(\ell.z_Q+z_P)+\theta_i(-\ell.z_Q+z_P)]\theta_i(0)}{\theta_i(z_P)\theta_i(\ell.z_Q)}$. We can now compute

$$\overline{e}_T(P,Q) = \frac{[\text{ScalarMult}(P+Q, Q, P, \widetilde{0}, \ell)_i + \text{ScalarMult}(P-Q, Q, P, \widetilde{0}, \ell)_i]\theta_i(0)}{\theta_i(z_P)\text{ScalarMult}(Q, Q, 0, \widetilde{0}, \ell)_i},$$

By an application of Lemma 3, the result of the preceding equation is a well defined element of $(K^*/K^{*\ell})_0$.

With this method, we have to compute 1 ScalarMult with value in $\mathscr{A}$ and 1 ScalarMult with value in K. It is interesing to note that it avoids the non determinism of the square root computation of the first method.

In some cryptographic applications, it is important to have a unique value as the result of the Tate pairing. In order to have this property, it is common to compose the Tate pairing with a ℓ^{th} root extraction on K which can be done in the case that K is a finite field by an exponentiation in K_0^*. This operation can be performed using the Montgomery ladder type algorithm presented above.

The symmetric Weil pairing computation. Since we compute $\overline{P+Q}$ with the first method, we can compute the Weil pairing as in the level 4 case.

We explain how to compute it with the second method: let $P, Q \in A[\ell]$ and denote by $\overline{P}, \overline{Q}$ the corresponding points in $\mathscr{K}_A$. Denote by $\theta_i(z)$, $i \in Z(\overline{2})$ the level 2 theta functions associated to Ω. By Corollary 2, we have:

$$\overline{e}_W(\overline{P}, \overline{Q}) = \frac{\theta_i(z_Q)\theta_i(\ell.z_P)}{\theta_i(z_P)\theta_i(\ell.z_Q)\theta_i(z_Q+\ell.z_P)\theta_i(z_Q-\ell.z_P)} \times \\ [\theta_i(\ell.z_Q+z_P)\theta_i(z_Q-\ell z_P) + \theta_i(\ell.z_Q-z_P)\theta_i(z_Q+\ell z_P)]. \quad (22)$$

The denominator of this expression can be easily computed from the knowledge of $\theta_i(z_Q)$, $\theta_i(\ell.z_Q)$, $\theta_i(z_P)$ and $\theta_i(\ell.z_P)$ by using the addition formula (1). The numerator can be computed in the algebra $\mathscr{A}$ in the following way: keeping the notations from above, we compute $R' = \text{ScalarMult}(T, Q, P, \widetilde{0}, \ell)_i.\text{ScalarMult}(gT, P, Q, \widetilde{0}, \ell)_i$. We obtain that $R' + g.R' = \text{ScalarMult}(P+Q, Q, P, \widetilde{0}, \ell)_i.\text{ScalarMult}(P-Q, P, Q, \widetilde{0}, \ell)_i + \text{ScalarMult}(P-Q, Q, P, \widetilde{0}, \ell)_i.\text{ScalarMult}(P+Q, P, Q, \widetilde{0}, \ell)_i$, which gives the numerator of (22).

6 An Example in Dimension 2

In this section we give an example of compution of the pairings on a dimension 2 Jacobian. Let H be the hyperelliptic curve over the prime field $\mathbb{F}_p$, $p = 331$, given by the equation:

$$Y^2 = X^5 + 204X^4 + 198X^3 + 80X^2 + 179X.$$

Let J be the Jacobian of H. The cardinal of $J(\mathbb{F}_p)$ is $2^6 \cdot 1889$ (since we are in level 2, all the 2-torsion points of J are rational), so that we let $\ell = 1889$, and

the embedding degree k corresponding to ℓ is 4. A theta null point of level 2 associated to J is given by $(328 : 213 : 75 : 1)$. Let $P = (255 : 89 : 30 : 1)$, we have $P \in J[\ell](\mathbb{F}_p)$. Let $\mathbb{F}_{p^k} \simeq \mathbb{F}_p(t)/(t^4 + 3t^2 + 290t + 3)$. We let Q be the $\mathbb{F}_{p^k}$-point of ℓ-torsion whose coordinates are:

$$(158t^3 + 67t^2 + 9t + 293 : 290t^3 + 25t^2 + 235t + 280 : 155t^3 + 84t^2 + 15t + 170 : 1).$$

We compute (and fix an arbitrary ordering):

$$P + Q = (217t^3 + 271t^2 + 33t + 303 : 308t^3 + 140t^2 + 216t + 312 : 274t^3 + 263t^2 + 284t + 302 : 1),$$
$$P - Q = (62t^3 + 16t^2 + 255t + 129 : 172t^3 + 157t^2 + 43t + 222 : 258t^3 + 39t^2 + 313t + 150 : 1).$$

Finally, we let $r = \frac{p^k-1}{\ell} = 6354480$ and $\zeta = t^r$ be a primitive ℓ^{th}-root of unity. We then compute using the doubling and differential addition algorithms:

$$\ell\widetilde{P} = (12, 141, 31, 327) = 327.\widetilde{0},$$
$$\ell\widetilde{Q} = (21t^3 + 280t^2 + 101t + 180, 164t^3 + 311t^2 + 111t + 129,$$
$$137t^3 + 282t^2 + 123t + 134, 324t^3 + 17t^2 + 187t + 271) = (324t^3 + 17t^2 + 187t + 271).\widetilde{0},$$
$$\text{ScalarMult}(\widetilde{P+Q}, \widetilde{Q}, \widetilde{P}, \widetilde{0}, \ell) = (45t^3 + 118t^2 + 219t + 308, 152t^3 + 97t^2 + 166t + 40,$$
$$200t^3 + 267t^2 + 201t + 192, 117t^3 + 42t^2 + 106t + 205) = (117t^3 + 42t^2 + 106t + 205).\widetilde{P},$$
$$\text{ScalarMult}(\widetilde{P+Q}, \widetilde{P}, \widetilde{Q}, \widetilde{0}, \ell) = (50t^3 + 31t^2 + 84t + 309, 168t^3 + 196t^2 + 275t + 234,$$
$$67t^3 + 186t^2 + 159t + 102, 243t^3 + 320t^2 + 222t + 200) = (243t^3 + 320t^2 + 222t + 200).\widetilde{Q}.$$

We then compute (following the previous ordering):

$$e_W(P,Q) = \frac{243t^3 + 320t^2 + 222t + 200}{327} \cdot \frac{324t^3 + 17t^2 + 187t + 271}{117t^3 + 42t^2 + 106t + 205} = \zeta^{-1},$$
$$e_T(P,Q) = \left(\frac{117t^3 + 42t^2 + 106t + 205}{324t^3 + 17t^2 + 187t + 271}\right)^r = \zeta^{1068},$$
$$e_T(Q,P) = \left(\frac{243t^3 + 320t^2 + 222t + 200}{327}\right)^r = \zeta^{1184}.$$

Here the Tate pairings are normalized by taking their $r = (p^k - 1)/\ell$-power. The symmetric pairings are then given by $\overline{e}_W(P,Q) = 61t^3 + 285t^2 + 196t + 257$ and $\overline{e}_T(P,Q) = 194t^3 + 163t^2 + 97t + 164$.

7 Conclusion

In this paper, we have presented an algorithm based on theta functions to compute Weil and Tate pairings. It would be interesting to carry out a fine grained study of the efficiency of our algorithm depending on the target implementation (software, hardware etc.) and to compare it with existing implementations based on Miller's algorithm.

Acknowledgement

The authors of this paper would like to thank anonymous referees for their careful reading and helpful comments on an earlier version of the paper.

References

1. Birkenhake, C., Lange, H.: Complex abelian varieties, 2nd edn. Grundlehren der Mathematischen Wissenschaften, Fundamental Principles of Mathematical Sciences, vol. 302. Springer, Berlin (2004)
2. Cohen, H., Frey, G., Avanzi, R., Doche, C., Lange, T., Nguyen, K., Vercauteren, F. (eds.): Handbook of elliptic and hyperelliptic curve cryptography. Discrete Mathematics and its Applications. Chapman & Hall/CRC (2006)
3. Galbraith, S., Lin, X.: Computing pairings using x-coordinates only. Designs, Codes and Cryptography (2008)
4. Gaudry, P.: Fast genus 2 arithmetic based on Theta functions. J. of Mathematical Cryptology 1, 243–265 (2007)
5. Gaudry, P., Lubicz, D.: The arithmetic of characteristic 2 Kummer surfaces and of elliptic Kummer lines. Finite Fields Appl. 15(2), 246–260 (2009)
6. Igusa, J.-i.: Theta functions. Springer, New York (1972); Die Grundlehren der mathematischen Wissenschaften, Band 194
7. Koizumi, S.: Theta relations and projective normality of Abelian varieties. Amer. J. Math. 98(4), 865–889 (1976)
8. Lang, S.: Reciprocity and correspondences. Amer. J. Math. 80, 431–440 (1958)
9. Lichtenbaum, S.: Duality theorems for curves over p-adic fields. Invent. Math. 7, 120–136 (1969)
10. Lubicz, D., Robert, D.: Computing isogenies between abelian varieties (2010), http://arxiv.org/abs/1001.2016
11. Mumford, D.: On the equations defining abelian varieties. I. Invent. Math. 1, 287–354 (1966)
12. Mumford, D.: On the equations defining abelian varieties. II. Invent. Math. 3, 75–135 (1967)
13. Mumford, D.: Abelian varieties. Tata Institute of Fundamental Research Studies in Mathematics, vol. 5. Published for the Tata Institute of Fundamental Research, Bombay (1970)
14. Mumford, D.: Tata lectures on theta I. Progress in Mathematics, vol. 28. Birkhäuser Boston Inc., Boston (1983); With the assistance of Musili, C., Nori, M., Previato E., Stillman, M.
15. Mumford, D.: Tata lectures on theta II. Progress in Mathematics, vol. 43. Birkhäuser Boston Inc., Boston (1984); Jacobian theta functions and differential equations, With the collaboration of Musili, C., Nori, M., Previato, E., Stillman, M., Umemura, H.
16. Silverman, J.H.: The arithmetic of elliptic curves. Graduate Texts in Mathematics, vol. 106. Springer, New York (1986); Corrected reprint of the 1986 original (1986)

Small-Span Characteristic Polynomials of Integer Symmetric Matrices

James McKee

Department of Mathematics, Royal Holloway, University of London,
Egham Hill, Egham, Surrey, TW20 0EX, England, UK
james.mckee@rhul.ac.uk

Abstract. Let $f(x) \in \mathbf{Z}[x]$ be a totally real polynomial with roots $\alpha_1 \leq \ldots \leq \alpha_d$. The *span* of $f(x)$ is defined to be $\alpha_d - \alpha_1$. Monic irreducible $f(x)$ of span less than 4 are special. In this paper we give a complete classification of those small-span polynomials which arise as characteristic polynomials of integer symmetric matrices. As one application, we find some low-degree polynomials that do not arise as the *minimal* polynomial of *any* integer symmetric matrix: these provide low-degree counterexamples to a conjecture of Estes and Guralnick [6].

1 Introduction

1.1 History of the Small Span Problem

Let $f(x) \in \mathbf{Z}[x]$ be a monic polynomial having only real roots. If these roots are $\alpha_1 \leq \ldots \leq \alpha_d$ then we say that $f(x)$ has *span* $\alpha_d - \alpha_1$. In the case where $f(x)$ is irreducible, the roots are (Galois) conjugates of each other and we then refer to $\{\alpha_1, \ldots, \alpha_d\}$ as a *conjugate set*. If a real interval I has length strictly less than 4, then it is known [19] that I contains only finitely many conjugate sets of algebraic integers. If I has length greater than 4 then it contains infinitely many such conjugate sets [17]. The problem remains open for intervals of length exactly 4, unless the endpoints are integers, in which case there are infinitely many such sets [11].

Monic $f(x) \in \mathbf{Z}[x]$ of span less than 4 have therefore attracted some interest: for convenience we shall call these *small-span* polynomials. The span is unchanged if we replace $f(x)$ by $\varepsilon^{\deg f} f(\varepsilon x + c)$ for any choice of $\varepsilon \in \{-1, 1\}$ and any integer c: two polynomials related in this way are deemed to be *equivalent*. The number of equivalence classes of small-span polynomials of any given degree is finite. Robinson [18] produced a complete list of representatives for degrees up to 6, with conjectured lists for degrees 7 and 8 that were later verified as complete. Recently Capparelli, Del Fra and Sciò [2] extended this computation (using new techniques) up to degree 14.

For any natural number m, the totally real algebraic integer $2\cos(2\pi/m)$ has its conjugate set lying in the interval $[-2, 2]$; we call the minimal polyomial of such a number a *cosine polynomial*. Examples of irreducible small-span $f(x)$ not equivalent to one of these cosine polynomials are of special interest.

G. Hanrot, F. Morain, and E. Thomé (Eds.): ANTS-IX 2010, LNCS 6197, pp. 270–284, 2010.

1.2 Characteristic Polynomials of Integer Symmetric Matrices

For any n-by-n integer symmetric matrix A we define its *characteristic polynomial*, $\chi_A(x)$, by $\chi_A(x) = \det(xI - A)$, where I is the n-by-n identity matrix. Clearly $\chi_A(x)$ is a monic polynomial with integer coefficients; moreover all its roots are real since A is a real symmetric matrix. We define the *span* of A to be the span of its characteristic polynomial, and we say that A is a *small-span* integer symmetric matrix if it has span less than 4.

A more usual measure of the size of the eigenvalues of A is its *spectral radius*, defined to be the largest modulus of any eigenvalue. Plainly the span of A is bounded above by twice its spectral radius. If the spectral radius is at most 2, then the characteristic polynomial is a small-span cosine polynomial (or a product of such polynomials). See [14] for a classification of all integer symmetric matrices of spectral radius below 2.019: there are no non-cosine small-span examples. There is a similar list in [14] of all $f(x)$ arising as characteristic polynomials of integer symmetric matrices for which the Mahler measure of $x^{\deg f} f(x + 1/x)$ is below 1.3: if the Mahler measure is 1, then one has a cosine example, and amongst those for which the Mahler measure is close to 1 one finds some, but not all, non-cosine small-span examples.

Petrović [16] classified all graphs whose characteristic polynomial has span at most 4. From this one can easily deduce which cases give span less than 4. The adjacency matrices of such graphs are special cases of integer symmetric matrices, with the entries restricted to $\{0, 1\}$, and with only zero entries on the main diagonal.

If $f(x) \in \mathbf{Z}[x]$ is monic and totally real, then one can sensibly ask whether or not it arises as the characteristic polynomial of an integer symmetric matrix. Not every such $f(x)$ arises in this way: we shall see some examples that do not, below. On the other hand, it is known (see [5], or [1]) that every totally real algebraic integer α is the eigenvalue of some integer symmetric matrix A, so that the minimal polynomial of α divides $\chi_A(x)$.

1.3 Minimal Polynomials of Integer Symmetric Matrices: A Conjecture of Estes and Guralnick

With mystery surrounding the question of which polynomials $f(x)$ arise as $\chi_A(x)$ for some integer symmetric matrix A, Estes and Guralnick [6] turned their attention to the *minimal* polynomial $m_A(x)$, defined as the monic polynomial in $\mathbf{Z}[x]$ of minimal degree such that $m_A(A) = 0$. One has that $m_A(x)$ divides $\chi_A(x)$, and that every root of χ_A is a root of m_A [9, §11.6]. For an integer symmetric matrix A, the minimal polynomial $m_A(x)$ must be separable (i.e., its roots are distinct) since A is diagonalisable. Estes and Guralnick showed [6, Corollary C] that if $f(x) \in \mathbf{Z}[x]$ has degree $n \leq 4$, has all roots real, and is monic and separable, then $f(x)$ is the minimal polynomial of a $2n$-by-$2n$ integer symmetric matrix.

For example, one can easily show that $x^2 - 3$ is not the characteristic polynomial of an integer symmetric matrix, but it satisfies all the hypotheses of the Estes-Guralnick theorem, and sure enough we find that

$$\begin{pmatrix} -1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & -1 \\ 0 & 1 & -1 & -1 \end{pmatrix}$$

has minimal polynomial $x^2 - 3$. For a less trivial example, we shall see below in §3 that $x^3 - 4x - 1$ is not the characteristic polynomial of any integer symmetric matrix. Yet it is the minimal polynomial of

$$\begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & -1 & 0 & 1 \\ 1 & 1 & -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

At the end of their paper [6], Estes and Guralnick ask whether or not *every* monic, separable, totally real $f(x) \in \mathbf{Z}[x]$ is the minimal polynomial of an integer symmetric matrix: they conjecture that the answer is 'yes' (p. 84). This question was answered in the negative by Dobrowolski [4]. He showed that any degree-n irreducible minimal polynomial of an integer symmetric matrix has discriminant at least n^n, and then observed that infinitely many cosine polynomials have smaller discriminant than this (for a precise formula for the discriminant of a cosine polynomial see [18, p. 554], derived from a formula in [12]). The smallest degree of any of Dobrowolski's counterexamples to the conjecture of Estes and Guralnick is 2880; we shall give below some counterexamples of degree 6, for which the discriminant is too large for Dobrowolski's argument to apply. It remains an open problem as to whether or not there are any counterexamples of degree 5.

1.4 The Contributions of This Paper

In this paper we ask which monic, irreducible, totally real polynomials in $\mathbf{Z}[x]$ of span less than 4 arise as characteristic polynomials of integer symmetric matrices. For this restricted class of polynomials, we are able to give a complete classification (Theorem 3; more precisely, Theorem 3 classifies the integer symmetric matrices that give rise to small-span characteristic polynomials). As a byproduct of this, we are able to address the conjecture of Estes and Guralnick about *minimal* polynomials [6, p. 84], and produce some counterexamples with degree as small as 6.

In §2 we describe the algorithm for computing the complete list of representatives of equivalence classes of small-span integer symmetric matrices up to any desired degree. This builds on similar algorithms in [13] and [14]. In §3 we detail the results. In §4, we prove a classification theorem for the small-span polynomials which arise as characteristic polynomials of integer symmetric matrices. The paper concludes by applying this to the conjecture of Estes and Guralnick.

2 The Growing Algorithm

2.1 Equivalence

Let $O_n(\mathbf{Z})$ be the orthogonal group of n-by-n signed permutation matrices. If A is an n-by-n integer symmetric matrix, and $P \in O_n(\mathbf{Z})$, then we call A and $P^{-1}AP = P^TAP$ *strongly equivalent*. Strongly equivalent matrices have the same characteristic polynomial.

Let A be an n-by-n integer symmetric matrix, and let c be any integer. Then $\chi_{A+cI}(x) = \chi_A(x-c)$. Also $\chi_{-A}(x) = (-1)^n\chi_A(-x)$. Thus if $f(x)$ is the characteristic polynomial of an integer symmetric matrix, then so is any polynomial equivalent to $f(x)$ in the sense of §1.1. We define integer symmetric matrices A and B to be *equivalent* if A is strongly equivalent to $\pm B + cI$ for some integer c. Thus equivalent matrices have equivalent characteristic polynomials. If A has span less than 4, then by adding cI for suitable c we can move to an equivalent matrix B with all eigenvalues in the interval $[-2, 3)$; if B has an eigenvalue greater than 2.5, then it has no eigenvalue smaller than -1.5, and we replace B by the equivalent matrix $-B + I$. We see that any small-span integer symmetric matrix is equivalent to one with all eigenvalues in the interval $[-2, 2.5)$.

Our conclusion is that in order to find which monic, totally real polynomials in $\mathbf{Z}[x]$ of degree n and span less than 4 arise as characteristic polynomials of integer symmetric matrices, it is enough to find all n-by-n integer symmetric matrices up to strong equivalence that satisfy both: (i) the span is less than 4; and (ii) all eigenvalues lie in the interval $[-2, 2.5)$.

2.2 Indecomposable Matrices

An integer symmetric matrix will be called *decomposable* if one can apply a permutation to the rows, and the same permutation to the columns, to produce a matrix in block diagonal form with more than one block. A matrix that is not decomposable is *indecomposable*. The characteristic polynomial of a decomposable matrix is the product of the characteristic polynomials of its blocks. In attempting to understand which polynomials arise as characteristic polynomials, it is therefore enough to restrict to indecomposable matrices.

There is a nice graph-theoretic description of the property of being indecomposable. The *underlying graph* of an integer symmetric matrix has vertices labelled by the rows, with an edge between vertex i and vertex j precisely when the (i, j)-entry in the matrix is non-zero. Then a matrix is indecomposable if and only if the underlying graph is connected. We record a standard lemma whose proof is obvious given this interpretation.

Lemma 1. *Let A be an n-by-n indecomposable matrix, with $n \geq 2$. Then there is a choice of i between 1 and n such that deleting row i and column i from A leaves an indecomposable submatrix.*

When convenient, we shall use the language of graphs to talk about our matrices. We speak of vertices to indicate rows, edges to indicate non-zero matrix entries,

with natural interpretations of paths, cycles, connectedness, and so on. The distance between two vertices will mean the minimal number of edges on a path from one to the other. If our matrix has a non-zero entry on the diagonal, then we refer to the corresponding vertex as being *charged*.

Lemma 1 is a corollary of the following slightly more precise result, which we shall exploit later.

Lemma 2. *Let G be a connected graph with at least 2 vertices, and let i and j be vertices for which the distance between i and j is maximal. Then deleting vertex i (and all incident edges) does not disconnect the graph.*

Proof. Suppose that after deleting i there was a vertex k not in the same component as j. Then every path from k to j in G would have to pass through i, and so the distance from k to j would be strictly greater than that from i to j, giving a contradiction.

2.3 Interlacing

We shall make much use of Cauchy's interlacing theorem [3] (for more accessible proofs, see [8], [10] or [7]).

Theorem 1 (Cauchy, 1829). *Let A be an n-by-n integer symmetric matrix, with $n \geq 2$, and let B be an $(n-1)$-by-$(n-1)$ submatrix formed by deleting row i and column i from A (for some choice of i between 1 and n). Let $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ be the eigenvalues of A, and let $\mu_1 \leq \ldots \leq \mu_{n-1}$ be those of B. Then these two sets of eigenvalues interlace:*

$$\lambda_1 \leq \mu_1 \leq \lambda_2 \leq \mu_2 \leq \ldots \leq \mu_{n-1} \leq \lambda_n \,.$$

From this we have an immediate corollary which will be of use in our algorithm for computing small-degree small-span integer symmetric matrices.

Corollary 1. *Let A be an n-by-n integer symmetric matrix, with $n \geq 2$, and let B be an $(n-1)$-by-$(n-1)$ submatrix formed by deleting row i and column i from A (for some choice of i between 1 and n). Then the span of A is at least as large as the span of B. Moreover, if A has all its eigenvalues in the interval $[-2, 2.5)$, then so does B.*

2.4 Reduction

Our situation would be considerably more pleasant if for any integer symmetric matrix A we could quickly find a canonical representative of its strong equivalence class. Unfortunately this is not the case, and we content ourselves with a quick 'reduction' process that gives us a semi-canonical representative, but with the possibility that there are several different 'reduced' elements in the same strong equivalence class. Some balance must be struck between the speed of reduction and the possible number of strongly-equivalent reduced matrices.

In practice we used two complementary reduction processes, which for convenience we call *fast reduction* and *slow reduction*. The first of these is generally much faster and was used to identify quickly many cases of strong equivalence. The slower reduction process was then used to produce further weeding of our lists of matrices. This double reduction was then repeated until no further weeding was achieved. Any matrices in the final list having the same characteristic polynomial (and sharing a few other invariants of strong equivalence) were flagged for further inspection: in all such cases, either an equivalence between the two examples was found, or some simple argument established that the two were not equivalent.

The principle of fast reduction is to give a 'score' to each row of the matrix, such that the multiset of scores is invariant under strong equivalence. The rows and columns would then be ordered according to this score. Finally, if the first non-zero entry of any row was negative (and not on the diagonal) then that row (and the corresponding column) would have its sign changed. A more complicated scoring system would take longer to compute but would reduce the number of rows having equal score and thereby reduce the risk of having more than one possible reduced matrix in the same strong equivalence class. The scoring system that we used was to compute the first three powers of the matrix A and then rank rows by a linear combination of: (i) the sum of the moduli of the entries in the row; (ii) the same for A^2; (iii) the same for A^3; (iv) the size of the diagonal entry.

The aim of slow reduction was to attempt to find the lexicographically smallest element of a strong equivalence class. If always successful then this would provide a perfect reduction process, but to achieve this perfection would be painfully slow. Instead one deemed a matrix to be reduced if it was 'locally minimal' with respect to lexicographical ordering in the sense that: (i) changing the sign of any row (and column) would give a larger matrix (in the sense of the ordering); (ii) swapping any two rows (and the corresponding columns) would give a larger matrix; (iii) cyclically permuting any three rows (and the corresponding columns) would produce a larger matrix.

There is no claim that the combination of fast and slow reduction detailed above is optimally efficient, but both reduction methods significantly reduced the number of matrices needing to be considered, and enabled the computations to proceed smoothly up to the sizes detailed below.

2.5 Bounds on Entries and Valencies

Using interlacing (Theorem 1 to bound the size of diagonal entries, and Corollary 1 to deal with off-diagonal entries) we can rapidly restrict the possible entries for integer symmetric matrices that are of interest to us.

Lemma 3. *Let A be a small-span integer symmetric matrix with all eigenvalues in the interval $[-2, 2.5)$. Then all entries of A have absolute value at most 2, and all off-diagonal entries have absolute value at most 1.*

Proof. Let a be a diagonal entry in A. Then since (a) has a as an eigenvalue, repeated use of Theorem 1 shows that A has an eigenvalue with modulus at least as large as $|a|$. Our restriction on the eigenvalues of A shows that $|a| \leq 2$.

Let b be an off-diagonal entry of A. Then deleting other rows and columns gives a submatrix of the shape $\begin{pmatrix} a & b \\ b & c \end{pmatrix}$. By repeated use of Corollary 1, this submatrix must have span less than 4, giving $\sqrt{(a-c)^2 + 4b^2} < 4$. This implies $|b| \leq 1$.

The cases where there is an entry that has absolute value 2 are extremely restricted. The following Lemma describes the complete list.

Lemma 4. *Up to strong equivalence, the only indecomposable small-span integer symmetric matrices with all eigenvalues in the interval* $[-2, 2.5)$ *and containing an entry of modulus greater than* 1 *are:*

$$(-2), (2), \begin{pmatrix} 2 & 1 \\ 1 & -1 \end{pmatrix}, \begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 2 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 2 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

(The first two matrices listed in Lemma 4 are equivalent, but not strongly equivalent.)

Proof. Each of the five 1-by-1 matrices (-2), (-1), (0), (1), (2) was grown in all possible ways to larger indecomposable small-span integer symmetric matrices with all eigenvalues in the interval $[-2, 2.5)$, allowing entries from $\{-2, -1, 0, 1, 2\}$ in accordance with Lemma 3. After producing a provisional list of 2-by-2 matrices, this list was weeded by reduction, as described in §2.4. Repeating this growing process three more times revealed that there are no 5-by-5 examples containing an entry having modulus greater than 1, and by interlacing the same must be true for all larger indecomposable integer symmetric matrices. The output of this computation also established the advertised list.

Having reduced to the problem of considering matrices that have absolute value at most 1, we now further restrict the possible entries in each row.

Lemma 5. *Let* A *be an indecomposable small-span integer symmetric matrix with all eigenvalues in the interval* $[-2, 2.5)$. *Then each row of* A *has at most* 4 *non-zero entries.*

Proof. After Lemma 4, we can suppose that all entries in A are from the set $\{-1, 0, 1\}$.

If Lemma 5 were false, then by interlacing (and making use of strong equivalence) there would be a small-span integer symmetric matrix M with all eigenvalues in the interval $[-2, 2.5)$ and with M being one of

$$\begin{pmatrix} -1 & 1 & 1 & 1 & 1 \\ 1 & a & b & c & d \\ 1 & b & e & f & g \\ 1 & c & f & h & i \\ 1 & d & g & i & j \end{pmatrix}, \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & a & b & c & d \\ 1 & b & e & f & g \\ 1 & c & f & h & i \\ 1 & d & g & i & j \end{pmatrix}, \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & a & b & c & d & e \\ 1 & b & f & g & h & i \\ 1 & c & g & j & k & l \\ 1 & d & h & k & m & n \\ 1 & e & i & l & n & o \end{pmatrix},$$

where the unspecified entries are all from $\{-1, 0, 1\}$. A computer search showed that no such matrix M exists.

2.6 The Algorithm

Lemma 1 and Corollary 1 suggest a means of 'growing' indecomposable small-span integer symmetric matrices with all eigenvalues in $[-2, 2.5)$ from smaller matrices. This idea has been used before for computing integer symmetric matrices with small spectral radius or small Mahler measure ([13] and [14]). Having established Lemmas 4 and 5, we grow indecomposable matrices with all entries coming from the set $\{-1, 0, 1\}$, and with the extra restriction that each row can contain no more than four non-zero entries. After producing a provisional list of r-by-r matrices, this list is weeded by reduction, as described in §2.4, before growing to produce a list of $(r+1)$-by-$(r+1)$ matrices.

The complete search up to 13-by-13 matrices was completed in under five hours on a single processor. This was enough to provide the computational element of the proof of Theorem 3 below. The computation was pushed up to 20-by-20 matrices in under six days; perfect agreement of the results with Theorems 2 and 3 for larger matrices provided confidence in the correctness of the output for smaller matrices. The PARI code for all of this is freely available from the author on request.

After each growing of a list of $(n-1)$-by-$(n-1)$ matrices to a list of n-by-n matrices, any examples from the first list that had not been grown to one or more examples in the second were recorded in a list of *maximal* examples. Some of these maximal examples fitted into infinite families, described in Theorem 2; others did not, and these we call *sporadic*.

3 Results

We shall call an indecomposable small-span integer symmetric matrix that has all eigenvalues in the interval $[-2, 2.5)$ *maximal* if it cannot be obtained by deleting rows (and corresponding columns) from any larger indecomposable small-span integer symmetric matrix with all eigenvalues in the interval $[-2, 2.5)$. It turns out that every indecomposable small-span integer symmetric matrix with all eigenvalues in the interval $[-2, 2.5)$ can be grown to a maximal one (part of Theorem 3). In view of Corollary 1, it is enough to describe all the maximal matrices. Up to strong equivalence there are 197 sporadic examples and 10 infinite families. In this section we tabulate the number of sporadic examples of

each size, found by computation as outlined above. The infinite families and the proof of completeness of the classification will follow in §4 (Theorems 2 and 3). Members of the infinite families all in fact have eigenvalues in the smaller interval $[-2, 2]$.

The following table includes the three maximal examples from Lemma 4. Maximal examples that are members of the infinite families of Theorem 2 are excluded: only the sporadic cases are counted. The computations had been done up to size 20-by-20, but the only maximal cases that were not covered by the infinite families of Theorem 2 were 12-by-12 or smaller. That no further sporadic maximal examples arise is the point of Theorem 3.

Sporadic maximal indecomposable small-span integer symmetric matrices with all eigenvalues in $[-2, 2.5)$, up to strong equivalence

n	n-by-n cosine examples	n-by-n non-cosine examples	total
1	1	0	1
2	0	1	1
3	0	1	1
4	10	9	19
5	0	19	19
6	0	43	43
7	0	28	28
8	11	39	50
9	0	15	15
10	0	15	15
11	0	2	2
12	0	3	3
total	22	175	197

For degrees up to 8, most small-span irreducible polynomials arise as characteristic polynomials of integer symmetric matrices: it is simpler to record which of Robinson's polynomials from [18] do *not* arise. It is interesting to note that all examples of degrees 4 and 5 appear. The missing examples for degrees 2 and 3 are those mentioned in §1.3 above, namely $x^2 - 3$ and $x^3 - 4x - 1$. The other missing polynomials are numbers $6g$, $6i$, $6k$, $7j$, $7k$, $7l$, $8a$, $8c$, $8l$, $8m$, $8t$, $8u$, $8y$ in Robinson's list [18].

For degree 9, both of the inequivalent cosine polynomials arise as characteristic polynomials, and three other irreducibles: $x^9 - x^8 - 9x^7 + 7x^6 + 28x^5 - 15x^4 - 34x^3 + 10x^2 + 12x - 1$, $x^9 - 4x^8 - 2x^7 + 21x^6 - 5x^5 - 37x^4 + 12x^3 + 24x^2 - 5x - 4$, $x^9 - 3x^8 - 5x^7 + 18x^6 + 7x^5 - 34x^4 - x^3 + 20x^2 - 3x - 1$. For degree 10, the only irreducible small-span characteristic polynomial is the non-cosine example $x^{10} - 5x^9 + x^8 + 26x^7 - 21x^6 - 49x^5 + 40x^4 + 42x^3 - 20x^2 - 15x - 1$. For degree 11, the only one (up to equivalence) is the cosine case.

For degree 13 and above, Theorem 3 (below) gives a complete description of which characteristic polynomials arise. All degree-13 examples that have span below 4 and all eigenvalues in the interval $[-2, 2.5)$ in fact have all eigenvalues in the subinterval $[-2, 2]$ (this is the content of Theorem 3), and hence are described in Theorem 2.

The following table compares the complete lists of [18] and [2] with the results of the computations for characteristic polynomials, restricting to irreducible polynomials.

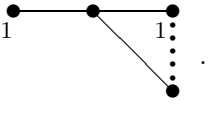
Degree	Number of irreducible small-span polynomials up to equivalence: cosine + non-cosine = total	Number that arise as characteristic polynomials of integer symmetric matrices: cosine + non-cosine = total
1	$1 + 0 = 1$	$1 + 0 = 1$
2	$3 + 1 = 4$	$2 + 1 = 3$
3	$2 + 3 = 5$	$2 + 2 = 4$
4	$4 + 10 = 14$	$4 + 10 = 14$
5	$1 + 14 = 15$	$1 + 14 = 15$
6	$4 + 13 = 17$	$1 + 13 = 14$
7	$0 + 15 = 15$	$0 + 12 = 12$
8	$5 + 21 = 26$	$5 + 14 = 19$
9	$2 + 19 = 21$	$2 + 3 = 5$
10	$3 + 15 = 18$	$0 + 1 = 1$
11	$1 + 10 = 11$	$1 + 0 = 1$
12	$7 + 9 = 16$	$0 + 0 = 0$
13	$0 + 4 = 4$	$0 + 0 = 0$

4 Classification of Small-Span Integer Symmetric Matrices

One result of our computations is that any indecomposable small-span 13-by-13 integer symmetric matrix with all its eigenvalues in $[-2, 2.5)$ in fact has all its eigenvalues in $[-2, 2]$. We shall now prove that this holds for all larger indecomposable matrices too. As a first step, we classify those indecomposable small-span integer symmetric matrices that have all their eigenvalues in the interval $[-2, 2]$.

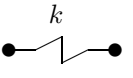
After Lemma 4, we are reduced to considering matrices that have entries 0, 1 or -1. These are conveniently represented by charged signed graphs. Vertices are labelled with their charges (corresponding to diagonal entries of the matrix); off-diagonal entries 1 and -1 are represented respectively by solid and dotted edges. Zero charges can be omitted to reduce clutter. For example, the matrix

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & -1 \\ 0 & 1 & -1 & 0 \end{pmatrix}$$

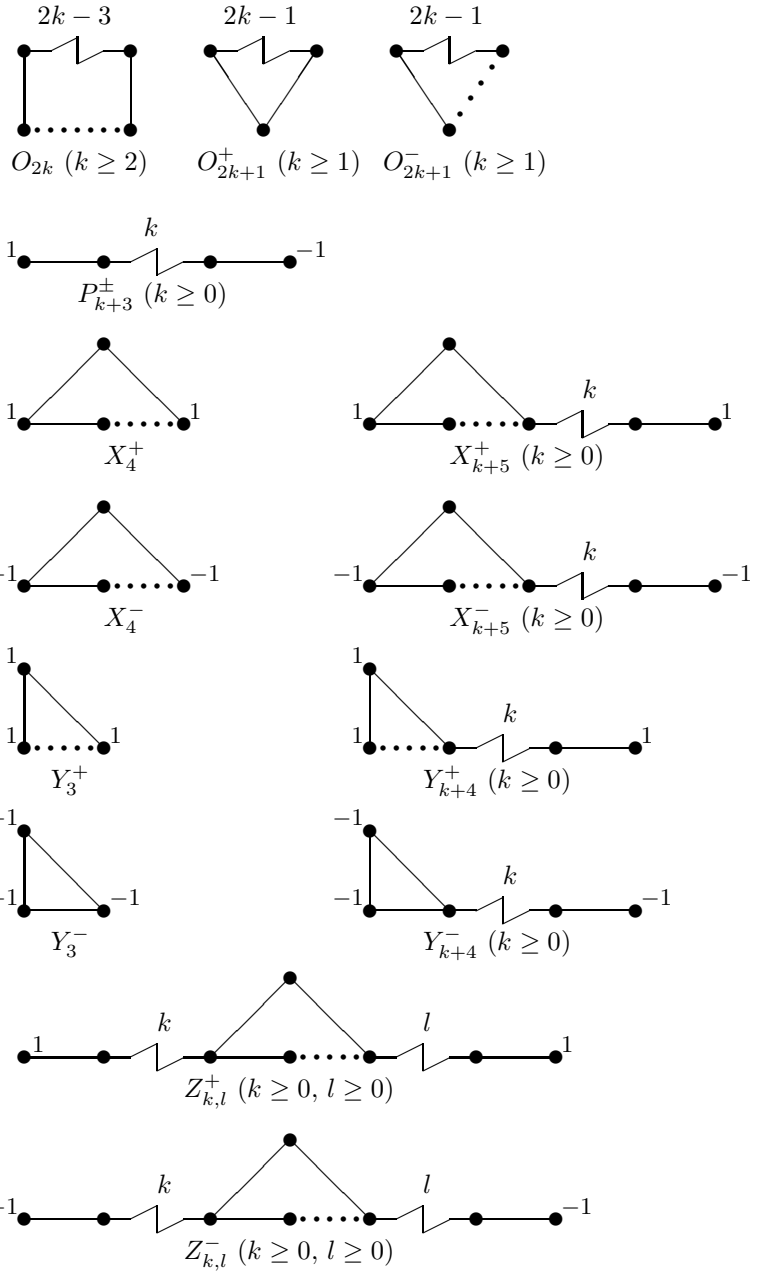
is drawn as 

.

In the graphs below, the symbol



denotes a path with k solid edges (and all vertices uncharged) between the displayed end vertices (if $k = 0$ then these end vertices are identified as a single vertex). Define graphs O_{2k}, O^+_{2k+1}, O^-_{2k+1}, $P^\pm_n$, X^+_n, X^-_n, Y^+_n, Y^-_n, $Z^+_{k,l}$, $Z^-_{k,l}$ as shown.

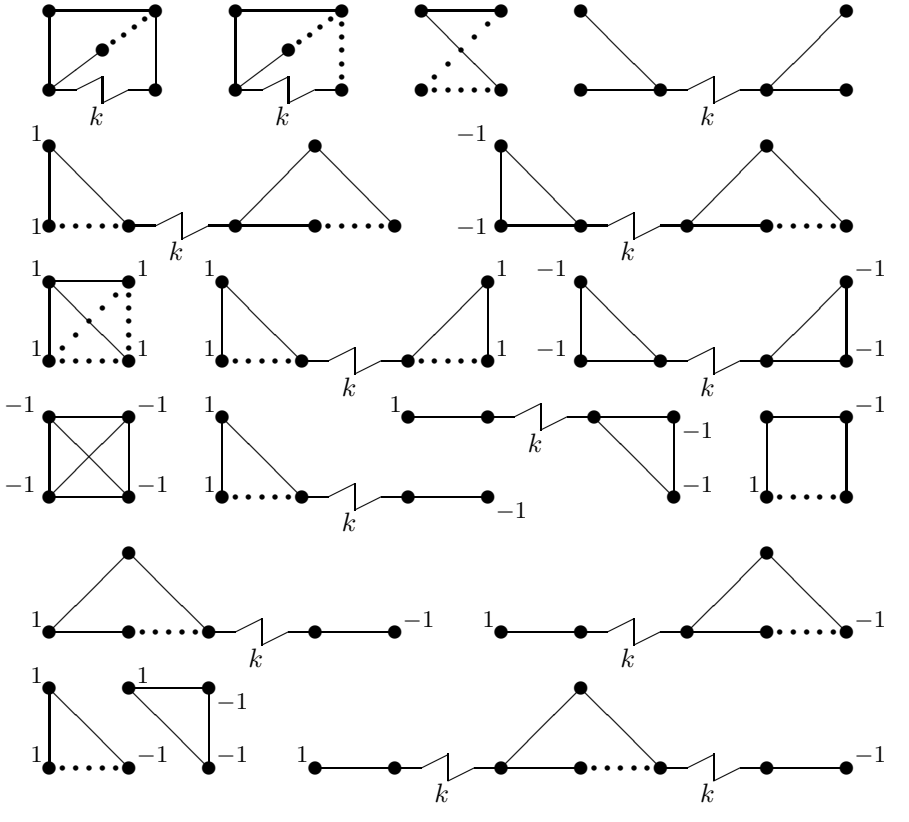


Theorem 2. *Every indecomposable small-span integer symmetric matrix M_1 that has all its eigenvalues in the interval $[-2, 2]$ is a submatrix of an indecomposable*

small-span integer symmetric matrix M_2 that is maximal subject to being small-span and having all its eigenvalues in $[-2, 2.5)$. Up to strong equivalence, the possibilities for M_2 are the sporadic maximal examples tabulated in Section 3 and the adjacency matrices of the charged signed graphs O_{2k} $(k \geq 4)$, O^+_{2k+1} $(k \geq 3)$, O^-_{2k+1} $(k \geq 2)$, $P^\pm_n$ $(n \geq 6)$, X^+_n $(n \geq 7)$, X^-_n $(n \geq 4)$, Y^+_n $(n \geq 6)$, Y^-_n $(n \geq 3)$, $Z^+_{k,l}$ $(k \geq l \geq 0$, except for $(k, l) \in \{(0, 0), (1, 0), (1, 1), (2, 1)\})$, $Z^-_{k,l}$ $(k \geq l \geq 0)$ pictured above.

Proof. This is a tedious but easy extension of the work in [13, §12] where all examples with eigenvalues in the open interval $(-2, 2)$ were described; here we relax this to consider the intervals $(-2, 2]$ and $[-2, 2)$. A convenient technique is that of Gram vectors. If an integer symmetric matrix A has all its eigenvalues in $[-2, 2]$, then both $B = A + 2I$ and $C = -A + 2I$ have all eigenvalues at least 0. Thus there are lists of Gram vectors $v_1, \ldots, v_n$ and $w_1, \ldots, w_n$ contained in $\mathbf{R}^n$ such that the (i, j)-entry of B (respectively C) is given by $v_i \cdot v_j$ (respectively $w_i \cdot w_j$). Now -2 is an eigenvalue of A if and only if $v_1, \ldots, v_n$ are linearly dependent, and 2 is an eigenvalue of A if and only if $w_1, \ldots, w_n$ are linearly dependent.

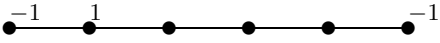
We start by noting that the following charged signed graphs have span 4: in each case one readily writes down linearly dependent sets of Gram vectors as above, showing that both -2 and 2 are eigenvalues, following the ideas in [13].



Next we note that the charged signed graphs O_{2k}, O^+_{2k+1}, O^-_{2k+1}, $P^\pm_n$, X^+_n, X^-_n, Y^+_n, Y^-_n, $Z^+_{k,l}$, $Z^-_{k,l}$ have all eigenvalues between -2 and 2 (they are equivalent to subgraphs of those listed in [13, §4]), and have span less than 4 (writing down Gram vector representations for each graph and its negative, one finds that in every case exactly one of the sets of Gram vectors is linearly independent).

Finally we check readily that any connected subgraph of one of those in [13, §4] that does not contain any subgraph equivalent to one of the span-4 examples listed above must be a subgraph of one of O_{2k}, O^+_{2k+1}, O^-_{2k+1}, $P^\pm_n$, X^+_n, X^-_n, Y^+_n, Y^-_n, $Z^+_{k,l}$, $Z^-_{k,l}$.

The restrictions on n, k and l require a trawl through the sporadic examples to see which of them contain any of the members of these 10 infinite families as subgraphs. For example, $P^\pm_5$ is a subgraph of the maximal sporadic example



.

Theorem 3. *Up to strong equivalence, the indecomposable small-span integer symmetric matrices with all eigenvalues in the interval* $[-2, 2.5)$ *are precisely the indecomposable submatrices of the* 197 *sporadic cases accounted for in* §3 *and the* 10 *infinite families of Theorem* 2. *In particular, every such matrix with more than* 12 *rows has all its eigenvalues in the interval* $[-2, 2]$.

Proof. In view of Theorem 2 and the computational results of §3, it is enough to show that every indecomposable integer symmetric matrix with more than 12 rows and all its eigenvalues in the interval $[-2, 2.5)$ in fact has all its eigenvalues in the interval $[-2, 2]$. Suppose for a contradiction that this is not the case. Let A be a counterexample that has as few rows as possible. We know from our computations that A has at least 14 rows, and this minimal counterexample would then have the property that any proper submatrix has all its eigenvalues in the interval $[-2, 2]$. The result now follows from the classification of all integer symmetric matrices minimal subject to not all eigenvalues being in the interval $[-2, 2]$: there are no such matrices with more than 10 rows [14]. But the current case is much easier, so we outline a direct proof. The key idea in the proof is that the property of having all eigenvalues in the interval $[-2, 2]$ is essentially described by local structure. In the general case treated in [14] this local structure is much more complicated than in the small-span case treated here.

Let G be the charged signed graph with adjacency matrix A (using Lemma 4). Pick vertices u and v as far apart as possible in G. Deleting either u or v leaves a connected (Lemma 2) charged signed graph with all eigenvalues in $[-2, 2]$ and with at least 13 vertices, and hence a connected subgraph of one of the infinite families of Theorem 2.

Deleting u leaves an underlying graph that is either a cycle or not. Suppose first that the underlying graph of G with u deleted is a cycle. Since u and v are maximally distant in G, we deduce that u is joined to vertices as far (or almost as far) as possible from v on this cycle, and since deleting v from G must give a connected subgraph of one of the infinite families of Theorem 2, the only possibility for G (up to strong equivalence) is a charged signed graph of the

shape formed by identifying the end vertices of $Z^+_{k,l}$, with the charges removed. But then A has all eigenvalues in the interval $[-2, 2]$ (see [13]) and in fact also has span 4, giving two contradictions.

Now suppose that deleting u does not leave a cycle. Then it leaves a structure that is up to strong equivalence either an uncharged path (perhaps with one negative edge) or is one of $P^{\pm}_n$, X^+_n, X^-_n, Y^+_n, Y^-_n, $Z^+_{k,l}$, $Z^-_{k,l}$, perhaps with one or more vertices removed in a way that does not disconnect the graph. Then either v is near the middle and u is adjacent to vertices at or near both ends of this structure, or v is at one end and u is adjacent to vertices at or near the other end. Again one sees (on considering deleting v, and using the classification in [13]) that A must have all eigenvalues in $[-2, 2]$, giving a contradiction.

5 Low-Degree Counterexamples to a Conjecture of Estes and Guralnick

Let $f(x)$ be a monic, irreducible, totally real, small-span polynomial of degree $n > 6$ that has all its eigenvalues in the interval $[-2, 2.5)$ but is not the characteristic polynomial of an integer symmetric matrix. Suppose further that $f(x)$ is *not* a cosine polynomial. Then $f(x)$ cannot be the *minimal* polynomial of any integer symmetric matrix. For if it were, then the smallest such matrix would be indecomposable and have characteristic polynomial $f(x)^r$ for some $r > 1$. But Theorem 3 precludes the existence of such characteristic polynomials, since the degree rn would be greater than 12. In particular, none of the polynomials $x^7 - x^6 - 7x^5 + 5x^4 + 15x^3 - 5x^2 - 10x - 1$, $x^7 - 8x^5 + 19x^3 - 12x - 1$ or $x^7 - 2x^6 - 6x^5 + 11x^4 + 11x^3 - 17x^2 - 6x + 7$ is the minimal polynomial of an integer symmetric matrix. These provide degree-7 counterexamples to the conjecture of Estes and Guralnick [6].

Finally we remark that none of the three degree-6 cosine polynomials $x^6 - x^5 - 6x^4 + 6x^3 + 8x^2 - 8x + 1$, $x^6 - 7x^4 + 14x^2 - 7$ and $x^6 - 6x^4 + 9x^2 - 3$ is the minimal polynomial of any integer symmetric matrix. Our computations revealed that these three do not arise as characteristic polynomials, nor as minimal polynomials for any 12-by-12 or 18-by-18 matrix. Moreover the smallest span of an indecomposable 19-by-19 matrix is already larger than the spans of all three of these degree-6 polynomials, so by interlacing they cannot appear as the minimal polynomial of any larger matrix.

It remains an open problem as to whether or not there exists a degree-5, monic, separable, totally real polynomial that does not arise as the minimal polynomial of an integer symmetric matrix. All the small-span cases are covered, so the techniques of this paper cannot be applied.

Acknowledgments

This work was prompted by conversations with Georges Rhin and Chris Smyth at a workshop on Discovery and Experimentation in Number Theory, at the

Fields Institute, Toronto in September 2009: I am grateful to the organisers of that workshop. I have also benefited from conversations with Gary Greaves. Finally, I thank the referees for their numerous helpful suggestions.

References

1. Bass, H., Guralnick, R., Estes, D.: Eigenvalues of symmetric matrices and graphs. J. Algebra 168, 536–567 (1994)
2. Capparelli, S., Del Fra, A., Sciò, C.: On the span of polynomials with integer coefficients. Math. Comp. 79, 967–981 (2010)
3. Cauchy, A.: Sur l'équation a l'aide de laquelle on determine les inégalités séculaires des mouvements des planètes. In: Oeuvres Complètes d' Augustin Cauchy Seconde Série IX, pp. 174–195. Gauthier-Villars, Berkeley (1891)
4. Dobrowolski, E.: A note on integer symmetric matrices and Mahler's measure. Canadian Mathematical Bulletin 51(1), 57–59 (2008)
5. Estes, D.: Eigenvalues of symmetric integer matrices. J. Number Theory 42, 292–296 (1992)
6. Estes, D.R., Guralnick, R.M.: Minimal polynomials of integral symmetric matrices. Linear Algebra and its Applications 192, 83–99 (1993)
7. Fisk, S.: A very short proof of Cauchy's interlace theorem. Amer. Math. Monthly 112, 118 (2005)
8. Godsil, C., Royle, G.: Algebraic Graph Theory. In: Graduate Texts in Mathematics, vol. 207. Springer, New York (2000)
9. Hartley, B., Hawkes, T.O.: Rings, modules and linear algebra. Chapman and Hall, Boca Raton (1970)
10. Hwang, S.-G.: Cauchy's interlace theorem for eigenvalues of Hermitian matrices. Amer. Math. Monthly 112, 157–159 (2004)
11. Kronecker, L.: Zwei sätse über gleichungen mit ganzzahligen coefficienten. J. Reine Angew. Math. 53, 173–175 (1857)
12. Lehmer, E.: A numerical function applied to cyclotomy. Bull. Amer. Math, Soc. 36, 291–298 (1930)
13. McKee, J.F., Smyth, C.J.: Integer symmetric matrices having all their eigenvalues in the interval $[-2, 2]$. J. Algebra 317, 260–290 (2007)
14. McKee, J.F., Smyth, C.J.: Integer symmetric matrices of small spectral radius and small Mahler measure, arXiv:0907.0371v1
15. Batut, C., Belebas, K., Bernardi, D., Cohen, H., Olivier, M.: PARI/GP version 2.3.4, http://pari.math.u-bordeaux.fr/
16. Petrović, M.M.: On graphs whose spectral spread does not exceed 4. Publ. Inst. Math. Beograd 34(48), 169–174 (1983)
17. Robinson, R.M.: Intervals containing infinitely many sets of conjugate algebraic integers. In: Mathematical Analysis and Related Topics: Essays in Honor of George Pólya, Stanford, pp. 305–315 (1962)
18. Robinson, R.M.: Algebraic equations with span less than 4. Math. Comp. 18(88), 547–559 (1964)
19. Schur, I.: Über die Verteilung der Wurzeln bei gewissen algebraischen Gleichungen mit ganzzahligen Koeffizienten. Math. Z. 1, 377–402 (1918)

Decomposition Attack for the Jacobian of a Hyperelliptic Curve over an Extension Field

Koh-ichi Nagao

Dept. of Engineering, Kanto Gakuin Univ.,
1-50-1 Mutsuura Higashi Kanazawa-ku Yokohama 236-8501, Japan
nagao@kanto-gakuin.ac.jp

Abstract. We propose some kind of new attack which gives the solution of the discrete logarithm problem for the Jacobian of a curve defined over an extension field $\mathbb{F}_{q^n}$, considering the set of the union of factor basis and large primes B_0 given by points of the curve whose x-coordinates lie in $\mathbb{F}_q$. In this attack, an element of the divisor group which is written by a sum of some elements of factor basis and large primes is called (potentially) decomposed and the set of the factors that appear in the sum, is called decomposed factors. So, it will be called decomposition attack. In order to analyze the running of the decomposition attack, a test for the (potential) decomposedness and the computation of the decomposed factors are needed. Here, we show that the test to determine if an element of the Jacobian (i.e., reduced divisor) is written by an ng sum of the elements of the decomposed factors and the computation of decomposed factors are reduced to the problem of solving some multivariable polynomial system of equations by using the Riemann-Roch theorem. In particular, in the case of hyperelliptic curves of genus g, we construct a concrete system of equations, which satisfies these properties and consists of $(n^2 - n)g$ quadratic equations. Moreover, in the case of $(g, n) = (1, 3), (2, 2)$ and $(3, 2)$, we give examples of the concrete computation of the decomposed factors by using the computer algebra system Magma.

Keywords: Decomposition Attack, Hyperelliptic curve, Discrete logarithm problem, Weil descent attack.

1 Introduction

In this work, we treat the solution of the discrete logarithm problem of the Jacobian of a curve C of genus g defined over an extension field $\mathbb{F}_{q^n}$ $(n \geq 2)$ by decomposition attack. In particular, when C is a hyperelliptic curve and $ng(\geq 3)$ is a small integer, we give the concrete algorithm for computing what is called decomposed factors. In [6], Gaudry proposes the decomposition attack for the Jacobian of a hyperelliptic curve defined over a general finite field $\mathbb{F}_q$ considering a set of factor basis given by the $\mathbb{F}_q$-rational points of the curve. This attack is usually called 'Index Calculus' and such variations are widely used [3], [11]. However, the behavior of this attack, when it is used for solving the discrete

G. Hanrot, F. Morain, and E. Thomé (Eds.): ANTS-IX 2010, LNCS 6197, pp. 285–300, 2010.

logarithm of algebraic curve, is quite different to the original index calculus, which is a method to compute indices, that is, discrete logarithms in multiplicative groups of finite prime fields. Because of this, we use the name decomposition attack to refer to the attack. By recent works on the decomposition attack, which are the improvements of [6], it is known that the techniques of 1) using rebalancing [5] and 2) using large primes [15], [13], [7] are available. On the contrary, the techniques of large prime variations of normal index calculus associated to number field sieve are known as no contribution and do not lead to a decrease of the complexity.

In [8](also c.f. [4]), Gaudry also presents the decomposition attack for an elliptic curve defined over an extension field $\mathbb{F}_{q^n}$ considering the set of factor basis given by points of the curve whose x-coordinates lie in $\mathbb{F}_q$. Actually, Gaudry proposes also the rebalancing and the large prime variations. In these variations, the set of factor basis B is taken by some subset of B_0 which is given by points of the curve whose x-coordinates lie in $\mathbb{F}_q$ and an element in $B_0 \backslash B$ is called large prime. In these methods, the test for the potential decomposedness of $P \in E(\mathbb{F}_{q^n})$ (i.e., for being a sum of n elements of the B_0) and the computation of the decomposed factors (i.e., n elements of B_0 whose summation equals to P) are reduced to the problem of solving some system of multivariable polynomial equations of degree 2^{n-1}, n variables, and n equations, using Semaev's summation polynomials [14]. Moreover, Gaudry generalizes this decomposition attack to the case of the abelian varieties defined over an extension field, including the case of Jacobians of curves. However, in the case of non-elliptic curves, Semaev's summation polynomials are not available. It is, in principle, possible to derive a similar system of equations using the group law. Unfortunately, such is cumbersome. In fact, in the case of the Jacobian of a hyperelliptic curve of genus g, the sum of ng generic points is needed. Assuming that an element of Jacobian is written by the Mumford representation and that the group law is done by the Cantor algorithm [2], since the Cantor algorithm needs $g-1$ times reduction steps, explosions of the degree and terms occur in this computation.

In this work, we show that instead of using the group law, another system of equations is obtained from the theory of Riemann-Roch spaces (only in the case of Jacobians of curves). With this tool, the system of the equations is now simple to compute, and its parameters are easily controlled. In particular, in the case of Jacobians of hyperelliptic curves, this system of the equations consists of $(n^2-n)g$ quadratic equations in $(n^2-n)g$ indeterminates.

So, under the heuristic assumption that this system of the equations is (essentially) projectively 0-dimensional, the computational amount for solving this system of equations is estimated by $O(2^{(n^2-n)g \cdot C})$ where C is some constant less than 3. In the case of an elliptic curve (i.e., $g=1$), this computational amount heuristically equals to that of Gaudry's original equations system using Semaev's summation polynomials.

2 Decomposition Attack for the Jacobian of a General Plane Curve

This section adapts the idea of [8] to the setting of a smooth plane curve with a single missing point at infinity, and presents an overview of the decomposition attack for the Jacobian of a general plane curve using the Riemann-Roch theorem. Let C_a be the affine curve of genus g defined over an extension field $\mathbb{F}_{q^n}$ (i.e., $n \geq 2$) given by the equation $f(x, y) = 0$, and let C be the corresponding non-singular complete curve. Assume that C_a is non-singular. From this, we have a canonical embedding $\iota : C_a \to C$. It is also assumed that $C \backslash \iota(C_a)$ only consists of a single $\mathbb{F}_{q^n}$-valued point, which is denoted by ∞ and is called the point at infinity. These assumptions are true for hyperelliptic curves so there is no problem for the main results of this work. Let D_0 be a divisor of the form

$$D_0 = Q_1 + .. + Q_g - (g)\infty \tag{1}$$

where $Q_1, .., Q_g \in C(\overline{\mathbb{F}}_{q^n})$ and the multiset $\{Q_1, .., Q_g\}$ is stable under the action of galois group $\mathrm{Gal}(\overline{\mathbb{F}}_{q^n}/\mathbb{F}_{q^n})$. Put

$$\phi_1(x) := \prod_{i=1}^{g}(x - x(Q_i)) \tag{2}$$

and note that it is in $\mathbb{F}_{q^n}[x]$.

Also put

$$B_0 := \{P \in C \,|\, P = (x, y) \in C(\mathbb{F}_{q^n}),\, x \in \mathbb{F}_q\},$$

as a set of factor basis and large primes. (Strictly saying, B_0 must be a subset of $\mathrm{Jac}_C(\mathbb{F}_{q^n})$, and it is the set of the elements of the divisors $P - \infty$ where P has the above properties. Here, the term "$-\infty$" is omitted for simplicity.)

Assumption 1. *Let n be a fixed positive integer. Then the number of the multisets $\boldsymbol{P} = \{P_1, .., P_{ng}\}$ with $P_i \in B_0$, which satisfy the relation $\sum_{i=1}^{ng} P_i \sim \sum_{i=1}^{ng} P'_i$ for some different ($\boldsymbol{P} \neq \boldsymbol{P'}$) multiset $\boldsymbol{P'} = \{P'_1, .., P'_{ng}\}$ with $P'_i \in B_0$, is less than $q^{ng-\varepsilon}$, where ε is some positive constant.*

Here, we shortly state the validity of this assumption in the case of hyperelliptic curve. Let $C : y^2 = f(x)$ be the equation of hyperelliptic curve. For any $P = (x, y) \in C$, put $\bar{P} = (x, -y) \in C$. So, there are series of trivial relations $P + \bar{P} \sim P' + \bar{P}'$ for any $P, P' \in B_0$. The number of the multisets satisfying the condition of Assumption 1 and coming from these trivial relations is only $O(q^{ng-1})$ and it seems to be no series including many trivial relations. So, Assumption 1 seems to be valid.

Assumption 2. $|B_0| \approx q$.

Here, we also state the validity of this assumption in the case of hyperelliptic curve. Let $C : y^2 = f(x)$ be the equation of hyperelliptic curve. If $f(x)$ is chosen

randomly, the probability that $f(x)$ $(x \in \mathbb{F}_q)$ is square in $\mathbb{F}_{q^n}$ is around $1/2$ and this assumption seems to hold.

In the following, we assume Assumption 1 and Assumption 2. From these assumptions, we see easily that since "the number of the divisors of the form(1)"$\approx q^{gn}$, the probability that there are some $P_1, P_2, .., P_{ng} \in B_0$ (exactly ng elements, $P_i = P_j$ for some $i \neq j$ being allowed) such that

$$\begin{aligned} &D_0 + P_1 + P_2 + ... + P_{ng} - (ng)\infty \\ &\quad = \textstyle\sum_{i=1}^{g} Q_i + P_1 + P_2 + ... + P_{ng} - (ng+g)\infty \sim 0, \end{aligned} \tag{3}$$

is approximately $1/(gn)!$, when $q \gg ng$.

Definition 1. *If a divisor D_0 is written by the form (3) for some $P_1, P_2, .., P_{ng} \in B_0$ (exactly ng elements, $P_i = P_j$ for some $i \neq j$ being allowed), D_0 is called potentially decomposed and in this case, the elements $P_1, P_2, .., P_{ng}$ are called decomposed factors and the multiset $\{P_i\}_{i=1}^{ng}$ is called decomposed divisor.*

We now fix D_0 and discuss how it can be tested that D_0 is potentially decomposed and the decomposed factors can be computed. So, $Q_1, ..., Q_g$ and $\phi_1(x)$, which are dependent on D_0, are also fixed.

Let $D = \sum_{P \in C(\overline{\mathbb{F}}_{q^n})} n_p P$, $n_p \in \mathbb{Z}$ be a divisor of $C/\mathbb{F}_{q^n}$. Assume that D is stable under the action of galois group $\mathrm{Gal}(\overline{\mathbb{F}}_{q^n}/\mathbb{F}_{q^n})$. Put $\deg(D) := \sum_{P \in C(\overline{\mathbb{F}}_{q^n})} n_p$, and $L(D) := \{f \in \mathbb{F}_{q^n}(C) \,|\, (f) + D \geq 0\} \cup \{0\}$. From the Riemann-Roch theorem (cf [10] Corollary A.4.2.3), we have the following lemma.

Lemma 1. *(**Riemann-Roch**) 1) $L(D)$ is an $\mathbb{F}_{q^n}$ vector space.*
2) If $\deg(D) \geq 2g-1$, $\dim L(D) = \deg(D) - g + 1$.

From this Lemma, $\dim L((ng)\infty - D_0) = \dim L((ng+g)\infty - \sum_{i=1}^{g} Q_i) = ng - g + 1$. Let $\{f_0(x,y), f_1(x,y), ..., f_{ng-g}(x,y)\}$ be a base of $L((ng)\infty - D_0))$ and an element $h \in L((ng)\infty - D_0)$ is written by

$$a_0 f_0(x,y) + a_1 f_1(x,y) + ... + a_{ng-g} f_{ng-g}(x,y) \tag{4}$$

where a_i are values in $\mathbb{F}_{q^n}$. From Hess [9], we have the following lemma.

Lemma 2. *A base of $L((ng)\infty - D_0)$ is computable within Poly($ng \log q$) time.*

Let

$$h(x,y) := A_0 f_0(x,y) + A_1 f_1(x,y) + ... + A_{ng-g} f_{ng-g}(x,y) \tag{5}$$

be a multivariable polynomial in $\mathbb{F}_{q^n}[A_0, ..., A_{ng-g}, x, y]$.

For

$$\boldsymbol{a}_{\mathrm{aff}} = (a_0, a_1, ..., a_{ng-g}) \in \mathbb{A}^{ng-g+1}(\mathbb{F}_{q^n})$$

and some polynomial $p(x) \in \mathbb{F}_{q^n}[A_0, ..., A_{ng-g}, x]$, let $p_{\boldsymbol{a}_{\mathrm{aff}}}(x)$ be the polynomial obtained from $p(x)$ by substituting a_i for A_i.

Definition 2. *A multivariable polynomial $p(x)$ in $\mathbb{F}_{q^n}[A_0, ..., A_{ng-g}, x]$ is called A-homogenous, when $p_{\boldsymbol{a}_{aff}}(x) = Const \times p_{k\boldsymbol{a}_{aff}}(x)$ holds for all $\boldsymbol{a}_{aff} = (a_0, a_1, ..., a_{ng-g}) \in \mathbb{A}^{ng-g+1}(\mathbb{F}_{q^n})$ and $k \in \mathbb{F}_{q^n}^*$.*

For

$$\boldsymbol{a}_{\mathrm{pro}} = (a_0, a_1, ..., a_{ng-g}) \in \mathbb{P}^{ng-g}(\mathbb{F}_{q^n})$$

and some A-homogenous polynomial $p(x) \in \mathbb{F}_{q^n}[A_0, ..., A_{ng-g}, x]$,
let $\mathrm{monic}(p_{\boldsymbol{a}_{\mathrm{aff}}}(x))$ be the polynomial obtained from $p(x)$ by substituting a_i for A_i and dividing by the leading coefficient. Now, we compute the intersections of $h_{\boldsymbol{a}_{\mathrm{pro}}}(x, y) = 0$ on C. Remember that the equation of C_a is $f(x, y) = 0$.
Put $S(x) := \mathrm{Resultant}_y(f(x, y), h(x, y))$. From this construction, we then have the following lemma.

Lemma 3.
1) $S(x)$ is a multivariable A-homogeneous polynomial in $\mathbb{F}_{q^n}[A_0, .., A_{ng-g}, x]$.
2) $deg_x S(x) = ng + g$.
3) $\phi_1(x) \mid S(x)$.

Proof. 1) is trivial. For any $\boldsymbol{a}_{\mathrm{pro}} = (a_0, a_1, ..., a_{ng-g}) \in \mathbb{P}^{ng-g}(\mathbb{F}_{q^n})$, since $h_{\boldsymbol{a}_{\mathrm{pro}}}(x, y)$ has only poles $(ng + g)\infty$ on points at infinity, we have 2) and since $h_{\boldsymbol{a}_{\mathrm{pro}}}(x, y)$ have zeros at each Q_i's, we have 3).

Put $g(x) := S(x)/\phi_1(x)$. Since $\phi_1(x) \in \mathbb{F}_{q^n}[x]$, $g(x)$ is also a multivariable A-homogeneous polynomial in $\mathbb{F}_{q^n}[A_0, .., A_{ng-g}, x]$. Thus, $g(x)$ is written in the form

$$g(x) = C_{ng}x^{ng} + C_{ng-1}x^{ng-1} + ... + C_0$$

where each $C_i \in \mathbb{F}_{q^n}[A_0, .., A_{ng-g}]$ has the same multi degree of A_i. Note that if the indeterminates $A_i's$ are replaced by values a_i and the obtained polynomial is divided by the leading coefficient, then one obtains a polynomial $\mathrm{monic}(g_{\boldsymbol{a}_{\mathrm{pro}}}(x))$ in $\mathbb{F}_{q^n}[x]$. The solutions of $\mathrm{monic}(g_{\boldsymbol{a}_{\mathrm{pro}}}(x)) = 0$ mean the x-coordinates of the intersections $h_{\boldsymbol{a}_{\mathrm{pro}}}(x, y) = 0$ on C except $Q_1, ..., Q_g$. So, we have the following lemma.

Lemma 4. *The condition that D_0 is potentially decomposed is equivalent to the following: There is some $\boldsymbol{a}_{pro} = (a_0, a_1, ..., a_{ng-g}) \in \mathbb{P}^{ng-g}(\mathbb{F}_{q^n})$ such that $monic(g_{\boldsymbol{a}_{pro}}(x)) \in \mathbb{F}_q[x]$ and $monic(g_{\boldsymbol{a}_{pro}}(x)) \in \mathbb{F}_q[x]$ factors completely in $\mathbb{F}_q[x]$.*

Now, we find such a_i's. Let $[\alpha_0(= 1), \alpha_1, .., \alpha_{n-1}]$ be a base of $\mathbb{F}_{q^n}/\mathbb{F}_q$. We fix this base. Let $A_{i,j}$ $(1 \le i \le ng, 0 \le j \le n - 1)$ be new indeterminates over $\mathbb{F}_q$, and let us consider the polynomials obtained by substituting A_0 by 1 and A_i by $\sum_{j=0}^{n-1} A_{i,j}\alpha_j$ $(1 \le i \le ng - g)$ in $g(x)$. Let us denote the coefficients obtained in this way again by C_i. Then the coefficients can be written in the form

$$C_i = \sum_{j=0}^{n-1} C_{i,j}\alpha_j, \quad C_{i,j} \in \mathbb{F}_q[\cup_{1 \le i \le ng,\, 0 \le j \le n-1}\{A_{i,j}\}].$$

Then, the condition that there is some $\boldsymbol{a}_{\mathrm{pro}} \in \mathbb{P}^{ng-g}(\mathbb{F}_{q^n})$ satisfying
1) $\mathrm{monic}(g_{\boldsymbol{a}_{\mathrm{pro}}}(x)) \in \mathbb{F}_q[x]$ and

2) First coordinate of $\boldsymbol{a}_{\mathrm{pro}}$ is non-zero,
is equivalent to the condition that the system of the equations

$$C_{i,j} = T_i C_{ng,j} \qquad (0 \leq i \leq ng-1, 0 \leq j \leq n-1) \tag{6}$$

of $(n^2+n)g$ indeterminates $\cup\{A_{i,j}\}$ and $T_0, ..., T_{ng-1}$ defined over $\mathbb{F}_q$ has some solutions $A_{i,j} = a_{i,j}, T_i = t_i$ in $\mathbb{F}_q$. In this case, monic$(g_{\boldsymbol{a}_{\mathrm{pro}}}(x))$ is written by

$$x^{ng-g} + t_{ng-g-1}x^{ng-g-1} + ... + t_1 x + t_0. \tag{7}$$

Thus, the test of the decomposedness of D_0 and the computation of the decomposed factors are reduced to find the solutions of the system of the equations (6) and factorizations of the polynomials (7).

In the next section, we will investigate the case of the hyperelliptic curve. In this case, there is a concrete representation of the Riemann-Roch space, and so we have a more concrete system of equations.

3 Decomposition Attack for the Jacobian of a Hyperelliptic Curve

Now, we discuss the special case of Jacobians of hyperelliptic curves. In this case, there are concrete representations of the Riemann-Roch space and some techniques that $g(x)$ can be taken as a monic polynomial, and from this, a simple system of equations is derived. Let C be a hyperelliptic curve (including an elliptic curve) of genus g of the form

$$C : y^2 = f(x), \text{ where } f(x) = x^{2g+1} + a_{2g}x^{2g} + ... + a_0$$

over $\mathbb{F}_{q^n}$ where the characteristic of $\mathbb{F}_q$ is not 2 and $n \geq 2$. Put ∞ by the unique point at infinity on C. Let D_0 be a reduced divisor (i.e.,$\mathbb{F}_{q^n}$-rational point of the Jacobian) of C. To represent D_0, we use the so-called Mumford representation:

$$D_0 = (\phi_1(x), \phi_2(x)),$$

where $\phi_1(x) \in \mathbb{F}_{q^n}[x]$ is a monic polynomial with $\deg(\phi_1(x)) \leq g$ and $\phi_2(x) \in \mathbb{F}_{q^n}[x]$ satisfies $\deg(\phi_2(x)) < \deg(\phi_1(x))$ and $f(x) - \phi_2(x)^2 \equiv 0 \bmod \phi_1(x)$. In the following, we will assume $\deg(\phi_1(x)) = g$. This assumption holds for all but a negligible fraction of divisor classes D_0. Note that there are $Q_1, .., Q_g \in C(\overline{\mathbb{F}}_{q^n})\backslash\{\infty\}$ satisfying the equation (1) and the multiset $\{Q_1, .., Q_g\}$ is stable under the action of galois group $\mathrm{Gal}(\overline{\mathbb{F}}_{q^n}/\mathbb{F}_{q^n})$.

Similarly, put $B_0 := \{P \in C \,|\, P = (x, y) \in C(\mathbb{F}_{q^n}), x \in \mathbb{F}_q\}$ as a set of factor basis and large primes. Then, from the Assumption 1 and Assumption 2, we can see easily that the probability, that there are some $P_1, P_2, .., P_{ng} \in B_0$ (exactly ng elements, $P_i = P_j$ for some $i \neq j$ being allowed) satisfying the equation (3), is approximately $1/(gn)!$, when $q \gg ng$.

In the following, we fix a reduced divisor D_0. So, $\phi_1(x)$, $\phi_2(x)$, and $Q_1, ..., Q_g$, which are dependent on D_0, are also fixed.

In this work, we show the following theorem.

Theorem 1. *Let $V_1, V_2, ..., V_{(n^2-n)g}$ be indeterminates and let D_0 be a reduced divisor of $C/\mathbb{F}_{q^n}$. Then there are some computable degree 2 polynomials*

$$C_{i,j} \in \mathbb{F}_q[V_1, V_2, ..., V_{(n^2-n)g}] \quad (0 \leq i \leq ng-1,\ 0 \leq j \leq n-1)$$

satisfying the following: The condition that D_0 is potentially decomposed is equivalent to the following 1) and 2):
1) The system of equations $\{C_{i,j} = 0 \,|\, 0 \leq i \leq ng-1,\ 1 \leq j \leq n-1\}$ has some solution $\boldsymbol{v} = (v_1, .., v_{(n^2-n)g}) \in \mathbb{A}^{(n^2-n)g}(\mathbb{F}_q)$.
2) Put $c_i = C_{i,0}(v_1, .., v_{(n^2-n)g})$ for $0 \leq i \leq ng-1$. Then $G(x) = x^{ng} + c_{ng-1}x^{ng-1} + ... + c_0 \in \mathbb{F}_q[x]$ factors completely.
Moreover, if D_0 is potentially decomposed, the x-coordinates of the decomposed factors are the solutions of $G(x) = 0$.

From this theorem, the test, whether D_0 is potentially decomposed and the computation of the decomposed factors (if possible), is reduced to solving the system of the equations $\{C_{i,j} = 0 \,|\, 0 \leq i \leq ng-1,\ 1 \leq j \leq n-1\}$ and factorizing the polynomials $G(x)$ obtained form the solutions of the system of these equations.

In the following, we construct such multivariable polynomials $\{C_{i,j}\}$ and show Theorem 1.

From the equation of C, we see $\mathrm{ord}_\infty x = 2$, and $\mathrm{ord}_\infty y = 2g+1$. Put $N_1 := \lfloor \frac{(n+1)g}{2} \rfloor$ and $N_2 := \lfloor \frac{ng-g-1}{2} \rfloor$.

Lemma 5. *1) $N_1 + N_2 = ng - 1$.*
2) $N_2 + g - 1 < N_1$.

Proof. Trivial.

Lemma 6. *$\{1, x, x^2, .., x^{N_1}, y, xy, ...x^{N_2}y\}$ is a base of $L((ng+g)\infty)$.*

Proof. From $\mathrm{ord}_\infty x = 2$, $\mathrm{ord}_\infty y = 2g+1$, each element in the above list is in $L((ng+g)\infty)$. The independence is from the definition of the hyperelliptic curve. Thus, since the number of the elements of the list $N_1 + N_2 + 2 = ng + 1$ is the same as the $\dim L((ng+g)\infty)$ (from Lemma 1), we finish the proof.

Lemma 7
$\{\phi_1(x), \phi_1(x)x, ..., \phi_1(x)x^{N_1-g}, (y-\phi_2(x)), (y-\phi_2(x))x, ..., (y-\phi_2(x))x^{N_2}\}$ is a base of $L((ng)\infty - D_0) = L((ng+g)\infty - \sum_{i=1}^{g} Q_i)$.

Proof. From the definition of $\phi_1(x)$ and $\phi_2(x)$, each element in the list has a zero at each Q_i. Since $\deg(\phi_1(x)) = g$, $\deg(\phi_2(x)) \leq g-1$, and $N_2 + g - 1 < N_1$(from Lemma 5), each element in the list has at most $(ng+g)$ poles at ∞. Then they are in $L((ng)\infty - D_0)$. Now, we show the independence. Assume they are not independent, and there are some non zero $f_1(x), f_2(x) \in \mathbb{F}_{q^n}[x]$ such that $\phi_1(x)f_1(x) + (y-\phi_2(x))f_2(x) = 0$. However, the relation $\phi_1(x)f_1(x) + (y-\phi_2(x))f_2(x) = 0$ induces $yf_2(x) \in \mathbb{F}_{q^n}[x]$ and $f_1(x) = f_2(x) = 0$. As this is a contradiction, they are independent. On the other hand, the number of the elements in the list is $N_1 + N_2 + 2 - g = ng - g + 1$ from Lemma 5, which is the same as the $\dim L((ng)\infty - D_0)$. So we finish the proof.

From Lemma 7, an element $h \in L((ng)\infty - D_0)$ is written by

$$h(x,y) = \phi_1(x)(a_0 + a_1 x + ... + a_{N_1-g}x^{N_1-g}) + (y - \phi_2(x))(b_0 + b_1 x + ... + b_{N_2}x^{N_2}) \tag{8}$$

where a_i,b_i are values in $\mathbb{F}_{q^n}$.

Lemma 8. *Let $h(x,y) \in L((ng)\infty - D_0)$. Assume div(h(x,y)) is written in the form $P_1 + P_2 + ... + P_{ng} + \sum_{i=1}^{g} Q_i - (ng+g)\infty$ for $P_i \in C(\mathbb{F}_{q^n})\backslash\{\infty\}$. Then we have the following:*
1) $a_{N_1-g} \neq 0$ when $ng+g$ is even.
2) $b_{N_2} \neq 0$ when $ng+g$ is odd.

Proof. When $ng+g$ is even, assume $a_{N_1-g} = 0$, thus we have the order of the pole of $h(x,y)$ at ∞ being truly less than $ng+g$ and div$(h(x,y))$ is not written by the form of (3). Similarly, when $ng+g$ is odd, assume $b_{N_2} = 0$. Thus we have the order of the pole of $h(x,y)$ at ∞ being truly less than $ng+g$ and div$(h(x,y))$ is not written by the form of (3). So, we can assume that $a_{N_1-g} \neq 0$, if $ng+g$ is even, and $b_{N_2} \neq 0$, if $ng+g$ is odd.

Now, we compute the intersections of $h(x,y) = 0$ on C. For this purpose, y must be eliminated. Note that the point (x,y) fulfills $h(x,y) = 0$, if and only if the equation

$$y = \frac{-\phi_1(x)(a_0 + a_1 x + ... + a_{N_1-g}x^{N_1-g}) + \phi_2(x)(b_0 + b_1 x + ... + b_{N_2}x^{N_2})}{b_0 + b_1 x + ... + b_{N_2}x^{N_2}}. \tag{9}$$

holds. By this y's representation, the number of the parameters must be decreased. So, put $a_{N_1-g} = 1$ when $ng+g$ is even and put $b_{N_2} = 1$ when $ng+g$ is odd (this can be done from the above lemma). Also put $M_1 = \begin{cases} N_1 - g - 1 \text{ when } ng+g \text{ is even} \\ N_1 - g \quad \text{when } ng+g \text{ is odd} \end{cases}$, and $M_2 = \begin{cases} N_2 \quad \text{when } ng+g \text{ is even} \\ N_2 - 1 \text{ when } ng+g \text{ is odd} \end{cases}$. Note that $M_1 + M_2 = ng - g - 2$ from Lemma 5.

Put

$$s(x) := \begin{cases} -(\text{denominator of (9)})^2 f(x) + (\text{numerator of (9)})^2, \text{ if } ng+g \text{ is even} \\ (\text{denominator of (9)})^2 f(x) - (\text{numerator of (9)})^2, \quad \text{if } ng+g \text{ is odd} \end{cases}.$$

and let $S(x)$ be the multivariable polynomial obtained from the definition of $s(x)$ replacing the values a_i and b_i by the indeterminates A_i and B_i. From the construction, $S(x)$ is a monic polynomial of the degree $ng+g$, whose coefficients are degree 2 polynomials in $\mathbb{F}_{q^n}[A_0, .., A_{M_1}, B_0, .., B_{M_2}]$, and $\phi_1(x)|S(x)$. Put $g(x) := S(x)/\phi_1(x)$. Since $\phi_1(x)$ is a monic polynomial in $\mathbb{F}_{q^n}[x]$, $g(x)$ is also a monic polynomial of degree ng, whose coefficients are degree 2 polynomials in $\mathbb{F}_{q^n}[A_0, .., A_{M_1}, B_0, .., B_{M_2}]$. Put $C_i \in \mathbb{F}_{q^n}[A_0, .., A_{M_1}, B_0, .., B_{M_2}]$ by i-th coefficient of $g(x)$, i.e.,

$$g(x) = x^{ng} + C_{ng-1}x^{ng-1} + ... + C_0.$$

Similarly, for

$$\boldsymbol{v} = (a_0, ...a_{M_1}, b_0, ..., b_{M_2}) \in \mathbb{A}^{M_1+M_2+2}(\mathbb{F}_{q^n})$$

and some polynomial $p(x)$ in $\mathbb{F}_{q^n}[A_0, ..., A_{M_1}, B_{M_0}, ..., B_{M_2}, x]$, let $p_{\boldsymbol{v}}(x)$ be the polynomial obtained from $p(x)$ by substituting a_i and b_i for A_i and B_i. Then, the zeros of $g_{\boldsymbol{v}}(x) = 0$ are the x-coordinate of the intersections of $h(x, y) = 0$ on C except $Q_1, ..., Q_g$. Thus, we have the following lemma.

Lemma 9. *The condition that D_0 is a potentially decomposed reduced divisor is equivalent to the following:*
There is some $\boldsymbol{v} = (a_0, .., a_{M_1}, b_0, ...b_{M_2}) \in \mathbb{A}^{M_1+M_2+2}(\mathbb{F}_{q^n})$ such that $g_{\boldsymbol{v}}(x) \in \mathbb{F}_q[x]$ and $g_{\boldsymbol{v}}(x) \in \mathbb{F}_q[x]$ factors completely in $\mathbb{F}_q[x]$.

We now show how to find a_i in $\mathbb{F}_{q^n}$ $(0 \le i \le M_1)$ and b_i in $\mathbb{F}_{q^n}$ $(0 \le i \le M_2)$ such that $g_{\boldsymbol{v}}(x)$ in $\mathbb{F}_q[x]$.

Let $[\alpha_0(= 1), \alpha_1, .., \alpha_{n-1}]$ be a base of $\mathbb{F}_{q^n}/\mathbb{F}_q$ and fix this base. Let $A_{i,j}$ $(0 \le i \le M_1, 0 \le j \le n-1)$ and $B_{i,j}$ $(0 \le i \le M_2, 0 \le j \le n-1)$ be new indeterminates over $\mathbb{F}_q$. Note that the number of the indeterminates $\{A_{i,j}\} \cup \{B_{i,j}\}$ is

$$(M_1 + M_2 + 2)n = (N_1 + N_2 - g + 1)n = (n^2 - n)g.$$

For simplicity, substitute the variables $A_{i,j}$ $(0 \le i \le M_1, 0 \le j \le n-1)$ and $B_{i,j}$ $(0 \le i \le M_2, 0 \le j \le n-1)$ by $\{V_1, V_2, ..., V_{(n^2-n)g}\}$. Let us consider the polynomials obtained by substituting A_i by $\sum_{j=0}^{n-1} A_{i,j}\alpha_j$ and B_i by $\sum_{j=0}^{n-1} B_{i,j}\alpha_j$ in $g(x)$. Also let us denote the coefficients obtained in this way again by C_i. Then the coefficients can be written in the form

$$C_i = \sum_{j=0}^{n-1} C_{i,j}\alpha_j, \quad C_{i,j} \in \mathbb{F}_q[V_1, V_2, ..., V_{(n^2-n)g}].$$

Thus from Lemma 9, the condition $g_{\boldsymbol{v}}(x) \in \mathbb{F}_q[x]$ is equivalent to the condition that there are some $v_1, v_2, ..., v_{(n^2-n)g} \in \mathbb{F}_q$ such that

$$C_{i,j}(v_1, v_2, ..., v_{(n^2-n)g}) = 0 \text{ for } 0 \le i \le ng-1,\ 1 \le j \le n-1.$$

Moreover, when $g_{\boldsymbol{v}}(x) \in \mathbb{F}_q[x]$, $g(x) = x^{ng} + C_{ng-1,0}x^{ng-1} + ... + C_{0,0}$. The condition that $g_{\boldsymbol{v}}(x)$ factors completely in $\mathbb{F}_q[x]$ is equivalent to the above condition, and $G(x) := x^{ng} + c_{ng-1}x^{ng-1} + ... + c_0$ factors completely in $\mathbb{F}_q[x]$ where $c_i = C_{i,0}(v_1, v_2, ..., v_{(n^2-n)g})$. In this case, the solutions of $G(x) = 0$ are the x-coordinates of the decomposed factor. Then, we finish the proof of proposition 1 and construct the equation system $\{C_{i,j} = 0\}$.

4 Example

In this section, we examine three computational experiments of the decomposed factors of Jacobian. The computations are done by using the computer algebra

system magma on a Windows XP preinstalled PC (CPU:Pentium M 2GHz, RAM:1GB). (In order to solve equation system, the function "variety" prepared in magma is used.) We compute three cases 1) $(g,n)=(1,3)$, 2) $(g,n)=(2,2)$, and 3) $(g,n)=(3,2)$ where g and n are the genus and the extension degree of the definition field of the chosen hyperelliptic/elliptic curve, respectively. In all cases, one trial, which means the judge as to whether a given element of Jacobian is decomposed or not and compute its decomposed factor, if it is decomposed, is done within 1 second. Since the probability that an element of Jacobian is decomposed is approximately $1/(gn)!$, the amount of the time for obtaining one potentially decomposed reduced divisor is within 6 sec, 24 sec, and 720 sec, respectively. Further, we will give the following three examples.

Case 1. Let $q=1073741789$(prime number), $\mathbb{F}_{q^3}:=\mathbb{F}_q[t]/(t^3+456725524t^2+251245663t+746495860)$, and let $E/\mathbb{F}_{q^3}$ be an elliptic curve defined by $y^2=x^3+(1073741788t^2+t)x+(126t+3969)$ and $P_0:=(t,t+63)\in E$. We investigate whether $nP_0:n=1,2,..30$ are decomposed and find the following 7 decompositions. ($24P_0$ is written by 2 forms.)

$$\begin{aligned}
2P_0 &= (1050861583, 6509843t^2+387051565t+920296030)\\
&\quad +(742900894, 362262801t^2+6480079t+886701711)\\
&\quad +(571975376, 938916909t^2+910769097t+139897863)\\
5P_0 &= (806296922, 113931706t^2+863383473t+133427995)\\
&\quad +(797256157, 360646567t^2+663390692t+1012046566)\\
&\quad +(389333914, 986077188t^2+829314065t+687783827)\\
8P_0 &= (1063441336, 113661172t^2+942865616t+744283566)\\
&\quad +(894045278, 863335768t^2+637284565t+937810737)\\
&\quad +(694935460, 740353309t^2+505910431t+597402219)\\
20P_0 &= (996570058, 341336613t^2+450680674t+72874200)\\
&\quad +(141768271, 589122734t^2+930205049t+713557032)\\
&\quad +(73505168, 432994198t^2+405986289t+233154172)\\
24P_0 &= (529735815, 20343700t^2+780030904t+490121669)\\
&\quad +(515960254, 269821984t^2+561547517t+348990487)\\
&\quad +(207183771, 712543643t^2+356522343t+895634732)\\
&= (818683055, 1034251164t^2+705927333t+1062879754)\\
&\quad +(754504105, 23461217t^2+961620879t+1015889110)\\
&\quad +(489159707, 271295793t^2+600348670t+1022482426)\\
26P_0 &= (628174301, 138296704t^2+104824480t+858118320)\\
&\quad +(371888603, 417445284t^2+850151153t+126970733)\\
&\quad +(55411433, 560274594t^2+609956706t+821692494)
\end{aligned}$$

Case 2. Let $q=1073741789$(prime number), $\mathbb{F}_{q^2}:=\mathbb{F}_q[t]/(t^2+746495860t+206240189)$, and let $C/\mathbb{F}_{q^2}$ be a hyperelliptic curve defined by

$$y^2=x^5+(673573223t+771820244)x+6t+9$$

and let

$$\begin{aligned}
D_0 := (&x^2+1073741787tx+327245929t+867501600,\\
&(1023168391t+350252228)x+658555356t+446913597)
\end{aligned}$$

be a reduced divisor of C. We investigate whether $nD_0 : n = 1, 2, ..100$ are decomposed and find the following 9 decompositions. ($71D_0$ is written by 2 forms.)

$6D_0 \sim (1025731975, 776505688t + 911495013) + (728060789, 648475468t + 1067025179)$
$+ (341799975, 145077925t + 187604034) + (61964999, 227570631t + 639782700) - 4\infty$
$19D_0 \sim (1039361498, 15180988t + 396695374) + (828360115, 179412594t + 719919461)$
$+ (483171045, 677645208t + 604714840) + (34566209, 753841024t + 14375633) - 4\infty$
$33D_0 \sim (970690833, 608141084t + 889165804) + (260086243, 894605411t + 261264640)$
$+ (208957980, 43330622t + 581461318) + (190782894, 124873649t + 510328990) - 4\infty$
$35D_0 \sim (699447787, 267523741t + 562899544) + (559470007, 197827114t + 99971197)$
$+ (472594781, 579187919t + 266558458) + (453661772, 449424806t + 977318920) - 4\infty$
$48D_0 \sim (1009979214, 959734525t + 990871450) + (995813251, 44186049t + 288496638)$
$+ (521299995, 556594200t + 468424666) + (17946008, 977064852t + 1071618742) - 4\infty$
$71D_0 \sim (1019155056, 573896856t + 103042116) + (944470217, 829781939t + 184620624)$
$+ (727156004, 462612591t + 582877732) + (281900623, 553507533t + 42660552) - 4\infty$
$\sim (502979299, 412632304t + 1036827718) + (74527656, 927651409t + 452588110)$
$+ (50078888, 801072540t + 888737005) + (2986754, 556402789t + 236723678) - 4\infty$
$73D_0 \sim (843747137, 682161676t + 600252618) + (829302257, 145878028t + 853397395)$
$+ (290487906, 645896278t + 279001181) + (184873704, 567002729t + 620354511) - 4\infty$
$80D_0 \sim (907811987, 216534804t + 936839244) + (808513243, 873487475t + 273845273)$
$+ (520893378, 757248670t + 381150138) + (486203744, 494475019t + 791571132) - 4\infty$

Case 3. Let $q = 1073741789$(prime number), $\mathbb{F}_{q^2} := \mathbb{F}_q[t]/(t^2 + 746495860t + 206240189)$, and let $C/\mathbb{F}_{q^2}$ be a hyperelliptic curve defined by

$$y^2 = x^7 + (111912375t + 1046743132)x + 6t + 9$$

and let

$$D_0 := (x^2 + 1073741787tx + 327245929t + 867501600, \\ (473621736t + 256126568)x + 145989647t + 687383736)$$

be a reduced divisor of C. We investigate whether $nD_0 : n = 1, 2, ..3000$ are decomposed and find the following 6 decompositions.

$414D_0 \sim (1001437837, 752632260t + 700158497) + (747112084, 656073918t + 400137619)$
$+ (620249588, 127943213t + 635474623) + (614180498, 206297635t + 445250468)$
$+ (515769009, 607297126t + 554290493) + (488549466, 627952783t + 854182612) - 6\infty$
$657D_0 \sim (939617127, 695261735t + 239531611) + (933351280, 935312661t + 961494096)$
$+ (799612924, 341923983t + 677495100) + (294787599, 279723229t + 760003067)$
$+ (273118782053704103t + 577497766) + (153381525, 983211238t + 517037777) - 6\infty$
$921D_0 \sim (1034634787, 400751409t + 829801342) + (763888873, 757155774t + 829936954)$
$+ (619620874, 800641683t + 200272230) + (603032615, 115219564t + 655011145)$
$+ (436423191, 285214454t + 450812747) + (125198811, 884750621t + 123305741) - 6\infty$
$1026D_0 \sim (1024020017, 267457905t + 41452942) + (794174628, 615676821t + 723336407)$
$+ (738567269, 433647609t + 128304659) + (629287731, 465842490t + 789390318)$
$+ (435082408, 878213106t + 603353206) + (79621979, 479459622t + 672937516) - 6\infty$
$1121D_0 \sim (764081031, 812350603t + 347878564) + (673426715, 687737442t + 381588704)$
$+ (61025220820071 39t + 99219637) + (467560104, 619342780t + 228756808)$
$+ (179787786, 333322906t + 75482151) + (59221667, 860686653t + 625301206) - 6\infty$
$2289D_0 \sim (729358563, 482925408t + 170057124) + (529840657, 42328987t + 857983002)$

$$+ (514618236, 436901100t + 416530686) + (350106356, 183495333t + 950710579)$$
$$+ (175898979, 411808870t + 427518366) + (96240558, 703780413t + 461022225) - 6\infty$$

5 Conclusion

In this manuscript, we have proposed an algorithm which checks whether a reduced divisor is potentially decomposed or not, and we have computed the decomposed factors, if it is potentially decomposed. From this algorithm, concrete computations of decomposed factors are done by computer experiments when the pairs of the genus of the hyperelliptic curve and the degree of extension field are $(1, 3), (2, 2)$, and $(3, 2)$.

Acknowledgment

The author would like to thank Professor Kazuto Matsuo in the Institute of Information Security for useful comments and fruitful discussions and Professor Lisa Bond in Kanto Gakuin University for English writing. Also, the author would like to thank the anonymous reviewers who pointed out many mistakes and suggested a revisal plan.

References

1. Adleman, M., DeMarrais, J., Huang, M.-D.: A subexponential algorithm for discrete logarithms over the rational subgroup of the Jacobians of large genus hyperelliptic curves over finite fields. In: Huang, M.-D.A., Adleman, L.M. (eds.) ANTS 1994. LNCS, vol. 877, pp. 28–40. Springer, Heidelberg (1994)
2. Cantor, D.G.: Computing in the Jacobian of hyperelliptic curve. Math. Comp. 48, 95–101 (1987)
3. Diem, C.: An Index Calculus Algorithm for Plane Curves of Small Degree. In: Hess, F., Pauli, S., Pohst, M. (eds.) ANTS 2006. LNCS, vol. 4076, pp. 543–557. Springer, Heidelberg (2006)
4. Diem, C.: On the discrete logarithm problem in class groups (2009) (preprint), http://www.math.uni-leipzig.de/~diem/preprints/small-genus.pdf
5. Enge, A., Gaudry, P.: A general framework for subexponential discrete logarithm algorithms. Acta Arith. 102(1), 83–103 (2002)
6. Gaudry, P.: An algorithm for solving the discrete log problem on hyperelliptic curves. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 19–34. Springer, Heidelberg (2000)
7. Gaudry, P., Thomé, E., Thériault, N., Diem, C.: A double large prime variation for small genus hyperelliptic decomposed attack. Math. Comp. 76, 475–492 (2007) Preprint Version, http://eprint.iacr.org/2004/153/
8. Gaudry, P.: Index calculus for abelian varieties of small dimension and the elliptic curve discrete logarithm problem. Journal of Symbolic Computation 44(12), 1690–1702 (2009), Preprint version http://eprint.iacr.org/2004/073
9. Hess, F.: Computing Riemann-Roch spaces in algebraic function fields and related topics. J. Symb. Comp. 11, 1–22 (2001)
10. Hindry, M., Silverman, J.H.: Diophantine Geometry An introduction. In: Graduate Texts in Math., vol. 201. Springer, Heidelberg (2000)

11. Granger, R., Vercauteren, F.: On the Discrete Logarithm Problem on Algebraic Tori. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 66–85. Springer, Heidelberg (2005)
12. LaMacchia, B.A., Odlyzko, A.M.: Solving large sparse linear systems over finite fields. In: Menezes, A., Vanstone, S.A. (eds.) CRYPTO 1990. LNCS, vol. 537, pp. 109–133. Springer, Heidelberg (1991)
13. Nagao, K.: Index calculus for Jacobian of hyperelliptic curve of small genus using two large primes. Japan Journal of Industrial and Applied Mathematics 24(3) (2007); Preprint version entitled by Improvement of Thériault Algorithm of decomposed attack for Jacobian of Hyperelliptic Curves of Small Genus, http://eprint.iacr.org/2004/161
14. Semaev, I.: Summation polynomials and the discrete logarithm problem on elliptic curves (2004) (preprint)
15. Thériault, N.: Index calculus for hyperelliptic curves of small genus. In: Laih, C.-S. (ed.) ASIACRYPT 2003. LNCS, vol. 2894, pp. 75–92. Springer, Heidelberg (2003)
16. Wiedemann, D.H.: Solving sparse linear equations over finite fields. IEEE Trans. Inform. Theory IT-32(1), 54–62 (1986)

6 Appendix

In the appendix, we estimate the complexity of the decomposition attack, as a function of q, for fixed g, n (i.e., g, n are considered as constants) under the Assumption 1 and Assumption 2. Here, we apply the ideas of the "Rebalancing method" [5],"One large prime method" [15], and "Two large prime method" [13] [7], which are the techniques of solving discrete logarithm of the Jacobian of a hyperelliptic curve over a general finite field, to our cost estimation for the case of an extension field. Note that as g and n are fixed, the input length is linear in $\log q$. These techniques are very complicated, and we only give the outline of the algorithm and estimation of the complexity.

In this estimation, since n, g are fixed, the cost for solving the system of the equations is considered as $\mathrm{Poly}(\log q)$. For simplicity, the terms of $\mathrm{Poly}(\log q)$-part of the complexity is omitted. For this purpose, we denote the symbol $\tilde{O}$ where the complexity $\tilde{O}(N(q))$ is estimated by

$$\tilde{O}(N(q)) < x(\log q)^y N(q) \quad \text{for some constants } x, y \in \mathbb{R}_{>0},$$

and the symbol $\approx$ that the relation $N_1(q) \approx N_2(q)$ is defined by

$$\frac{N_2(q)}{x_1(\log q)^{y_1}} < N_1(q) < x_2(\log q)^{y_2} N_2(q) \text{ for some constants } x_1, x_2, y_1, y_2 \in \mathbb{R}_{>0},$$

where $N(q), N_1(q)$ and $N_2(q)$ are functions of input size q.

Now, let G be a general finite abelian group whose group law is written additively and we consider the general decomposition attack over G. In the following, we also assume that
i) The group order is known, and
ii) G has a prime order.
The assumption ii) is not an essential assumption, but make here for simplicity. Let us now fix a set B_0 subset of G.

Definition 3. *Let N be a fixed positive integer(fixed constant).*
1) An element of $g \in G$ written by $g = g_1 + .. + g_N$ for $g_1, ...g_N \in B_0$ is called potentially decomposed.
2) $g_1, ..., g_N$ are then called decomposed factors and the multiset $\{g_1, ..., g_N\}$ is called decomposed divisor.

Further, we also assume the following iii), iv), v), and vi):
iii) The probability that $g \in G$ is potentially decomposed is $O(1)$.
iv) For a $g \in G$, the cost for checking whether g is potentially decomposed or not is $\tilde{O}(1)$.
v) For the potentially decomposed $g \in G$ the cost of computing decomposed divisor $\{g_1, ..., g_N\}$ from g is $\tilde{O}(1)$. (If there are several decomposed divisors, the computation of all decomposed divisors is needed.)
vi) $|B_0|^2 \ll |G|$.

Note that $o(|G|) < |B_0|^N$ from ii) and $|B_0|^N < \tilde{O}(|G|)$ from iv). (Otherwise, the expected number of decomposed divisors is bigger than $\tilde{O}(q^{\varepsilon})$ for some $\varepsilon > 0$ and iv) does not hold.) In the normal index calculus, the number of B_0 which are used for the decomposition is basically large (i.e.,$N \gg 1$). So, the randomly chosen element is basically written by some linear sum of B_0 in many ways. However, it is difficult to compute such linear sums, so, by the use of the lifting to integer or number field ring and by the use of the sieving method, one can find some decomposition of randomly chosen element. So, remark carefully that the prerequisite condition of the normal index calculus for number field sieve and that of the decomposition attack for the Jacobian of algebraic curve is quite different.

In our case (i.e., G being the Jacobian of a hyperelliptic curve of genus g over extension field $\mathbb{F}_{q^n}$, B_0 being the set of $\mathbb{F}_{q^n}$-rational point of the curve whose x-coordinate lie in $\mathbb{F}_q$, $N = ng$), iii) is from Assumption 1 and Assumption 2, iv) and v) are from Theorem 1, and vi) is from the notations.

Let us now fix a set B subset of B_0. The set B is called the factor base and an element in $B_0 \backslash B$ is called a large prime.

Definition 4. *1) An element of $g \in G$ written by $g = g_1 + .. + g_N$ for $g_1, ...g_N \in B$ is called decomposed.*
2) An element of $g \in G$ written by $g = g_1 + .. + g_N$ for one $g_i \in B_0 \backslash B$, and the other $g_j \in B$ $(1 \leq j \leq N, j \neq i)$ is called almost decomposed.
3) An element of $g \in G$ written by $g = g_1 + .. + g_N$ for two $g_{i1}, g_{i2} \in B_0 \backslash B$, and the other $g_j \in B$ $(1 \leq j \leq N, j \neq i_1, i_2)$is called 2-almost decomposed.
4)In every case, $g_1, ..., g_N$ are also called decomposed factors and the multiset $\{g_1, ..., g_N\}$ is called decomposed divisor.

Now, we give the outlines of the algorithms named 'rebalancing method', 'one large prime method' and 'two large prime method', which are the variants of the decomposition attack [5], [15], [13], and [7], by Algorithm 1 and Algorithm 2.

Note that Algorithm 1 and Algorithm 2 are probabilistic, since they need random numbers. Also note that the probability that $r_1 a + r_2 b$ is potentially

Algorithm 1. The outline of the Rebalancing method

Input: $a, b \in G$ s.t. $a = nb$ for some unknown $n \in \mathbb{Z}/|G|\mathbb{Z}$.
Output: find n.
1: Initializing the list of the relations $L = \{\}$
2: **while** $|L| <$ suitable number N_0 **do**
3: For a pair of random numbers (r_1, r_2), computing $r_1 a + r_2 b$.
4: **if** $r_1 a + r_2 b$ being decomposed **then**
5: adding the informations of (r_1, r_2) and the decomposed factor to L.
6: (If there are several decomposed factors, choosing one decomposed factor randomly.)
7: Solving the linear algebraic computation of roughly $|B| \times |B|$ size, modulo $|G|$
8: Computing n

Algorithm 2. The outlines of the One (resp. Two)large prime method

Input: $a, b \in G$ s.t. $a = nb$ for some unknown $n \in \mathbb{Z}/|G|\mathbb{Z}$.
Output: find n.
1: Initializing the list of the relations $L = \{\}$
2: **while** $|L| <$ suitable numberN_1 (resp. N_2) **do**
3: For a pair of random numbers (r_1, r_2), computing $r_1 a + r_2 b$.
4: **if** $r_1 a + r_2 b$ being almost-decomposed (resp. 2-almost decomposed) **then**
5: adding the informations of (r_1, r_2) and the decomposed factor to L.
6: (If there are several decomposed factors, choosing one decomposed factor randomly.)
7: Updating L by the elimination of the terms of external elements.
8: Solving the linear algebraic computation of roughly $|B| \times |B|$ size, modulo $|G|$
9: Computing n

decomposed is $O(1)$, since $|G|$ is a prime number and $r_1 a + r_2 b$ can be considered as a random element of G. In Algorithm 1 and Algorithm 2, N_0 (resp. N_1, resp N_2) be the number of decomposed (resp. almost decomposed , resp. 2-almost decomposed) elements of G which are required in the rebalancing method (resp. one large prime method, resp. two large prime method). From the ideas of [5], [15], [13], and [7], the estimations of the following conjecture is expected.

Conjecture . *1)* N_0 *is estimated by* $Const \times |B|$*, i.e.,* $N_0 = O(|B|)$.
2) $N_1^2/|B_0|$ *is estimated by* $Const \times |B|$*, i.e.,* $N_1 = O(|B|^{1/2}|B_0|^{1/2})$.
3) N_2 *is estimated by* $Const \times |B_0|$*, i.e.,* $N_2 = O(|B_0|)$.

Further, we have the following estimations of the complexity.

Lemma 10. *Under the assumptions of i),ii),iii), iv) v), vi), and Conjecture, we have the following:*
1) The complexity of the general decomposition attack taking B *as a set of factor basis by the rebalancing method is minimized at* $|B| \approx |B_0|^{N/(N-1)}$*, and it is estimated by* $\tilde{O}(|B_0|^{(2N)/(N+1)})$.
2) The complexity of the general decomposition attack taking B *as a set of factor basis and taking* $B_0 \backslash B$ *as a set of large primes by the one large prime method is minimized at* $|B| \approx |B_0|^{(2N-1)/(2N+1)}$*, and it is estimated by* $\tilde{O}(|B_0|^{(4N-2)/(2N+1)})$.

3) The complexity of the general decomposition attack taking B as a set of factor basis and taking $B_0 \backslash B$ as a set of large primes by the two large prime method is minimized at $|B| \approx |B_0|^{(N-1)/N}$, and it is estimated by $\tilde{O}(|B_0|^{(2N-2)/N})$.

Proof. (Sketch of the proof) In every case, the cost of the part of linear algebra is $\tilde{O}(|B|^2)$, and for the rebalance, which is needed for minimizing the complexity, it is the same as the cost of the collecting divisors. So, we only need to estimate the optimized size $|B|$.
1)In the case of rebalancing method: The probability that the randomly chosen $g \in G$ is a decomposed is $O(|B/B_0|^N)$. So, the cost to obtain one decomposed g is $\tilde{O}(|B_0/B|^N)$. From Conjecture , we must have $O(|B|)$ number of such g. So

$$|B_0/B|^N \cdot |B| \approx |B|^2$$

where the left hand side is the cost for collecting enough decomposed group elements, and the right hand side is the cost for the linear algebra. Thus we have $|B| \approx |B_0|^{N/(N+1)}$.
2) In the case of one large prime method: The probability that the randomly chosen $g \in G$ is an almost decomposed is $O(|B/B_0|^{N-1})$. From Conjecture , we must have $O(|B|^{1/2}|B_0|^{1/2})$ number of such g. Similarly, we have

$$|B_0/B|^{N-1} \cdot |B|^{1/2}|B_0|^{1/2} \approx |B|^2$$

and $|B| \approx |B_0|^{(2N-1)/(2N+1)}$ is obtained.
3) In the case of two large prime method: The probability that the randomly chosen $g \in G$ is a 2-almost is $O(|B/B_0|^{N-2})$. From Conjecture , we must have $O(|B_0|)$ number of such g. Similarly, we have

$$|B_0/B|^{N-2} \cdot |B_0| \approx |B|^2$$

and $|B| \approx |B_0|^{(N-1)/N}$ is obtained.

Now, we apply this lemma for the decomposition attack for the Jacobian of a curve over an extension field. Note that $B_0 = \{P - \infty \,|x(P) \in \mathbb{F}_q\}$, $|B_0| \approx q$, $N = ng$ and thus, we have the following claim, which is based on the assumptions i),ii),iii),iv),v),vi),and Conjecture.

Claim . *1) The complexity of the decomposition attack with the rebalancing method is estimated by $\tilde{O}(q^{(2ng)/(ng+1)})$.*
2) The complexity of the decomposition attack with the one large prime method is estimated by $\tilde{O}(q^{(4ng-2)/(2ng+1)})$.
3) The complexity of the decomposition attack with the two large prime method is estimated by $\tilde{O}(q^{(2ng-1)/(ng)})$.

Factoring Polynomials over Local Fields II

Sebastian Pauli

Department of Mathematics and Statistics
University of North Carolina at Greensboro, Greensboro, NC 27412, USA
s_pauli@uncg.edu

Abstract. We present an algorithm for factoring polynomials over local fields, in which the Montes algorithm is combined with elements from Zassenhaus Round Four algorithm. This algorithm avoids the computation of characteristic polynomials and the resulting precision problems that occur in the Round Four algorithm.

1 Introduction

Polynomial factorization is fundamental in working with local fields. In addition to the irreducible factors of a given polynomial, computer algebra systems that support extensions of local fields (*e.g.*, Magma [1], Sage [16]) require explicit representations of the unramified and totally ramified parts of the extensions generated by arbitrary irreducible polynomials, as these systems represent such extensions as a tower of unramified and totally ramified extensions. Moreover, there are many applications of global fields that include the construction of integral bases, decomposition of ideals, and the computation of completions.

The algorithms [2,4,7,14] for factoring a polynomial $\Phi(x)$ over a local field find successively better approximations to the irreducible factors of $\Phi(x)$ until gaining sufficient precision to apply Hensel lifting. The algorithms differ in how the approximations are computed.

Algorithms based on the Zassenhaus Round Four algorithm (*e.g.* [3,4,14]) suffer from loss of precision in computing characteristic polynomials and approximating greatest common divisors. The Montes algorithm [10,11,7,8] avoids the computation of characteristic polynomials by exploiting Newton polygons of higher order. Here the most expensive operations are division with remainder and polynomial factorization over finite fields.

We present the algorithm of Montes in the terminology of [14] and use the techniques of the Round Four algorithm to derive a factorization when a breaking element is found. We also give a complexity analysis.

Notation

Let K be a field complete with respect to a non-archimedian exponential valuation ν with finite residue class field $\underline{\mathsf{K}} \cong \mathbb{F}_q$ of characteristic p; we call K a *local field*. Assume ν is normalized with $\nu(\pi) = 1$ for the uniformizing element

G. Hanrot, F. Morain, and E. Thomé (Eds.): ANTS-IX 2010, LNCS 6197, pp. 301–315, 2010.

π in the valuation ring $\mathcal{O}_{\mathsf{K}}$ of K. For $\gamma \in \mathcal{O}_{\mathsf{K}}$ denote by $\underline{\gamma}$ the class $\gamma + (\pi)$ in $\underline{\mathsf{K}}$. The unique extension of ν to an algebraic closure $\overline{\mathsf{K}}$ of K (or to any intermediate field) is also denoted ν.

In our algorithm we will be concerned with the first non-zero coefficient of the expansion of an element in a finite subextension of $\overline{\mathsf{K}}/\mathsf{K}$. We introduce an equivalence relation on the elements of $\overline{\mathsf{K}}$ which reflects this (also see [9]).

Definition 1. For $\gamma \in \overline{\mathsf{K}}^*$ and $\delta \in \overline{\mathsf{K}}^*$ we write $\gamma \sim \delta$ if

$$\nu(\gamma - \delta) > \nu(\gamma)$$

and make the supplementary assumption $0 \sim 0$. For $\varphi(x) = \sum_{i=0}^{n} \varphi_i x^i$ and $\vartheta(x) = \sum_{i=0}^{n} \vartheta_i x^i$ in $\overline{\mathsf{K}}[x]$ we write $\varphi(x) \sim \vartheta(x)$ if

$$\min{}_{0 \le i \le n}\ \nu(\varphi_i - \vartheta_i)\ >\ \min{}_{0 \le i \le n}\ \nu(\varphi_i).$$

Let L be a finite extension of K with uniformizing element π_{L}. Two elements $\gamma = \gamma_0 \pi_{\mathsf{L}}^v \in \mathsf{L}$ and $\delta = \delta_0 \pi_{\mathsf{L}}^w \in \mathsf{L}$ with $\nu(\gamma_0) = \nu(\delta_0) = 0$ are equivalent with respect to $\sim$ if and only if $v = w$ and $\gamma_0 \equiv \delta_0 \bmod (\pi_{\mathsf{L}})$. It follows immediately that the relation $\sim$ is symmetric, transitive, and reflexive.

2 Reducibility

Assume we want to factor a polynomial $\Phi \in \mathcal{O}_{\mathsf{K}}[x]$ of degree N. If $\Phi(x)$ splits into the product of two co-prime factors over the residue class field $\underline{\mathsf{K}}$ of K, say $\underline{\Phi}(x) = \underline{\Phi}_1(x) \cdot \underline{\Phi}_2(x)$, then Hensel lifting yields a factorization of $\Phi(x)$ to any given precision. In addition to this classic situation we give two further situations that we can exploit to obtain a factorization of $\Phi(x)$.

We consider a polynomial $\vartheta(x) \in \mathcal{O}_{\mathsf{K}}[x]$ as a representative of an element in the algebra $\mathsf{K}[x]/(\Phi(x))$ and determine a polynomial $\chi_\vartheta(x) \in \mathsf{K}[x]$ from $\vartheta(x)$ such that $\chi_\vartheta(\vartheta(\xi)) = 0$ for all roots ξ of $\Phi(x)$.

Definition 2. Let $\Phi(x) = \prod_{j=1}^{N}(x - \xi_j) \in \mathcal{O}_{\mathsf{K}}[x]$, where $\xi_j \in \overline{\mathsf{K}}$ for $1 \le j \le N$ and $\vartheta(x) \in \mathsf{K}[x]$. Then we set

$$\chi_\vartheta(y) := \prod_{i=1}^{N}(y - \vartheta(\xi_i)) = \operatorname{res}_x(\Phi(y), y - \vartheta(x)).$$

Assume we find $\vartheta \in \mathsf{K}[x]$ such that $\chi_\vartheta(y) = \chi_1(y)\chi_2(y)$ with $\gcd(\chi_1, \chi_2) = 1$. Reordering the roots ξ_i $(1 \le i \le N)$ of $\Phi(x)$ if necessary, we may write

$$\chi_1(y) = (y - \vartheta(\xi_1)) \cdots (y - \vartheta(\xi_r)) \text{ and } \chi_2(y) = (y - \vartheta(\xi_{r+1})) \cdots (y - \vartheta(\xi_N)),$$

where $1 \le r < N$ and obtain a proper factorization of $\Phi(x)$:

$$\Phi(x) = \gcd(\Phi(x), \chi_1(\vartheta(x))) \cdot \gcd(\Phi(x), \chi_2(\vartheta(x))). \tag{1}$$

Definition 3. We say a polynomial $\vartheta(x) \in \mathsf{K}[x]$ with $\chi_\vartheta(t) \in \mathcal{O}_\mathsf{K}[t]$ passes the *Hensel test* if $\underline{\chi}_\vartheta(t) = \underline{\rho}(t)^g$ for some irreducible polynomial $\underline{\rho}(t) \in \underline{\mathsf{K}}[t]$.

If $\vartheta(x) \in \mathsf{K}[x]$ fails the Hensel test, that is, $\chi_\vartheta(y)$ splits into two co-prime factors over $\underline{\mathsf{K}}$, say $\underline{\chi}_\vartheta(y) = \underline{\chi}_1(y)\underline{\chi}_2(y)$, then Hensel lifting yields a factorization $\chi_\vartheta(y) = \chi_1(y)\chi_2(y)$ and equation (1) gives a proper factorization of $\Phi(x)$.

Definition 4. For $\vartheta \in \mathsf{K}[x]$ we set $v^*_\Phi(\vartheta) := \min_{\Phi(\xi)=0} \nu(\vartheta(\xi))$ and say the polynomial $\vartheta(x)$ passes the *Newton test* if $\nu(\vartheta(\xi)) = \nu(\vartheta(\xi'))$ for all roots ξ and ξ' of $\Phi(x)$.

If $\varphi(x) \in \mathsf{K}[x]$ fails the Newton test, the Newton polygon of $\chi_\varphi(y)$ consists of at least two segments. Let $h/e = v^*_\Phi(\varphi)$ be the minimum of the valuations $\nu(\varphi(\xi_i))$ $(1 \le i \le N)$ in lowest terms. Then $-h/e$ is the gentlest slope of the segments of the Newton polygon of $\chi_\varphi(y)$. We set $\vartheta(x) := \varphi(x)^e/\pi^h$ and obtain $\nu(\vartheta(\xi)) = 0$ for all roots ξ of $\Phi(x)$ with $\nu(\varphi(\xi)) = h/e$ and $\nu(\vartheta(\xi)) > 0$ for all roots ξ of $\Phi(x)$ with $\nu(\varphi(\xi)) > h/e$. Thus $\underline{\chi}_\vartheta(t)$ splits into two co-prime factors and the considerations above yield a proper factorization of $\Phi(x)$.

3 Irreducibility and the Sequence $\big(\varphi_t(x)\big)_t$

In the polynomial factorization algorithm we construct a sequence of polynomials $\varphi_t(x) \in \mathcal{O}_\mathsf{K}[x]$ such that $\nu(\varphi_{t+1}(\xi)) > \nu(\varphi_t(\xi))$ for all roots ξ of $\Phi(x)$ until we either find a polynomial that fails the Newton test, which leads to a factorization of $\Phi(x)$ or we have established the irreducibility of $\Phi(x)$. If we assure that the degrees of the polynomials $\varphi_t(x)$ are less than or equal to the degree of all irreducible factors of $\Phi(x)$, we either obtain a factorization of $\Phi(x)$ or we establish the irreducibility of $\Phi(x)$ in finitely many steps [14]:

Theorem 5. *Let $\xi_1, \ldots, \xi_N$ be elements of an algebraic closure of a local field K and assume the following hypotheses hold.*

- *$\Phi(x) = \prod_{j=1}^N (x - \xi_j)$ is a square-free polynomial in $\mathcal{O}_\mathsf{K}[x]$.*
- *$\varphi(x) \in \mathsf{K}[x]$.*
- *$N\nu(\varphi(\xi_j)) > 2\nu(\operatorname{disc}\Phi)$ for $1 \le j \le N$.*
- *The degree of any irreducible factor of $\Phi(x)$ is greater than or equal to $\deg\varphi$.*

Then $N = \deg\varphi$ and $\Phi(x)$ is irreducible over K.

While we construct the sequence of polynomials $\varphi_t(x)$ we gather information about the extensions generated by the irreducible factors of $\Phi(x)$. In particular we will at all times know divisors E_t and F_t of the ramification index and inertia degree of these extensions respectively. If we find that not all of these extensions have the same inertia degree and ramification index, we will have encountered a polynomial that fails the Hensel or the Newton test. On the other hand if $E_t \cdot F_t = \deg\Phi$ we know that $\Phi(x)$ is irreducible.

Definition 6. Let $\Phi(x) \in \mathcal{O}_{\mathsf{K}}[x]$ be irreducible and let ξ be a root of $\Phi(x)$. We call a pair of polynomials $\Pi(x) \in \mathsf{K}[x]$ and $\Gamma(x) \in \mathsf{K}[x]$ with $\nu(\Pi(\xi)) = 1/E$ and $F = \left[\underline{\mathsf{K}}\left(\underline{\Gamma(\xi)}\right) : \underline{\mathsf{K}}\right]$ such that $E \cdot F = \deg \Phi$ a *two element certificate* for the irreducibility of $\Phi(x)$.

Remark 7. If a two element certificate exists then $\Phi(x)$ is irreducible and an integral basis of the extension of $\mathsf{K}(\xi)/\mathsf{K}$ generated by a root ξ of $\Phi(x)$ is given by the elements $\Gamma(\xi)^i \Pi(\xi)^j$ with $0 \le i \le F-1$ and $0 \le j \le E-1$.

In the polynomial factorization algorithm we construct a sequence of polynomials $(\varphi_t(x))_{t \in \mathbb{N}}$ where $\varphi_t \in \mathcal{O}_{\mathsf{K}}[x]$ such that

1. $\nu(\varphi_{t+1}(\xi)) > \nu(\varphi_t(\xi))$ for all roots ξ of $\Phi(x)$,
2. $\nu(\varphi_t(\xi)) = \nu(\varphi_t(\xi'))$ for all roots ξ and ξ' of $\Phi(x)$, and
3. the degree of $\varphi_t(x)$ is less than or equal to the degree of any irreducible factor of $\Phi(x)$.

In the following we assume that all polynomials that occur in our constructions pass the Hensel and Newton tests, as we can otherwise derive a factorization of $\Phi(x)$. For convenience of notation we define:

Definition 8. If $v^*_\Phi(\varphi - \vartheta) > v^*_\Phi(\varphi)$ for polynomials $\varphi(x) \in \overline{\mathsf{K}}[x]$ and $\vartheta(x) \in \overline{\mathsf{K}}[x]$ we write $\varphi \underset{\Phi}{\sim} \vartheta$. For polynomials $\chi(y) = \sum_{i=0}^{n} a_i(x) y^i \in \mathsf{K}[x][y]$ and $\tau(y) = \sum_{i=0}^{n} b_i(x) y^i \in \mathsf{K}[x][y]$ we write $\chi(y) \underset{\Phi}{\sim} \tau(y)$ if

$$\min{}_{0 \le i \le n}\ v^*_\Phi(a_i - b_i)\ >\ \min{}_{0 \le i \le n}\ v^*_\Phi(a_i).$$

4 The First Iteration

Let $\Phi(x) = \sum_{i=0}^{N} c_i x^i$ and $\varphi_1(x) := x \in \mathcal{O}_{\mathsf{K}}[x]$. Assume the Newton polygon of $\Phi(x)$ consists of one segment and let $-h_1/E_1$ be its slope in lowest terms. Then $\nu(\varphi_1(\xi)) = \nu(\xi) = h_1/E_1$ for all roots ξ of $\Phi(x)$. This implies that the ramification index of all extension generated by irreducible factors of $\Phi(x)$ is divisible by E_1. Let $\beta \in \overline{\mathsf{K}}$ with $\beta^{E_1} = \pi^{h_1}$ where π is the uniformizing element of K. We flatten the Newton polygon of $\Phi(x)$ so that it lies on the x-axis:

$$\Phi^\flat(y) := \frac{\Phi(\beta y)}{\beta^N} = \sum_{i=0}^{N} c_i \beta^{i-N} y^i.$$

Because we can only have $\nu(c_i \beta^{i-N}) = 0$ when $E_1 \mid i$, we have

$$\Phi^\flat(y) \sim \sum_{j=0}^{N/E_1} c_{j \cdot E_1} \pi^{h_1(j - N/E_1)} y^{j \cdot E_1}.$$

Replacing y^{E_1} by z yields

$$A_1(z) := \sum_{j=0}^{N/E_1} c_{j \cdot E_1} \pi^{h_1(j - N/E_1)} z^j.$$

The polynomial $\underline{A}_1(z) \in \underline{\mathsf{K}}[z]$ is called the *associated polynomial* [11,10] or *residual polynomial* [7,8] of $\Phi(x)$ with respect to $\varphi_1(x)$. Assume that $\underline{A}_1(z) = \underline{\rho}_1(z)^r$ for some irreducible polynomial $\underline{\rho}_1 \in \underline{\mathsf{K}}$. Otherwise $\varphi_1(x)^{E_1}/\pi^{h_1} = x^{E_1}/\pi^{h_1}$ would fail the Hensel test and (1) would yield a factorization of $\Phi(x)$. All fields $\mathsf{K}(\xi)$, where ξ is a root of $\Phi(x)$, contain an element ξ^{E_1}/π^{h_1}, whose minimal polynomial is a power of $\underline{\rho}_1(z)$ over $\underline{\mathsf{K}}[z]$; therefore their ramification indices are divisible by $F_1 := \deg \underline{\rho}_1$. Let $\gamma_1 \in \overline{\mathsf{K}}$ be a root of a lift $\rho_1(z) \in \mathcal{O}_{\mathsf{K}}[z]$ of $\underline{\rho}_1(z)$. In the unramified extension $\mathsf{K}_1 := \mathsf{K}(\gamma_1)$ we have the relation $x^{E_1} \underset{\Phi}{\sim} \pi^{h_1} \cdot \gamma_1$. Since $\nu\big(\rho_1(\varphi_1(\xi)^{E_1}/\pi^{h_1})\big) > 0$ for all roots ξ of $\Phi(x)$, we get

$$\nu\left(\pi^{h_1F_1}\rho_1\left(\frac{\varphi_1(\xi)^{E_1}}{\pi^{h_1}}\right)\right) > \nu(\pi^{h_1}) = \nu\big(\varphi_1^{E_1}(\xi)\big) > \nu\big(\varphi_1(\xi)\big) = \nu(\xi).$$

We set $\varphi_2(x) := \pi^{h_1F_1}\rho_1(\varphi_1(x)^{E_1}/\pi^{h_1})$ and continue the construction of our sequence of polynomials $(\varphi_t)_t$. Obviously $\deg \varphi_2 = E_1F_1$, which divides the degree of every irreducible factor of $\Phi(x)$.

Remark 9. Because the Newton polygon of $\varphi_2(x)$ consists of one segment of slope $-h_1/E_1$ with $\gcd(h_1, E_1) = 1$ and its associated polynomial with respect to x is $\underline{\rho}_1(z)$ of degree F_1, the extensions $\mathsf{K}(\alpha)$, where α is a root of $\varphi_2(x)$, have inertia degree F_1 and ramification index E_1. Hence $\varphi_2(x)$ with $\deg \varphi_2 = E_1F_1$ is irreducible.

5 The Second Iteration

Definition 10. Let $\Phi(x) \in \mathcal{O}_{\mathsf{K}}[x]$ of degree N and $\varphi(x) \in \mathcal{O}_{\mathsf{K}}[x]$ of degree n be monic polynomials and assume $n \mid N$. We call

$$\Phi(x) = \sum_{i=0}^{N/n} a_i(x)\varphi^i(x)$$

with $\deg(a_i) < \deg(\varphi)$ the φ-expansion of $\Phi(x)$.

We use the φ_2-expansion of $\Phi(x)$ to find the valuations $\nu(\varphi_2(\xi))$. Set $n_2 := \deg \varphi_2$ and let $\Phi(x) = \sum_{i=0}^{N/n_2} a_i(x)\varphi_2^i(x)$ be the φ_2-expansion of $\Phi(x)$. For each root ξ of $\Phi(x)$ we have

$$0 = \Phi(\xi) = \sum_{i=0}^{N/n_2} a_i(\xi)\varphi_2^i(\xi).$$

Hence

$$\chi_{2,\xi}(y) = \sum_{i=0}^{m} a_i(\xi)y^i \in \mathcal{O}_{\mathsf{K}(\xi)}[y]$$

with $m = N/n_2 = \deg(\Phi)/\deg(\varphi_2)$ is a polynomial with root $\varphi_2(\xi)$. Assume that $a_i(x) = \sum_{j=0}^{n_2-1} a_{i,j}x^j$. As the valuations

$$v_\Phi^*(\varphi_1) = \frac{h_1}{E_1}, \ldots, v_\Phi^*(\varphi_1^{E_1-1}) = \frac{(E_1-1)h_1}{E_1}$$

are distinct (and not in $\mathbb{Z}$) and

$$1, \frac{\varphi_1(x)^{E_1}}{\pi^{h_1}} \underset{\Phi}{\sim} \gamma_1, \ldots, \left(\frac{\varphi_1(x)^{E_1}}{\pi^{h_1}}\right)^{F_1-1} \underset{\Phi}{\sim} \gamma_1^{F_1-1}$$

are linearly independent over K, we have

$$v_\Phi^*(a_i) = \min_{0 \le j \le n_2-1} \nu(a_{i,j})(h_1/E_1)j.$$

If the Newton polygon of $\chi_{2,\xi}(y)$ consists of more than one segment then $\varphi_2(x)$ fails the Newton test and we can derive a factorization of $\Phi(x)$. Otherwise let $-h_2/e_2$ be the slope of the Newton polygon of $\chi_{2,\xi}(y)$ in lowest terms. Then $\nu(\varphi_2(\xi)) = h_2/e_2$ for all roots ξ of $\Phi(x)$. We set $E_2^+ := e_2/\gcd(E_1, e_2)$. For all roots ξ of $\Phi(x)$ the ramification index of $\mathsf{K}(\xi)$ is divisible by $E_2 := E_1 \cdot E_2^+$. Because the denominator of $E_2^+ h_2/e_2$ is a divisor of E_1 there is

$$\psi_2(x) := \pi^{s_\pi} \varphi_1(x)^{s_1} = \pi^{s_\pi} x^{s_1} \in \mathsf{K}[x]$$

with $s_1 \in \{0, \ldots, E_1 - 1\}$ and $s_\pi \in \mathbb{Z}$ such that $v_\Phi^*(\psi_2) = E_2^+ h_2/e_2$.

We flatten the Newton polygon of $\chi_{2,\xi}(y)$. Let $\beta \in \overline{\mathsf{K}}$ with $\beta^{E_2^+} = \psi_2(x)$ and consider the polynomial $\chi_{2,\xi}^\flat(y) := \chi_{2,\xi}(\beta y)/\beta^m$. As only the valuations of the coefficients of $y^{i \cdot E_2^+}$ $(0 \le i \le m/E_2^+)$ can be zero we get

$$\begin{aligned}\underline{\chi}_{2,\xi}^\flat(y) &= \sum_{i=0}^{m/E_2^+} \underline{a_{i\cdot E_2^+}(\xi)\beta^{i\cdot E_2^+ - m}} y^{i\cdot E_2^+} \\ &= \sum_{i=0}^{m/E_2^+} \underline{a_{i\cdot E_2^+}(\xi)\psi_2(\xi)^{i-m/E_2^+}} y^{i\cdot E_2^+} \in \underline{\mathsf{K}}_2[y].\end{aligned}$$

Using the relation $x^{E_1} \underset{\Phi}{\sim} \pi^{h_1} \cdot \gamma_1$, which is independent of ξ, we find coefficients $\widehat{a}_i \in \mathsf{K}_1$ with $\widehat{a}_i \underset{\Phi}{\sim} a_{i\cdot E_2^+}(x)\psi_2^{i-m/E_2^+}(x)$. We set

$$A_2(z) := \sum_{i=0}^{m/E_2^+} \widehat{a}_i z^i \underset{\Phi}{\sim} \sum_{i=0}^{m/E_2^+} a_{i\cdot E_2^+}(x)\psi_2^{i-m/E_2^+}(x) z^i$$

and obtain the *associated polynomial* $\underline{A}_2(z) \in \underline{\mathsf{K}}_1[z]$ of $\Phi(x)$ with respect to $\varphi_2(x)$.

If $\underline{A}_2(y)$ splits into two or more co-prime factors over $\underline{\mathsf{K}}_1 = \underline{\mathsf{K}}(\gamma_1)$, we can derive a factorization of $\Phi(x)$: Since $\deg \psi_2(x)$ is less than the degree of any irreducible factor of $\Phi(x)$ we have $\gcd(\psi_2(x), \Phi(x)) = 1$ and the extended Euclidean algorithm yields $\psi_2^{-1}(x) \in \mathcal{O}_{\mathsf{K}_1}[x]$ such that $\psi_2(x) \cdot \psi_2^{-1}(x) \equiv 1 \mod \Phi(x)$. The polynomial $\varphi_2^{E_2^+}(x) \cdot \psi_2^{-1}(x)$ fails the Hensel test.

Otherwise $\underline{A}_2(z) = \underline{\rho}_2(z)^{r_2}$ for some irreducible polynomial $\underline{\rho}_2(z) \in \underline{\mathsf{K}}_1[z]$. We set $\mathsf{K}_2 := \mathsf{K}(\gamma_2)$ where γ_2 is a root of a lift $\rho_2(z) \in \mathcal{O}_{\mathsf{K}_1}[z]$ of $\underline{\rho}_2(z) \in \underline{\mathsf{K}}_1[z]$, let $F_2^+ := \deg \rho_2$, and obtain $\varphi_2(x)^{E_2^+} \underset{\Phi}{\sim} \gamma_2 \psi_2(x)$.

Next we construct $\varphi_3(x) \in \mathcal{O}_{\mathsf{K}}[x]$ with $v^*_{\Phi}(\varphi_3) > v^*_{\Phi}(\varphi_2)$ and $\deg \varphi_3 = E_2F_2$. The coefficients of $\rho_2(z) \in \mathcal{O}_{\mathsf{K}_1}$ can be written as polynomials in $\gamma_1 \underset{\Phi}{\sim} x^{E_1}/\pi^{h_1}$, say

$$\rho_2(z) = \sum_{i=0}^{F_2^+} \sum_{j=0}^{F_1-1} r_{i,j} \gamma_1^j z^i$$

where $r_{i,j} \in \mathcal{O}_{\mathsf{K}}$. We are looking for

$$\varphi_3(x) \underset{\Phi}{\sim} \psi_2(x)^{F_2^+} \rho_2\left(\frac{\varphi_2(x)^{E_2^+}}{\psi_2(x)}\right) = \sum_{i=0}^{F_2^+} \sum_{j=0}^{F_1-1} r_{i,j} \left(\frac{x^{E_1}}{\pi^{h_1}}\right)^j \psi_2(x)^{F_2^+-i} \varphi_2(x)^{iE_2^+}$$

with $\deg \varphi_3 = E_2F_2 = E_2^+F_2^+E_1F_1$. We have $v^*_{\Phi}\big(\rho_1(x^{E_1}/\pi^{h_1})\big) > 0$. If we write $\rho_1(z) = z^{F_1} + \rho_1^*(z)$ with $\deg(\rho_1^*) < F_1$ this implies

$$\varphi_1^{E_1F_1} \underset{\Phi}{\sim} -(\pi^{h_1})^{F_1} \rho_1^*\left(\frac{x^{E_1}}{\pi^{h_1}}\right).$$

It follows that we can find a polynomial $R_{i,j}(x)$ with $\deg R_{i,j} < E_1F_1$ such that

$$R_{i,j}(x) \underset{\Phi}{\sim} r_{i,j} \left(\frac{x^{E_1}}{\pi^{h_1}}\right)^j \psi_2(x)^{F_2^+-i} = r_{i,j} \left(\frac{x^{E_1}}{\pi^{h_1}}\right)^j (\pi^{s_\pi} x^{s_1})^{F_2^+-i}.$$

Thus the polynomial

$$\varphi_3(x) = \varphi_2(x)^{E_2^+F_2^+} + \sum_{i=0}^{F_2^+-1} \sum_{j=0}^{F_1-1} R_{i,j}(x) \varphi_2(x)^{iE_2^+}$$

has the desired properties $v^*_{\Phi}(\varphi_3) > v^*_{\Phi}(\varphi_2)$ and $\deg \varphi_3 = E_2F_2$.

Remark 11. $\varphi_3(x) \in \mathcal{O}_{\mathsf{K}}[x]$ is irreducible.

6 Data and Relations

In the algorithm we continue the construction of the sequence of polynomials $(\varphi_t)_t$ from the previous two sections. In the following steps the computation of $\psi_t(x)$, the valuation of the coefficients $a_i(x)$ of the φ_t-expansion of $\Phi(x)$, the coefficients of the associated polynomial, and φ_{t+1} becomes more involved and relies on the data computed in the previous iteration. We initially set

$$\mathsf{K}_0 := \mathsf{K}, \quad \varphi_1 := x, \quad E_0 := 1, \quad F_0 := 1$$

and compute the following data in every iteration:

$\varphi_t(x) \in \mathcal{O}_\mathsf{K}[x]$	with $v^*_\Phi(\varphi_t) > v^*_\Phi(\varphi_{t-1})$ and $n_t = \deg(\varphi_t) = E_{t-1}F_{t-1}$; an approximation to an irreducible factor of $\Phi(x)$
$h_t/e_t = v^*_\Phi(\varphi_t)$	with $\gcd(h_t, e_t) = 1$
$E_t^+ = \dfrac{e_t}{\gcd(E_{t-1}, e_t)}$	the increase of the maximum known ramification index
$E_t = E_t^+ \cdot E_{t-1}$	the maximum known ramification index
$\psi_t(x) = \pi^{s_\pi} \prod_{i=1}^{t-1} \varphi_i^{s_i}$	with $s_\pi \in \mathbb{Z}$ and $0 \le s_i < E_i^+$ such that $v^*_\Phi(\psi_t) = v^*_\Phi(\varphi_t^{E_t^+})$
$\underline{A}_t(y) \in \underline{\mathsf{K}}_{t-1}[y]$	the associated polynomial of $\Phi(x)$ with respect to $\varphi_t(x)$
$\underline{\rho}_t(y) \in \underline{\mathsf{K}}_{t-1}[y]$	irreducible with $\underline{\rho}_t^{r_t}(y) = \underline{A}_t(y)$
$\gamma_t \in \mathsf{K}_t$	such that $\varphi_t^{E_t^+} \underset{\Phi}{\sim} \gamma_t \psi_t$
$\mathsf{K}_t = \mathsf{K}_{t-1}(\gamma_t)$	the maximum known unramified subfield
$F_t^+ = [\mathsf{K}_t : \mathsf{K}_{t-1}]$	the increase of the maximum known inertia degree
$F_t = F_t^+ \cdot F_{t-1}$	the maximum known inertia degree

7 The u-th Iteration

Assume we have computed the data and relations given above for t up to $u-1$ and that $\varphi_u(x)$ of degree $n_u = E_u F_u$ is the best approximation to an irreducible factor of $\Phi(x)$ found so far. We compute the φ_u-expansion $\Phi(x) = \sum_{i=0}^{N/n_u} a_i(x)\varphi_u(x)^i$ of $\Phi(x)$ and set $\chi_u(y) := \sum_{i=0}^{N/n_u} a_i(x) y^i$.

Definition 12. Let $a(x) \in \mathcal{O}_\mathsf{K}[x]$ with $\deg a < E_{t-1}F_{t-1}$. We call

$$a(x) = \sum_{j_{t-1}=0}^{E_{t-1}^+ F_{t-1}^+ - 1} \varphi_{t-1}^{j_{t-1}}(x) \cdots \sum_{j_2=0}^{E_2^+ F_2^+ - 1} \varphi_2^{j_2}(x) \sum_{j_1=0}^{E_1 F_1 - 1} x^{j_1} \cdot a_{j_1,\ldots,j_{t-1}},$$

where $a_{j_1,\ldots,j_{t-1}} \in \mathcal{O}_\mathsf{K}$ $(0 \le j_i \le E_i,\ 0 \le i \le t)$, the $(\varphi_1, \ldots, \varphi_{t-1})$-expansion of $a(x)$.

From the $(\varphi_1, \ldots, \varphi_{u-1})$-expansion of $a_i(x)$ we obtain the valuations of $a_i(\xi)$ and see that they are independent of the choice of the root ξ of $\Phi(x)$. Since, by construction, the values

$$v^*_\Phi(\varphi_1), \ldots, v^*_\Phi(\varphi_1^{E_1-1}), v^*_\Phi(\varphi_2), \ldots, v^*_\Phi(\varphi_2^{E_2^+-1}), v^*_\Phi(\varphi_3), \ldots\ldots, v^*_\Phi(\varphi_{u-1}^{E_{u-1}^+-1})$$

are distinct (and not in $\mathbb{Z}$) and for $0 \le t \le u-1$ the elements

$$1, \gamma_t \underset{\Phi}{\sim} \varphi_t(x)^{E_t^+}/\psi_t(x), \ldots, \gamma_t^{F_t^+-1} \underset{\Phi}{\sim} \left(\varphi_t(x)^{E_t^+}/\psi_t(x)\right)^{F_t^+-1}$$

are linearly independent over $\mathsf{K}_{t-1} = \mathsf{K}(\gamma_1, \ldots, \gamma_{t-1})$ we have (see [7, Lemma 4.21]):

Lemma 13. *Let $a(x) \in \mathcal{O}_{\mathsf{K}}[x]$ with $\deg a < E_{t-1}F_{t-1}$ and let $a_{j_1,\ldots,j_{t-1}}$, with $0 \le j_i < E_i^+F_i^+ - 1$, be the coefficients of the $(\varphi_1,\ldots,\varphi_{t-1})$-expansion of $a(x)$. Then*

$$v_{\Phi}^*(a) = \min_{\substack{1\le i\le t-1\\ 1\le j_i<E_i^+}} v_{\Phi}^*\big(\varphi_{t-1}^{j_{t-1}}(x)\cdots\varphi_2^{j_2}(x)\cdot x^{j_1}\cdot a_{j_1,\ldots,j_{t-1}}\big).$$

If the Newton polygon of $\chi_t(y)$ consists of one segment, say of slope $-h_u/e_u$, with $\gcd(h_u,e_u)=1$, then $\varphi_t(x)$ passes the Newton test. We set $E_u^+ := \frac{e_u}{\gcd(E_{u-1},e_u)}$ and construct

$$\psi_u(x) = \pi^{s_\pi}\prod_{t=1}^{u-1}\varphi_t(x)^{s_t}$$

with $s_\pi \in \mathbb{Z}$ and $0 \le s_t < E_t^+$ $(1 \le t < u)$ such that $v_{\Phi}^*(\psi_u) = E_u^+h_u/e_u$ using the following algorithm. For $q \in \mathbb{Q}$ we denote by $\mathrm{den}(q)$ the denominator of q in lowest terms.

Algorithm 14 (Psi)

Input: $v_{\Phi}^*(\varphi_i)$ and E_i^+ for $0 \le i \le t$, $E = E_0^+\cdots E_t^+$, $v \in \mathbb{Q}$ with $E \mid \mathrm{den}(v)$.
Output: $s_\pi \in \mathbb{Z}$, $0 \le s_i \le E_i^+$ $(1 \le i \le t)$ such that $v_{\Phi}^*(\pi^{s_\pi}\varphi_0^{s_0}\cdots\varphi_t^{s_t}) = v$.

- $d \leftarrow E$, $i \leftarrow t$
- for i from t to 1 by -1:
  - $d \leftarrow d/E_i^+$, $v' \leftarrow v\cdot d$, $e \leftarrow v_{\Phi}^*(\varphi_i)\cdot d$
  - Find s_i such that $e\cdot s_i \equiv v' \mod \mathrm{den}(d\cdot e)$
  - $v \leftarrow v - s_i v_{\Phi}^*(\varphi_i)$
- $s_\pi \leftarrow v$
- return $s_\pi, s_1, \ldots, s_t$

Next we determine the associated polynomial $\underline{A}_u(y)$ of $\Phi(x)$ with respect to $\varphi_u(x)$. Because we have representations of $a_i(x)$ $(0 \le i \le N/n_i)$ and $\psi_u(x)$ by power products of $\pi, \varphi_1, \ldots, \varphi_{u-1}$ we can use the relations $\varphi_t(x)^{E_t^+} \underset{\Phi}{\sim} \gamma_t\psi_t(x)$ to find the coefficients $\widehat{a}_i \in \mathsf{K}_{u-1}$ such that $\widehat{a}_i \underset{\Phi}{\sim} a_{i\cdot E_u^+}(x)\psi_u(x)^{i-m/E_u^+}$. We get the associated polynomial

$$\underline{A}_u(z) = \sum_{i=0}^{m/E_u^+}\underline{\widehat{a}}_i z^i$$

where $m = N/n_u$. Assume that $\underline{A}_u(z) = \underline{\rho}_u(z)^r$ for some irreducible polynomial $\underline{\rho}_u(z) \in \underline{\mathsf{K}}_{u-1}(z)$. Otherwise we can find $\vartheta(x) \in \mathsf{K}[x]$ with $\vartheta(x) \underset{\Phi}{\sim} \varphi_u(x)^{E_u^+}/\psi_u(x)$ that fails the Hensel test, which yields a factorization of $\Phi(x)$. Let $\rho_u(z) \in \mathsf{K}_{u-1}$ be a lift of $\underline{\rho}_u(z)$, and set $F_u^+ := \deg\rho_u$.

Finally we construct $\varphi_{u+1}(x) \in \mathcal{O}_{\mathsf{K}}[x]$ of degree $E_uF_u = E_u^+F_u^+E_{u-1}F_{u-1}$ such that

$$\varphi_{u+1}(x) \underset{\Phi}{\sim} \sum_{i=0}^{F_u^+}\vartheta_i(x)\varphi_u(x)^{iE_u^+} = \psi_u(x)^{F_u^+}\rho_u(\varphi_u^{E_u^+}(x)/\psi_u(x)), \tag{2}$$

where the $\vartheta_i(x)$ are sums of power products of $\pi, \varphi_1, \ldots, \varphi_{u-1}$. For $t = u-1, u-2, \ldots, 0$ we recursively apply

$$v_\Phi^*\left(\rho_t\left(\frac{\varphi_t^{E_t^+}}{\psi_t}\right)\right) > 0$$

to reduce the maximum exponent of $\varphi_t(x)$ to $E_t^+F_t^+ - 1$, such that the degree of the $\varphi_t(x)$ term is at most $\deg(\varphi_t(x)^{E_t^+F_t^+-1}) = (E_{t-1}F_{t-1})(E_t^+F_t^+ - 1)$. Thus we can find a $\varphi_{u+1}(x)$ that fulfills the degree condition $\deg\varphi_{u+1} = E_uF_u$. Furthermore

$$v_\Phi^*(\varphi_{u+1}) = v_\Phi^*\left(\psi_u^{F_u^+}\rho_u\left(\frac{\varphi_u(x)^{E_u^+}}{\psi_u(x)}\right)\right) > v_\Phi^*\left(\psi_u^{F_u^+}\right) \geq v_\Phi^*(\varphi_u).$$

As a preparation for the next iteration we set $\mathsf{K}_u := \mathsf{K}_{u-1}(\gamma_u)$ with γ_u a root of $\rho_u(z)$ and obtain the relation $\varphi^{E_u^+}(x) \underset{\Phi}{\sim} \gamma_u\psi_u(x)$.

Remark 15. $\varphi_{u+1}(x) \in \mathcal{O}_\mathsf{K}[x]$ is irreducible.

8 The Algorithm

We summarize the steps for the construction of the sequence $(\varphi_t(x))_t$ in an algorithm. Although we use the unramified extensions K_t/K above and in the algorithm, in practice the γ_i are represented as elements in the residue class field $\underline{\mathsf{K}}_t$. Furthermore, many of the manipulations in the algorithm can be conducted on the representations of $\psi_t(x)$ as power products of $\pi, \varphi_1(x), \ldots, \varphi_{t-1}(x)$ and of $a_i(x)$ as sums of power products of $\pi, \varphi_1(x), \ldots, \varphi_{t-1}(x)$ thus reducing these operations to operations of vectors of integers.

Algorithm 16 (Polynomial Factorization)
Input: a monic, separable, squarefree polynomial $\Phi(x)$ over a local field K.
Output: a proper factorization of $\Phi(x)$ if one exists,
a two-element certificate for $\Phi(x)$ otherwise.

(**1**) Initialize $t \leftarrow 1$, $\varphi_1(x) \leftarrow x$, $E_0 = 1$, $F_0 = 1$, $\mathsf{K}_0 = \mathsf{K}$.
(**2**) Repeat:
  (**a**) Find the φ_t expansion $\Phi(x) = \sum_{i=1}^{N/\deg\varphi_t} a_i(x)\varphi(x)^i$ of $\Phi(x)$.
  (**b**) Find $v_\Phi^*(a_i)$ for $0 \leq i \leq N/\deg\varphi_t$.
  (**c**) If $\varphi_t(x)$ fails the Newton test: return a proper factorization of $\Phi(x)$.
  (**d**) $h_t/e_t \leftarrow v_\Phi^*(\varphi)$ with $\gcd(h_t, e_t) = 1$; $E_t^+ \leftarrow \frac{e_t}{\gcd(e_t,E)}$; $E_t \leftarrow E_t^+ \cdot E_{t-1}$.
  (**e**) Construct $\psi_t(x) = \pi^{s_\pi}\prod_{i=1}^{t-1}\varphi_i(x)^{s_i}$ with $v_\Phi^*(\psi_t) = E_t^+v_\Phi^*(\varphi_t)$, $s_\pi \in \mathbb{N}$, $0 \leq s_i < E_i^+$ $(1 \leq i \leq t-1)$, $\deg\psi_t < E_iF_i$.
  (**f**) Compute the associate polynomial $\underline{A}_t(z)$.
  (**g**) Find a factorization of $\underline{A}_t(z) \in \mathsf{K}_t(z)$.
  (**h**) If $\underline{A}_t(z)$ has two co-prime factors: return a proper factorization of $\Phi(x)$.

(**i**) $F_t^+ \leftarrow \deg \underline{\rho}$ where $\underline{\rho}_t(z)^r = \underline{A}_t(z)$, $\underline{\rho}_t(z) \in \underline{\mathsf{K}}_{t-1}[z]$ irreducible; $F_t \leftarrow F_t^+ \cdot F_{t-1}$, $\mathsf{K}_t \leftarrow \mathsf{K}[x]/(\rho_t(x))$.
(**j**) If $E_t F_t = \deg \Phi$: return a two-element certificate for $\Phi(x)$.
(**k**) Find $\varphi_{t+1}(x) \underset{\Phi}{\sim} \rho_t\big(\varphi_t(x)^{E_t^+} \,/\, \psi_t(x)^{\deg(\rho)}\big)$ of degree $n_{t+1} = E_t F_t$ in $\mathcal{O}_\mathsf{K}[x]$.
(**l**) $t \leftarrow t+1$.

Certificates for Irreducibility

If $\Phi(x)$ is irreducible we will have $E_t F_t = N$ for some t. We obtain the two element certificate (Definition 6) for the irreducibility of $\Phi(x)$ as follows. A polynomial $\Pi(x) \in \mathsf{K}[x]$ with $v_\Phi^*(\Pi) = 1/E_t$ can be found using Algorithm 14. If $F_t = 1$ we can choose $\Gamma(x) = x$. If $F_t \neq 1$, let i be maximal with $F_i^+ \neq 0$. We find $\Gamma(x) \in \mathsf{K}[x]$ with $\Gamma(x) \underset{\Phi}{\sim} \varphi_i(x)^{E_i^+} / \psi_i(x)$.

9 Complexity

We restrict our analysis of the complexity of the algorithm to the main loop. The first complexity estimate for the Montes algorithm, restricted to irreducibility testing, was given by Veres [17] and improved by Ford and Veres [5]. The complexity estimate for determining the irreducibility of a polynomial $\Phi(x) \in \mathbb{Z}_p[x]$ of degree N using this algorithms is $O(N^{3+\varepsilon}\nu(\operatorname{disc}\Phi) + N^{2+\varepsilon}\nu(\operatorname{disc}\Phi)^{2+\varepsilon})$. The running time of the Round Four algorithm is analyzed in [14], but without taking into account the precision loss in the computation of greatest common divisors. Both estimates rely on Theorem 5 to bound the number of iterations and the required precision and only differ slightly in the exponent of the discriminant of $\Phi(x)$.

Lemma 17. *Let $\Phi(x) \in \mathcal{O}_\mathsf{K}[x]$ be of degree N and let $\varphi(x) \in \mathcal{O}_\mathsf{K}[x]$ be monic of degree n. Then the φ-expansion of $\Phi(x)$ can be computed in $O(N^2)$ operations in $\mathcal{O}_\mathsf{K}$.*

Proof. In order to determine the φ-expansion $\Phi(x) = \sum_{i=1}^{N/n} a_i(x)\varphi(x)^i$ we first compute $q_0(x), a_0(x) \in \mathcal{O}_\mathsf{K}[x]$ with $\Phi(x) = \varphi(x)q_0(x) + a_0(x)$, which can be done in $O((N-n)n)$ operations in $\mathcal{O}_\mathsf{K}[x]$. Next we determine $q_1(x), a_1(x) \in \mathcal{O}_\mathsf{K}[x]$ with $q_0(x) = \varphi(x)q_1(x) + a_1(x)$ ($O((N-2n)n)$ operations in $\mathcal{O}_\mathsf{K}[x]$), and so on. Therefore the φ-expansion of $\Phi(x)$ can be computed in

$$O((N-n)n) + O((N-2n)n) + \cdots + O((2n)n) = O\left(n\left(\frac{N^2}{n} - n\sum_{i=0}^{N/n} i\right)\right) = O(N^2)$$

operations in $\mathcal{O}_\mathsf{K}$.

The computation of the $(\varphi_1, \ldots, \varphi_{t-1})$-expansion of a polynomial $a(x) \in \mathcal{O}_\mathsf{K}[x]$ of degree $m \leq \deg \varphi_t - 1$ consists of the recursive computation of $\varphi_{t-1}, \varphi_{t-2}, \ldots,$

φ_2, and φ_1-expansions. Let $n_i = \deg \varphi_i$ $(1 \le i \le t)$. The φ_{t-1}-expansion of $a(x)$ yields up to m/n_{t-1} polynomials of degree less than n_t. The φ_{t-2}-expansions of these polynomials yield up to $m/n_{t-1} \cdot n_{t-1}/n_{t-2} = m/n_{t-2}$ of degree less than n_{t-2}. Thus the $(\varphi_1, \ldots, \varphi_{t-1})$-expansion of $a(x)$ can be computed in

$$O\left(m^2\right) + O\left(\tfrac{m}{n_t} n_t^2\right) + O\left(\tfrac{m}{n_{t-1}} n_{t-1}^2\right) + \cdots + O\left(\tfrac{m}{n_1} n_1^2\right) + O(m)$$

operations in $\mathcal{O}_\mathsf{K}$. Because $n_{i+1}/n_i \ge 2$ this is less than

$$O\left(m^2\right) + O\left(\tfrac{m^2}{2}\right) + \cdots + O\left(\tfrac{m^2}{2^{t-1}}\right) + O(m) = O\left(m^2 \textstyle\sum_{i=0}^{\lfloor \log_2 m \rfloor} 2^{-i}\right) = O(m^2).$$

Lemma 18. *The $(\varphi_0, \ldots, \varphi_{t-1})$-expansion of $a(x) \in \mathcal{O}_\mathsf{K}[x]$ with $m = \deg a \le \deg \varphi_t - 1$ can be computed in $O(m^2)$ operations in $\mathcal{O}_\mathsf{K}$.*

By Theorem 5 the polynomial $\Phi(x)$ is irreducible, if $N v^*_\Phi(\varphi_t) > 2\nu(\operatorname{disc} \Phi)$ for some $t \in \mathbb{N}$. In every iteration the increase from $v^*_\Phi(\varphi_t)$ to $v^*_\Phi(\varphi_{t+1})$ is at least $2/N$, unless $E = N$, but that would imply irreducibility. Thus the algorithm terminates after at most $\nu(\operatorname{disc} \Phi)$ iterations.

In our analysis of the cost of the steps in the main loop we exclude the cost of finding a proper factorization to a desired precision using the methods of section 2 in steps **(c)** and **(h)**. We assume that two polynomials of degree up to n can be multiplied in $O(n \log n \log\log n) = O(n^{1+\varepsilon})$ operations in their coefficient ring [15].

(a,b,c,d) By Lemma 18 the φ_t-expansion

$$\Phi(x) = \varphi_t(x)^{N/n_t} + \sum_{i=0}^{N/n_t - 1} a_i(x) \varphi_t(x)^i$$

of $\Phi(x)$ and the $(\varphi_1, \ldots, \varphi_t)$-expansion of the $a_i(x)$ can be computed in $O(N^2)$ operations in $\mathcal{O}_\mathsf{K}$.

(e) The exponents $s_\pi, s_1, \ldots, s_{t-1}$ in $\psi_t(x) = \pi^{s_\pi} \varphi_1(x)^{s_0} \cdots \varphi_{t-1}(x)^{s_{t-1}}$ with $v^*_\Phi(\psi) = h_t/e_t$ can be computed with Algorithm 14. The most expensive computation is the extended Euclidean construction, which for integers less than N runs in time $O((\log N)^2)$, at most $\log_2 N$ times.

(f) We have a representation of $a_i(x)\psi_t(x)^{i-(N/n_t)}$ $(1 \le i \le N/n_t)$ as n_t sums of power products of $\pi, \varphi_1(x), \ldots, \varphi_{t-1}(x)$. In this representation only the exponents of $\varphi_i(x)$ where $E_i^+ F_i^+ \ne 1$ are non-zero. There are at most $\log_2 N$ such indices i. Let m_t be the number of $i < t$ with $E_i^+ F_i^+ \ne 1$. Reducing the coefficients of the associated polynomial in this representation using the relations $\varphi_i(x)^{E_i^+}/\psi_i(x) \underset{\Phi}{\sim} \gamma_i$ $(1 \le i \le m_t)$ takes at most $N \sum_{i=1}^{m_t} i = O(N(\log N)^2)$ integer additions and $N(t-1) = O(N \log N)$ multiplications in the finite field $\underline{\mathsf{K}}_t$ with q^F elements.

(g,h) The factorization of a polynomial of degree at most N/F over a finite field with at most q^F elements can be done in $O((N/F)^2 \log q^F)$ bit operations [6].

(j) The cost of finding the exponents for the representation of $\Pi(x) \in \mathsf{K}[x]$ with $v^*_\Phi(\Pi) = 1/E$ as a power product of $\pi, \varphi_1(x), \ldots, \varphi_t(x)$ is the same as the cost of finding $\psi(x)$ in step **(f)**. The polynomial $\Gamma(x)$ can be computed in the same way as the coefficients $\vartheta_i(x)$ in step **(l)**.

(k) The polynomial $\varphi_{t+1}(x)$ is constructed as a polynomial in $\varphi_t(x)^{E_t^+}$ of degree F_t^+ with coefficients $\vartheta_i(x)$, $0 \le i \le F_t^+$, (see (2)), obtained from the representations of the elements γ_u as $\varphi_u(x)^{E_u}/\psi_u(x)$ and

$$v^*_\Phi\big(\rho_u(\varphi_u(x)^{E_u}/\psi_u(x))\big) > 0$$

for $1 \le u \le t-1$. This is done by manipulating the exponents in the representation of the polynomials as sums of power products of $\pi, \varphi_1(x), \ldots, \varphi_t(x)$. The computation of $\varphi_t(x)^{E_t^+}$ takes $\log_2 E_t$ multiplications of polynomials of degree up to $E_t^+ E_{t-1} F_{t-} t < N$. For $2 \le j \le F_t^+$ the polynomial $\big(\varphi_t(x)^{E_t^+}\big)^j$ can be computed in F_t^+ multiplications of polynomials of degree up to $E_t F_t < N$. For $1 \le t-2$ the exponent of $\varphi_i(x)$ in the representation of $\vartheta_i(x)$ as a power product of $\varphi_1(x), \ldots, \varphi_{t-1}(x)$ is less than $E_i^+ F_i^+$. This gives less than $\log N$ multiplications of polynomials of degree less than N. As in (e) the exponents of at most $\log N$ of the $\varphi_i(x)$ are nonzero. Therefore in total this step can be conducted in $O(N^{2+\varepsilon})$ operations in $\mathcal{O}_\mathsf{K}[x]$.

By Theorem 5 the maximum of the valuations $\nu(v^*_\Phi(\xi))$, where ξ is a root of $\Phi(x)$, is less than $2\big(\nu(\operatorname{disc}\Phi)\big)/N$. This is also the maximal (absolute) slope of the Newton polygon of the polynomials under consideration. Therefore a precision of $2\nu(\operatorname{disc}\Phi)$ is sufficient for all operations in the main loop.

Theorem 1. *Let p be a fixed prime. We can find a breaking element or a two element certificate for the irreducibility of a polynomial $\Phi(x) \in \mathbb{Z}_p[x]$ in at most $O(N^{2+\varepsilon}\nu(\operatorname{disc}\Phi)^{2+\varepsilon})$ operations of integers less than p.*

10 Example

We show that $\Phi(x) = x^{32} + 16 \in \mathbb{Z}_2[x]$ is irreducible using Algorithm 16.

Initially we set $\varphi_1(x) = x$, $E_0 = 1$, $F_0 = 1$, $\mathsf{K}_0 = \mathbb{Q}_2$.

(a) The φ_1-expansion of $\Phi(x)$ is $\Phi(x) = \sum_{i=0}^{32} a_i(x)\varphi_0(x)^i = x^{32} + 16$.
(b) The valuations of the coefficients are $v^*_\Phi(a_0) = 4$, $v^*_\Phi(a_i) = \infty$ for $1 \le i \le 31$, and $v^*_\Phi(a_{32}) = 0$.
(c,d) $\varphi_1(x)$ passes the Newton test; we get $v^*_\Phi(\varphi_1) = \frac{h_1}{e_1} = \frac{4}{32} = \frac{1}{8}$, so $E_1^+ = 8$ and $E_1 = 8$.
(e) We set $\psi_1(x) = 2$ as $v^*_\Phi(\varphi_1^{E_1^+}) = v^*_\Phi(x^8) = 1$.
(f,g) $A_1(z) = z^4 + 1$ with $\underline{A}_1(z) = (z-1)^4$ in $\mathbb{F}_2[z]$.
(h,i) $\frac{\varphi_1(x)^8}{\psi_1(x)}$ passes the Hensel test; we get $F_1^+ = 1$, $\mathsf{K}_1 = \mathbb{Q}_2$, $F_1 = 1$.
(k) We obtain the next approximation of an irreducible factor of $\Phi(x)$:

$$\varphi_2(x) = 2\left(\frac{x^8}{2} - 1\right) = x^8 - 2.$$

Second iteration:

(a) The φ_2-expansion of $\Phi(x)$ is

$$\Phi(x) = \varphi_2(x)^4 + 8\varphi_2(x)^3 + 24\varphi_2(x)^2 + 32\varphi_2(x) + 32.$$

(b) The valuations of the coefficients are $v^*_\Phi(32) = 5$, $v^*_\Phi(24) = 3$, $v^*_\Phi(8) = 3$, and $v^*_\Phi(1) = 0$.
(c,d) $\varphi_2(x)$ passes the Newton test; we get $\frac{h_2}{e_2} = \frac{5}{4}$, so $E_2^+ = 1$, $E_2 = 8$.
(e) We set $\psi_2(x) = \frac{x^2}{2}$, so that $v^*_\Phi(\psi_2) = \frac{5}{4}$.
(f,g) The associated polynomial with respect to $\varphi_2(x)$ is $A_2(z) = z^4 + 1 = (z-1)^4 \in \mathbb{F}_2[z]$.
(h,i) $\frac{\varphi_2(x)}{\psi_2(x)}$ passes the Hensel test, we get $F_2^+ = 1$, $\mathsf{K}_2 = \mathbb{Q}_2$, $F_2 = 1$.
(l) We set

$$\varphi_3(x) = \psi_2(x)\left(\frac{\varphi_2(x)}{\psi_2(x)} - 1\right) = x^8 - 2x^2 - 2.$$

Third iteration:

(a) The φ_3-expansion of $\Phi(x)$ is

$$\Phi(x) = \varphi_3(x)^4 + a_3(x)\varphi_3(x)^3 + a_2(x)\varphi_3(x)^2 + a_1(x)\varphi_3(x) + a_0(x)$$

where $a_3(x) = 8x^2 + 8$, $a_2(x) = 24x^4 + 48x^2 + 24$, $a_1(x) = 32x^6 + 96x^4 + 96x^2 + 48$, $a_0(x) = 64x^6 + 96x^4 + 96x^2 + 64$.
(b) The valuations of the coefficients are $v^*_\Phi(a_0) = \frac{21}{4}$, $v^*_\Phi(a_1) = 4$, $v^*_\Phi(a_2) = 3$, $v^*_\Phi(a_3) = 3$, and $v^*_\Phi(1) = 0$.
(c,d) $\varphi_3(x)$ passes the Newton test; we get $v^*_\Phi(\varphi_3) = \frac{h_3}{e_3} = \frac{21}{16}$, $E_3^+ = 2$, $E_3 = 16$.
(e) We find $\psi_3(x) = 2^2x^5$; so that $v^*_\Phi(\psi_3) = v^*_\Phi(\varphi_3^{E_3^+}) = \frac{21}{8}$.
(f,g) The associated polynomial with respect to $\varphi_3(x)$ is $\underline{A}_2(z) = z^2 + 3 = (z-1)^3 \in \mathbb{F}_2[z]$.
(h,i) $\frac{\varphi_3(x)}{\psi_3(x)}$ passes the Hensel test; we get $F_3^+ = 1$, $\mathsf{K}_3 = \mathbb{Q}_2$, $F_3 = 1$.
(l) We set

$$\varphi_4(x) = x^{16} - 4x^{10} - 4x^8 - 4x^5 + 4x^4 + 8x^2 + 4.$$

Fourth iteration:

(a) Let $\Phi(x) = \varphi_4(x)^2 + a_1(x)\varphi_4(x) + a_0(x)$ be the φ_4-expansion of $\Phi(x)$.
(b) We have $v^*_\Phi(a_0) = 85/16$ and $v^*_\Phi(a_1) = 3$.
(c,d) $\varphi_4(x)$ passes the Newton test; we get $\frac{h_4}{e_4} = \frac{85}{32}$, $E_4^+ = 2$, $E_4 = 32$.
(g) Now $E_4F_4 = 32 = \deg\Phi$ which implies the irreducibility of $\Phi(x) = x^{32} + 16$.

Acknowledgments

The author would like to thank the anonymous referees and David Ford for their numerous comments. He apologizes to them for the large number of small mistakes.

References

1. Cannon, J.J., et al.: The computer algebra system Magma. University of Sydney (2010), http://magma.maths.usyd.edu.au/magma/
2. Cantor, D.G., Gordon, D.: Factoring polynomials over p-adic fields. In: Bosma, W. (ed.) ANTS 2000. LNCS, vol. 1838, pp. 185–208. Springer, Heidelberg (2000)
3. Ford, D., Letard, P.: Implementing the Round Four maximal order algorithm. Journal de Théorie des Nombres de Bordeaux 6, 39–80 (1994)
4. Ford, D., Pauli, S., Roblot, X.-F.: A Fast Algorithm for Polynomial Factorization over $\mathbb{Q}_p$. Journal de Théorie des Nombres de Bordeaux 14, 151–169 (2002)
5. Ford, D., Veres, O.: On the Complexity of the Montes Ideal Factorization Algorithm. In: Hanrot, G., Morain, F., Thomé, E. (eds.) ANTS-IX, July 19-23. LNCS, vol. 6197, pp. 174–185. Springer, Heidelberg (2010)
6. Kaltofen, E., Shoup, V.: Subquadratic-time factoring of polynomials over finite fields. Math. Comp. 67 (1998)
7. Guardia, J., Montes, J., Nart, E.: Newton polygons of higher order in algebraic number theory (2008), arXiv:0807.2620
8. Guardia, J., Montes, J., Nart, E.: Higher Newton polygons in the computation of discriminants and prime ideal decomposition in number fields (2008), arXiv:0807.4065
9. MacLane, S.: A Construction for absolute values in polynomial rings. Trans. Amer. Math. Soc. 40, 363–395 (1936)
10. Montes, J., Nart, E.: On a Theorem of Ore. Journal of Algebra 146, 318–334 (1992)
11. Montes, J.: Polígonos de Newton de orden superior y aplicaciones aritméticas, PhD Thesis, Universitat de Barcelona (1999)
12. Ore, Ö.: Newtonsche Polygone in der Theorie der algebraischen Körper. Math. Ann. 99, 84–117 (1928)
13. PARI/GP, version 2.3.4, Bordeaux (2008), http://pari.math.u-bordeaux.fr/
14. Pauli, S.: Factoring polynomials over local fields. J. Symb. Comp. 32, 533–547 (2001)
15. Schönhage, A., Strassen, V.: Schnelle Multiplikation großer Zahlen. Computing 7, 281–292 (1971)
16. Stein, W., et al.: SAGE: Software for Algebra and Geometry Experimentation (2007), http://www.sagemath.org
17. Veres, O.: On the Complexity of Polynomial Factorization over p-adic Fields, PhD Dissertation, Concordia University, Montreal (2009)

On a Problem of Hajdu and Tengely

Samir Siksek[1] and Michael Stoll[2]

[1] Institute of Mathematics, University of Warwick, Coventry CV4 7AL, UK
s.siksek@warwick.ac.uk
[2] Mathematisches Institut, Universität Bayreuth, 95440 Bayreuth, Germany
Michael.Stoll@uni-bayreuth.de

Abstract. We prove a result that finishes the study of primitive arithmetic progressions consisting of squares and fifth powers that was carried out by Hajdu and Tengely in a recent paper: The only arithmetic progression in coprime integers of the form (a^2, b^2, c^2, d^5) is $(1, 1, 1, 1)$. For the proof, we first reduce the problem to that of determining the sets of rational points on three specific hyperelliptic curves of genus 4. A 2-cover descent computation shows that there are no rational points on two of these curves. We find generators for a subgroup of finite index of the Mordell-Weil group of the last curve. Applying Chabauty's method, we prove that the only rational points on this curve are the obvious ones.

1 Introduction

Euler ([9, pages 440 and 635]) proved Fermat's claim that four distinct squares cannot form an arithmetic progression. Powers in arithmetic progressions are still a subject of current interest. For example, Darmon and Merel [8] proved that the only solutions in coprime integers to the Diophantine equation $x^n + y^n = 2z^n$ with $n \geq 3$ satisfy $xyz = 0$ or ± 1. This shows that there are no non-trivial three term arithmetic progressions consisting of n-th powers with $n \geq 3$. The result of Darmon and Merel is far from elementary; it needs all the tools used in Wiles' proof of Fermat's Last Theorem and more.

An arithmetic progression $(x_1, x_2, \ldots, x_k)$ of integers is said to be *primitive* if the terms are coprime, i.e., if $\gcd(x_1, x_2) = 1$. Let S be a finite subset of integers ≥ 2. Hajdu [11] showed that if

$$(a_1^{\ell_1}, \ldots, a_k^{\ell_k}) \tag{1}$$

is a non-constant primitive arithmetic progression with $\ell_i \in S$, then k is bounded by some (inexplicit) constant $C(S)$. Bruin, Győry, Hajdu and Tengely [2] showed that for any $k \geq 4$ and any S, there are only finitely many primitive arithmetic progressions of the form (1), with $\ell_i \in S$. Moreover, for $S = \{2, 3\}$ and $k \geq 4$, they showed that $a_i = \pm 1$ for $i = 1, \ldots, k$.

A recent paper of Hajdu and Tengely [12] studies primitive arithmetic progressions (1) with exponents belonging to $S = \{2, n\}$ and $\{3, n\}$. In particular, they

G. Hanrot, F. Morain, and E. Thomé (Eds.): ANTS-IX 2010, LNCS 6197, pp. 316–330, 2010.

show that any primitive non-constant arithmetic progression (1) with exponents $\ell_i \in \{2, 5\}$ has $k \leq 4$. Moreover, for $k = 4$ they show that

$$(\ell_1, \ell_2, \ell_3, \ell_4) = (2, 2, 2, 5) \quad \text{or} \quad (5, 2, 2, 2). \tag{2}$$

Note that if $(a_i^{\ell_i} : i = 1, \ldots, k)$ is an arithmetic progression, then so is the reverse progression $(a_i^{\ell_i} : i = k, k-1, \ldots, 1)$. Thus there is really only one case left open by Hajdu and Tengely, with exponents $(\ell_1, \ell_2, \ell_3, \ell_4) = (2, 2, 2, 5)$. This is also mentioned as Problem 11 in a list of 22 open problems recently compiled by Evertse and Tijdeman [10]. In this paper we deal with this case.

Theorem 1. *The only arithmetic progression in coprime integers of the form*

$$(a^2, b^2, c^2, d^5)$$

is $(1, 1, 1, 1)$.

This together with the above-mentioned results of Hajdu and Tengely completes the proof of the following theorem.

Theorem 2. *There are no non-constant primitive arithmetic progressions of the form* (1) *with* $\ell_i \in \{2, 5\}$ *and* $k \geq 4$.

The primitivity condition is crucial, since otherwise solutions abound. Let for example (a^2, b^2, c^2, d) be any arithmetic progression whose first three terms are squares — there are infinitely many of these; one can take $a = r^2 - 2rs - s^2$, $b = r^2 + s^2$, $c = r^2 + 2rs - s^2$ — then $\big((ad^2)^2, (bd^2)^2, (cd^2)^2, d^5\big)$ is an arithmetic progression whose first three terms are squares and whose last term is a fifth power.

For the proof of Thm. 1, we first reduce the problem to that of determining the sets of rational points on three specific hyperelliptic curves of genus 4. A 2-cover descent computation (following Bruin and Stoll [3]) shows that there are no rational points on two of these curves. We find generators for a subgroup of finite index of the Mordell-Weil group of the last curve. Applying Chabauty's method, we prove that the only rational points on this curve are the obvious ones. All our computations are performed using the computer package MAGMA [1].

The result we prove here may perhaps not be of compelling interest in itself. Rather, the purpose of this paper is to demonstrate how we can solve problems of this kind with the available machinery. We review the relevant part of this machinery in Sect. 3, after we have constructed the curves pertaining to our problem in Sect. 2. Then, in Sect. 4, we apply the machinery to these curves. The proofs are mostly computational. We have tried to make it clear what steps need to be done, and to give enough information to make it possible to reproduce the computations (which have been performed independently by both authors as a consistency check).

2 Construction of the Curves

Let (a^2, b^2, c^2, d^5) be an arithmetic progression in coprime integers. Since a square is $\equiv 0$ or $1 \bmod 4$, it follows that all terms are $\equiv 1 \bmod 4$, in particular, a, b, c and d are all odd.

Considering the last three terms, we have the relation

$$(-d)^5 = b^2 - 2c^2 = (b + c\sqrt{2})(b - c\sqrt{2}) .$$

Since b and c are odd and coprime, the two factors on the right are coprime in $R = \mathbb{Z}[\sqrt{2}]$. Since $R^\times/(R^\times)^5$ is generated by $1 + \sqrt{2}$, it follows that

$$b + c\sqrt{2} = (1 + \sqrt{2})^j (u + v\sqrt{2})^5 = g_j(u, v) + h_j(u, v)\sqrt{2} \tag{3}$$

with $-2 \le j \le 2$ and $u, v \in \mathbb{Z}$ coprime (with u odd and $v \equiv j + 1 \bmod 2$). The polynomials g_j and h_j are homogeneous of degree 5 and have coefficients in $\mathbb{Z}$.

Now the first three terms of the progression give the relation

$$a^2 = 2b^2 - c^2 = 2g_j(u, v)^2 - h_j(u, v)^2 .$$

Writing $y = a/v^5$ and $x = u/v$, this gives the equation of a hyperelliptic curve of genus 4,

$$C_j : y^2 = f_j(x)$$

where $f_j(x) = 2g_j(x, 1)^2 - h_j(x, 1)^2$. Every arithmetic progression of the required form therefore induces a rational point on one of the curves C_j.

We observe that taking conjugates in (3) leads to

$$(-1)^j b + (-1)^{j+1} c\sqrt{2} = (1 + \sqrt{2})^{-j} (u + (-v)\sqrt{2})^5 ,$$

which implies that $f_{-j}(x) = f_j(-x)$ and therefore that C_{-j} and C_j are isomorphic and their rational points correspond to the same arithmetic progressions. We can therefore restrict attention to C_0, C_1 and C_2. Their equations are as follows.

$$\begin{aligned}
C_0 : y^2 = f_0(x) &= 2x^{10} + 55x^8 + 680x^6 + 1160x^4 + 640x^2 - 16 \\
C_1 : y^2 = f_1(x) &= x^{10} + 30x^9 + 215x^8 + 720x^7 + 1840x^6 + 3024x^5 \\
&\quad + 3880x^4 + 2880x^3 + 1520x^2 + 480x + 112 \\
C_2 : y^2 = f_2(x) &= 14x^{10} + 180x^9 + 1135x^8 + 4320x^7 + 10760x^6 + 18144x^5 \\
&\quad + 21320x^4 + 17280x^3 + 9280x^2 + 2880x + 368
\end{aligned}$$

The trivial solution $a = b = c = d = 1$ corresponds to $j = 1$, $(u, v) = (1, 0)$ in the above and therefore gives rise to the point ∞_+ on C_1 (this is the point at infinity where y/x^5 takes the value $+1$). Changing the signs of a, b or c leads to $\infty_- \in C_1(\mathbb{Q})$ (the point where $y/x^5 = -1$) or to the two points at infinity on the isomorphic curve C_{-1}.

3 Background on Rational Points on Hyperelliptic Curves

Our task will be to determine the set of rational points on each of the curves C_0, C_1 and C_2 constructed in the previous section. In this section, we will give an overview of the methods we will use, and in the next section, we will apply these methods to the given curves.

We will restrict attention to *hyperelliptic* curves, i.e., curves given by an affine equation of the form

$$C : y^2 = f(x)$$

where f is a squarefree polynomial with integral coefficients. The smooth projective curve birational to this affine curve has either one or two additional points 'at infinity'. If the degree of f is odd, there is one point at infinity, which is always a rational point. Otherwise there are two points at infinity corresponding to the two square roots of the leading coefficient of f. In particular, these two points are rational if and only if the leading coefficient is a square. For example, C_1 above has two rational points at infinity, whereas the points at infinity on C_0 and C_2 are not rational. We will use C in the following to denote the smooth projective model; $C(\mathbb{Q})$ denotes as usual the set of rational points including those at infinity.

3.1 Two-Cover Descent

It will turn out that C_0 and C_2 do not have rational points. One way of showing that $C(\mathbb{Q})$ is empty is to verify that $C(\mathbb{R})$ is empty or that $C(\mathbb{Q}_p)$ is empty for some prime p. This does not work for C_0 or C_2; both curves have real points and p-adic points for all p. (This can be checked by a finite computation.) So we need a more sophisticated way of showing that there are no rational points. One such method is known as *2-cover descent*. We sketch the method here; for a detailed description, see [3].

An important ingredient of this and other methods is the algebra

$$L := \mathbb{Q}[T] = \frac{\mathbb{Q}[x]}{\mathbb{Q}[x] \cdot f(x)},$$

where T denotes the image of x. If f is irreducible (as in our examples), then L is the number field generated by a root of f. In general, L will be a product of number fields corresponding to the irreducible factors of f. We now assume that f has even degree $2g + 2$, where g is the genus of the curve. This is the generic case; the odd degree case is somewhat simpler. We can then set up a map, called the *descent map* or $x - T$ *map*:

$$x - T : C(\mathbb{Q}) \longrightarrow H := \frac{L^\times}{\mathbb{Q}^\times (L^\times)^2}.$$

Here $L^\times$ denotes the multiplicative group of L, and $(L^\times)^2$ denotes the subgroup of squares. On points $P \in C(\mathbb{Q})$ that are neither at infinity nor Weierstrass points (i.e., points with vanishing y coordinate), the map is defined as

$$(x - T)(P) = x(P) - T \bmod \mathbb{Q}^\times (L^\times)^2 .$$

Rational points at infinity map to the trivial element, and if there are rational Weierstrass points, their images can be determined using the fact that the norm of $x(P) - T$ is $y(P)^2$ divided by the leading coefficient of f. If we can show that $x - T$ has empty image on $C(\mathbb{Q})$, then it follows that $C(\mathbb{Q})$ is empty.

We obtain information of the image by considering again $C(\mathbb{R})$ and $C(\mathbb{Q}_p)$. We can carry out the same construction over $\mathbb{R}$ and over $\mathbb{Q}_p$, leading to an algebra L_v ($v = p$, or $v = \infty$ when working over $\mathbb{R}$), a group H_v and a map

$$(x - T)_v : C(\mathbb{Q}_v) \longrightarrow H_v \qquad (\text{where } \mathbb{Q}_\infty = \mathbb{R}).$$

We have inclusions $C(\mathbb{Q}) \hookrightarrow C(\mathbb{Q}_v)$ and canonical homomorphisms $H \to H_v$. Everything fits together in a commutative diagram

$$\begin{array}{ccc} C(\mathbb{Q}) & \xrightarrow{x-T} & H \\ \downarrow & & \downarrow \\ \prod_v C(\mathbb{Q}_v) & \xrightarrow{\prod_v (x-T)_v} & \prod_v H_v \end{array}$$

where v runs through the primes and ∞. If we can show that the images of the lower horizontal map and of the right vertical map do not meet, then the image of $x - T$ and therefore also $C(\mathbb{Q})$ must be empty. We can verify this by considering a finite subset of 'places' v.

In general, we obtain a finite subset of H that contains the image of $x-T$; this finite subset is known as the *fake 2-Selmer set* of $C/\mathbb{Q}$. It classifies either pairs of (isomorphism classes of) 2-covering curves of C that have points *everywhere locally*, i.e., over $\mathbb{R}$ and over all $\mathbb{Q}_p$, or else it classifies such 2-covering curves, in which case it is the (true) 2-Selmer set. Whether it classifies pairs or individual 2-coverings depends on a certain condition on the polynomial f. This condition is satisfied if either f has an irreducible factor of odd degree, or if $\deg f \equiv 2 \bmod 4$ and f factors over a quadratic extension $\mathbb{Q}(\sqrt{d})$ as a constant times the product of two conjugate polynomials. A *2-covering* of C is a morphism $\pi : D \to C$ that is unramified and becomes Galois over a suitable field extension of finite degree, with Galois group $(\mathbb{Z}/2\mathbb{Z})^{2g}$. It is known that every rational point on C lifts to a rational point on some 2-covering of C.

The actual computation splits into a global and a local part. The global computation uses the ideal class group and the unit group of L (or the constituent number fields of L) to construct a finite subgroup of H containing the image of $x - T$. The local computation determines the image of $(x - T)_v$ for finitely many places v.

3.2 The Jacobian

Most other methods make use of another object associated to the curve C: its *Jacobian variety* (or just *Jacobian*). This is an abelian variety J (a higher-dimensional analogue of an elliptic curve) of dimension g, the genus of C. It

reflects a large part of the geometry and arithmetic of C; its main advantage is that its points form an abelian group, whereas the set of points on C does not carry a natural algebraic structure.

For our purposes, we can more or less forget the structure of J as a projective variety. Instead we use the description of the points on J as the elements of the degree zero part of the *Picard group* of C. The Picard group is constructed as a quotient of the group of divisors on C. A *divisor* on C is an element of the free abelian group Div_C on the set $C(\bar{\mathbb{Q}})$ of all algebraic points on C. The absolute Galois group of $\mathbb{Q}$ acts on Div_C; a divisor that is fixed by this action is *rational*. This does not mean that the points occurring in the divisor must be rational; points with the same multiplicity can be permuted. A nonzero rational function h on C with coefficients in $\bar{\mathbb{Q}}$ has an associated divisor $\mathrm{div}(h)$ that records its zeros and poles (with multiplicities). If h has coefficients in $\mathbb{Q}$, then $\mathrm{div}(h)$ is rational. The homomorphism $\deg : \mathrm{Div}_C \to \mathbb{Z}$ induced by sending each point in $C(\bar{\mathbb{Q}})$ to 1 gives the *degree* of a divisor. Divisors of functions have degree zero.

Two divisors $D, D' \in \mathrm{Div}_C$ are *linearly equivalent* if their difference is the divisor of a function. The equivalence classes are the elements of the *Picard group* Pic_C defined by the following exact sequence.

$$0 \longrightarrow \bar{\mathbb{Q}}^\times \longrightarrow \bar{\mathbb{Q}}(C)^\times \xrightarrow{\mathrm{div}} \mathrm{Div}_C \longrightarrow \mathrm{Pic}_C \longrightarrow 0$$

Since divisors of functions have degree zero, the degree homomorphism descends to Pic_C. We denote its kernel by Pic^0_C. It is a fact that $J(\bar{\mathbb{Q}})$ is isomorphic as a group to Pic^0_C. The rational points $J(\mathbb{Q})$ correspond to the elements of Pic^0_C left invariant by the Galois group. In general it is not true that a point in $J(\mathbb{Q})$ can be represented by a rational divisor, but this is the case when C has a rational point, or at least points everywhere locally. The most important fact about the group $J(\mathbb{Q})$ is the statement of the *Mordell-Weil Theorem:* $J(\mathbb{Q})$ is a *finitely generated* abelian group. For this reason, $J(\mathbb{Q})$ is often called the *Mordell-Weil group* of J or of C.

If $P_0 \in C(\mathbb{Q})$, then the map $C \ni P \mapsto [P - P_0] \in J$ is a $\mathbb{Q}$-defined embedding of C into J. We use $[D]$ to denote the linear equivalence class of the divisor D. The basic idea of the methods described below is to try to recognise the points of C embedded in this way among the rational points on J.

We need a way of representing elements of $J(\mathbb{Q})$. Let $P \mapsto P^-$ denote the *hyperelliptic involution* on C; this is the morphism $C \to C$ that changes the sign of the y coordinate. Then it is easy to see that the divisors $P + P^-$ all belong to the same class $W \in \mathrm{Pic}_C$. An effective divisor D (a divisor such that no point occurs with negative multiplicity) is *in general position* if there is no point P such that $D - P - P^-$ is still effective. Divisors in general position not containing points at infinity can be represented in a convenient way by pairs of polynomials $(a(x), b(x))$. This pair represents the divisor D such that its image on the projective line (under the x-coordinate map) is given by the roots of a; the corresponding points on C are determined by the relation $y = b(x)$. The polynomials have to satisfy the relation $f(x) \equiv b(x)^2 \bmod a(x)$. This is

the *Mumford representation* of D. The polynomials a and b can be chosen to have rational coefficients if and only if D is rational. (The representation can be adapted to allow for points at infinity occurring in the divisor.)

If the genus g is even, then it is a fact that every point in $J(\mathbb{Q})$ has a unique representation of the form $[D] - nW$ where D is a rational divisor in general position of degree $2n$ and $n \geq 0$ is minimal. The Mumford representation of D is then also called the Mumford representation of the corresponding point on J. It is fairly easy to add points on J using the Mumford representation, see [5]. This addition procedure is implemented in MAGMA, for example.

There is a relation between 2-coverings of C and the Jacobian J. Assume C is embedded in J as above. Then if D is any 2-covering of C that has a rational point P, D can be realised as the preimage of C under a map of the form $Q \mapsto 2Q + Q_0$ on J, where Q_0 is the image of P on $C \subset J$. A consequence of this is that two rational points $P_1, P_2 \in C(\mathbb{Q})$ lift to the same 2-covering if and only if $[P_1 - P_2] \in 2J(\mathbb{Q})$.

3.3 The Mordell-Weil Group

We will need to know generators of a finite-index subgroup of the Mordell-Weil group $J(\mathbb{Q})$. Since $J(\mathbb{Q})$ is a finitely generated abelian group, it will be a direct sum of a finite torsion part and a free abelian group of rank r; r is called the *rank* of $J(\mathbb{Q})$. So what we need is a set of r independent points in $J(\mathbb{Q})$.

The torsion subgroup of $J(\mathbb{Q})$ is usually easy to determine. The main tool used here is the fact that the torsion subgroup injects into $J(\mathbb{F}_p)$ when p is an odd prime not dividing the discriminant of f. If the orders of the finite groups $J(\mathbb{F}_p)$ are coprime for suitable primes p, then this shows that $J(\mathbb{Q})$ is torsion-free.

We can find points in $J(\mathbb{Q})$ by search. This can be done by searching for rational points on the variety parameterising Mumford representations of divisors of degree 2, 4, ... We can then check if the points found are independent by again mapping into $J(\mathbb{F}_p)$ for one or several primes p.

The hard part is to know when we have found enough points. For this we need an upper bound on the rank r. This can be provided by a *2-descent* on the Jacobian J. This is described in detail in [16]. The idea is similar to the 2-cover descent on C described above in Sect. 3.1. Essentially we extend the $x - T$ map from points to divisors. It can be shown that the value of $(x - T)(D)$ only depends on the linear equivalence class of D. This gives us a homomorphism from $J(\mathbb{Q})$ into H, or more precisely, into the kernel of the norm map $N_{L/\mathbb{Q}} : H \to \mathbb{Q}^\times/(\mathbb{Q}^\times)^2$. It can be shown that the kernel of this $x - T$ map on $J(\mathbb{Q})$ is either $2J(\mathbb{Q})$, or it contains $2J(\mathbb{Q})$ as a subgroup of index 2. The former is the case when f satisfies the same condition as that mentioned in Sect. 3.1.

We can then bound $(x - T)(J(\mathbb{Q}))$ in much the same way as we did when doing a 2-cover descent on C. The global part of the computation is identical. The local part is helped by the fact that we now have a group homomorphism (or a homomorphism of $\mathbb{F}_2$-vector spaces), so we can use linear algebra. We obtain a bound for the order of $J(\mathbb{Q})/2J(\mathbb{Q})$, from which we can deduce a bound for

the rank r. If we are lucky and found that same number of independent points in $J(\mathbb{Q})$, then we know that these points generate a subgroup of finite index.

The group containing $(x-T)(J(\mathbb{Q}))$ we compute is known as the *fake 2-Selmer group* of J [13]. If the polynomial f satisfies the relevant condition, then this fake Selmer group is isomorphic to the true 2-Selmer group of J (that classifies 2-coverings of J that have points everywhere locally).

3.4 The Chabauty-Coleman Method

If the rank r is less than the genus g, there is a method available that allows us to get tight bounds on the number of rational points on C. This goes back to Chabauty [6], who used it to prove Mordell's Conjecture in this case. Coleman [7] refined the method. We give a sketch here; more details can be found for example in [15].

Let p be a prime of good reduction for C (this is the case when p is odd and does not divide the discriminant of f). We use $\Omega^1_C(\mathbb{Q}_p)$ and $\Omega^1_J(\mathbb{Q}_p)$ to denote the spaces of regular 1-forms on C and J that are defined over $\mathbb{Q}_p$. If $P_0 \in C(\mathbb{Q})$ and $\iota : C \to J$, $P \mapsto [P - P_0]$ denotes the corresponding embedding of C into J, then the induced map $\iota^* : \Omega^1_J(\mathbb{Q}_p) \to \Omega^1_C(\mathbb{Q}_p)$ is an isomorphism that is independent of the choice of basepoint P_0. Both spaces have dimension g. There is an integration pairing

$$\Omega^1_C(\mathbb{Q}_p) \times J(\mathbb{Q}_p) \longrightarrow \mathbb{Q}_p, \quad (\iota^*\omega, Q) \longmapsto \int_0^Q \omega = \langle \omega, \log Q \rangle \,.$$

In the last expression, $\log Q$ denotes the p-adic logarithm on $J(\mathbb{Q}_p)$ with values in the tangent space of $J(\mathbb{Q}_p)$ at the origin, and $\Omega^1_J(\mathbb{Q}_p)$ is identified with the dual of this tangent space. If $r < g$, then there are (at least) $g-r$ linearly independent differentials $\omega \in \Omega^1_C(\mathbb{Q}_p)$ that annihilate the Mordell-Weil group $J(\mathbb{Q})$. Such a differential can be scaled so that it reduces to a non-zero differential $\bar{\omega}$ mod p. Now the important fact is that if $\bar{\omega}$ does not vanish at a point $\bar{P} \in C(\mathbb{F}_p)$, then there is at most one rational point on $C(\mathbb{Q})$ whose reduction is $\bar{P}$. (There are more general bounds valid when $\bar{\omega}$ does vanish at $\bar{P}$, but we do not need them here.)

4 Determining the Rational Points

In this section, we determine the set of rational points on the three curves C_0, C_1 and C_2. To do this, we apply the methods described in Sect. 3.

We first consider C_0 and C_2. We apply the 2-cover-descent procedure described in Sect. 3.1 to the two curves and find that in each case, there are no 2-coverings that have points everywhere locally. For C_0, only 2-adic information is needed in addition to the global computation, for C_2, we need 2-adic and 7-adic information. Note that the number fields generated by roots of f_0 or f_2 are sufficiently small in terms of degree and discriminant that the necessary class and unit group computations can be done unconditionally. This leads to the following.

Proposition 3. *There are no rational points on the curves C_0 and C_2.*

Proof. The 2-cover descent procedure is available in recent releases of MAGMA. The computations leading to the stated result can be performed by issuing the following MAGMA commands.

```
> SetVerbose("Selmer",2);
> TwoCoverDescent(HyperellipticCurve(Polynomial(
    [-16,0,640,0,1160,0,680,0,55,0,2])));
> TwoCoverDescent(HyperellipticCurve(Polynomial(
    [368,2880,9280,17280,21320,18144,10760,4320,1135,180,14])));
```

We explain how the results can be checked independently. We give details for C_0 first. The procedure for C_2 is similar, so we only explain the differences.

The polynomial f_0 is irreducible, and it can be checked that the number field generated by one of its roots is isomorphic to $L = \mathbb{Q}(\sqrt[10]{288})$. Using MAGMA or pari/gp, one checks that this field has trivial class group. The finite subgroup $\tilde{H}$ of H containing the Selmer set is then given as $\mathcal{O}_{L,S}^{\times}/(\mathbb{Z}_{\{2,3,5\}}^{\times}(\mathcal{O}_{L,S}^{\times})^2)$, where S is the set of primes in $\mathcal{O}_L$ above the 'bad primes' 2, 3 and 5. The set S contains two primes above 2, of degrees 1 and 4, respectively, and one prime above 3 and 5 each, of degree 2 in both cases. Since L has two real embeddings and four pairs of complex embeddings, the unit rank is 5. The rank (or $\mathbb{F}_2$-dimension) of $\tilde{H}$ is then 7. (Note that 2 is a square in L.) The descent map takes its values in the subset of $\tilde{H}$ consisting of elements whose norm is twice a square. This subset is of size 32; elements of $\mathcal{O}_L$ representing it can easily be obtained. Let δ be such a representative. We let T be a root of f_0 in L and check that the system of equations

$$y^2 = f_0(x), \quad x - T = \delta c z^2$$

has no solutions with $x, y, c \in \mathbb{Q}_2$, $z \in L \otimes_{\mathbb{Q}} \mathbb{Q}_2$. The second equation leads, after expanding δz^2 as a $\mathbb{Q}$-linear combination of $1, T, T^2, \ldots, T^9$, to eight homogeneous quadratic equations in the ten unknown coefficients of z. Any solution to these equations gives a unique x, for which $f_0(x)$ is a square. The latter follows by taking norms on both sides of $x - T = \delta c z^2$. So we only have to check the intersection of eight quadrics in $\mathbb{P}^9$ for existence of $\mathbb{Q}_2$-points. Alternatively, we evaluate the descent map on $C_0(\mathbb{Q}_2)$, to get its image in $H_2 = L_2^{\times}/(\mathbb{Q}_2^{\times}(L_2^{\times})^2)$, where $L_2 = L \otimes_{\mathbb{Q}} \mathbb{Q}_2$. Then we check that none of the representatives δ map into this image.

When dealing with C_2, the field L is generated by a root of $x^{10} - 6x^5 - 9$. Since the leading coefficient of f_2 is 14, we have to add (the primes above) 7 to the bad primes. As before, the class group is trivial, and we have the same splitting behaviour of 2, 3 and 5. The prime 7 splits into two primes of degree 1 and two primes of degree 4. The group of S-units of L modulo squares has now rank 14, the group $\tilde{H}$ has rank 10, and the subset of H consisting of elements whose norm is 14 times a square has 128 elements. These elements now have to be tested for compatibility with the 2-adic and the 7-adic information, which can be done using either of the two approaches described above. The 7-adic check is

only necessary for one of the elements; the 127 others are already ruled out by the 2-adic check. □

We cannot hope to deal with C_1 in the same easy manner, since C_1 has two rational points at infinity coming from the trivial solutions. We can still perform a 2-cover-descent computation, though, and find that there is only one 2-covering of C_1 with points everywhere locally, which is the covering that lifts the points at infinity. Only 2-adic information is necessary to show that the fake 2-Selmer set has at most one element, so we can get this result using the following MAGMA command.

```
> TwoCoverDescent(HyperellipticCurve(Polynomial(
    [112,480,1520,2880,3880,3024,1840,720,215,30,1]))
      : PrimeCutoff := 2);
```

(In some versions of MAGMA this returns a two-element set. However, as can be checked by pulling back under the map returned as a second value, these two elements correspond to the images of 1 and -1 in $L^\times/(L^\times)^2\mathbb{Q}^\times$ and therefore both represent the trivial element. The error is caused by MAGMA using 1 instead of -1 as a 'generator' of $\mathbb{Q}^\times/(\mathbb{Q}^\times)^2$. This bug is corrected in recent releases.)

The computation can be performed in the same way as for C_0 and C_2. The relevant field L is generated by a root of $x^{10} - 18x^5 + 9$; it has class number 1, and the primes 2, 3 and 5 split in the same way as before. The subset H' (in fact a subgroup) of $\tilde{H}$ consisting of elements with square norm has size 32. Of these, only the element represented by 1 is compatible with the 2-adic constraints.

We remark that by the way it is given, the polynomial f_1 factors over $\mathbb{Q}(\sqrt{2})$ into two conjugate factors of degree 5. This implies that the 'fake 2-Selmer set' computed by the 2-cover descent is the true 2-Selmer set, so that there is really only one 2-covering that corresponds to the only element of the set computed by the procedure. We state the result as a lemma. We fix $P_0 = \infty_- \in C_1$ as our basepoint and write J_1 for the Jacobian variety of C_1. Then, as described in Sect. 3.2,

$$\iota : C_1 \longrightarrow J_1\,, \quad P \longmapsto [P - P_0]$$

is an embedding defined over $\mathbb{Q}$.

Lemma 4. *Let $P \in C_1(\mathbb{Q})$. Then the divisor class $[P - P_0]$ is in $2J_1(\mathbb{Q})$.*

Proof. Let D be the unique 2-covering of C_1 (up to isomorphism) that has points everywhere locally. The fact that D is unique follows from the computation of the 2-Selmer set. Any rational point $P \in C_1(\mathbb{Q})$ lifts to a rational point on some 2-covering of C_1. In particular, this 2-covering then has a rational point, so it also satisfies the weaker condition that it has points everywhere locally. Since D is the only 2-covering of C_1 satisfying this condition, P_0 and P must both lift to a rational point on D. This implies by the remark at the end of Sect. 3.2 that $[P - P_0] \in 2J_1(\mathbb{Q})$. □

To make use of this information, we need to know $J_1(\mathbb{Q})$, or at least a subgroup of finite index. A computer search reveals two points in $J_1(\mathbb{Q})$, which are given in Mumford representation (see Sect. 3.2) as follows.

$$Q_1 = \left(x^4 + 4x^2 + \tfrac{4}{5}, \quad -16x^3 - \tfrac{96}{5}x\right)$$
$$Q_2 = \left(x^4 + \tfrac{24}{5}x^3 + \tfrac{36}{5}x^2 + \tfrac{48}{5}x + \tfrac{36}{5}, \quad -\tfrac{1712}{75}x^3 - \tfrac{976}{25}x^2 - \tfrac{1728}{25}x - \tfrac{2336}{25}\right)$$

We note that $2Q_1 = [\infty_+ - \infty_-]$; this makes Lemma 4 explicit for the known two points on C_1.

Lemma 5. *The Mordell-Weil group $J_1(\mathbb{Q})$ is torsion-free, and Q_1, Q_2 are linearly independent. In particular, the rank of $J_1(\mathbb{Q})$ is at least 2.*

Proof. The only primes of bad reduction for C_1 are 2, 3 and 5. It is known that the torsion subgroup of $J_1(\mathbb{Q})$ injects into $J_1(\mathbb{F}_p)$ when p is an odd prime of good reduction. Since $\#J_1(\mathbb{F}_7) = 2400$ and $\#J_1(\mathbb{F}_{41}) = 2633441$ are coprime, there can be no nontrivial torsion in $J_1(\mathbb{Q})$.

We check that the image of $\langle Q_1, Q_2\rangle$ in $J_1(\mathbb{F}_7)$ is not cyclic. This shows that Q_1 and Q_2 must be independent. □

The next step is to show that the Mordell-Weil rank is indeed 2. For this, we compute the 2-Selmer group of J_1 as sketched in Sect. 3.3 and described in detail in [16]. We give some details of the computation, since it is outside the scope of the functionality that is currently provided by MAGMA (or any other software package).

We first remind ourselves that f_1 factors over $\mathbb{Q}(\sqrt{2})$. This implies that the kernel of the $x - T$ map on $J(\mathbb{Q})$ is $2J(\mathbb{Q})$. Therefore the 'fake 2-Selmer group' that we compute is in fact the actual 2-Selmer group of J_1. Since $J_1(\mathbb{Q})$ is torsion-free, the order of the 2-Selmer group is an upper bound for 2^r, where r is the rank of $J_1(\mathbb{Q})$.

The global computation is the same as that we needed to do for the 2-cover descent. In particular, the Selmer group is contained in the group H' from above, consisting of the S-units of L with square norm, modulo squares and modulo $\{2, 3, 5\}$-units of $\mathbb{Q}$. For the local part of the computation, we have to compute the image of $J_1(\mathbb{Q}_p)$ under the local $x-T$ map for the primes p of bad reduction. We check that there is no 2-torsion in $J_1(\mathbb{Q}_3)$ and $J_1(\mathbb{Q}_5)$ (f_1 remains irreducible both over $\mathbb{Q}_3$ and over $\mathbb{Q}_5$). This implies that the targets of the local maps $(x - T)_3$ and $(x - T)_5$ are trivial, which means that these two primes need not be considered as bad primes for the descent computation. The real locus $C_1(\mathbb{R})$ is connected, which implies that there is no information coming from the local image at the infinite place. (Recall that C_1 denotes the smooth projective model of the curve. The real locus of the affine curve $y^2 = f_1(x)$ has two components, but they are connected to each other through the points at infinity.) Therefore, we only need to use 2-adic information in the computation. We set $L_2 = L \otimes_{\mathbb{Q}} \mathbb{Q}_2$ and compute the natural homomorphism

$$\mu_2 : H' \longrightarrow H_2 = \frac{L_2^\times}{\mathbb{Q}_2^\times (L_2^\times)^2}\,.$$

Let I_2 be the image of $J_1(\mathbb{Q}_2)$ in H_2. Then the 2-Selmer group is $\mu_2^{-1}(I_2)$.

It remains to compute I_2, which is the hardest part of the computation. The 2-torsion subgroup $J_1(\mathbb{Q}_2)[2]$ has order 2 (f_1 splits into factors of degrees 2 and 8 over $\mathbb{Q}_2$); this implies that $J_1(\mathbb{Q}_2)/2J_1(\mathbb{Q}_2)$ has dimension $g+1=5$ as an $\mathbb{F}_2$-vector space. This quotient is generated by the images of Q_1 and Q_2 and of three further points of the form $[D_i] - \frac{\deg D_i}{2}W$, where D_i is the sum of points on C_1 whose x-coordinates are the roots of

$$\begin{aligned} D_1 &: \quad \left(x-\tfrac{1}{2}\right)\left(x-\tfrac{1}{4}\right), \\ D_2 &: \quad x^2-2x+6, \\ D_3 &: \quad x^4+4x^3+12x^2+36, \end{aligned}$$

respectively. These points were found by a systematic search, using the fact that the local map $(x-T)_2$ is injective in our situation. We can therefore stop the search procedure as soon as we have found points whose images generate a five-dimensional $\mathbb{F}_2$-vector space. We thus find $I_2 \subset H_2$ and then can compute the 2-Selmer group. In our situation, μ_2 is injective, and the intersection of its image with I_2 is generated by the images of Q_1 and Q_2. Therefore, the $\mathbb{F}_2$-dimension of the 2-Selmer group is 2.

Lemma 6. *The rank of $J_1(\mathbb{Q})$ is 2, and $\langle Q_1, Q_2\rangle \subset J_1(\mathbb{Q})$ is a subgroup of finite odd index.*

Proof. The Selmer group computation shows that the rank is ≤ 2, and Lemma 5 shows that the rank is ≥ 2. Regarding the second statement, it is now clear that we have a subgroup of finite index. The observation stated just before the lemma shows that the given subgroup surjects onto the 2-Selmer group under the $x-T$ map. Since the kernel of the $x-T$ map is $2J_1(\mathbb{Q})$, this implies that the index is odd. □

Now we want to use the Chabauty-Coleman method sketched in Sect. 3.4 to show that ∞_+ and ∞_- are the only rational points on C_1. To keep the computations reasonably simple, we want to work at $p=7$, which is the smallest prime of good reduction.

For p a prime of good reduction, we write ρ_p for the two 'reduction mod p' maps $J_1(\mathbb{Q}) \to J_1(\mathbb{F}_p)$ and $C_1(\mathbb{Q}) \to C_1(\mathbb{F}_p)$.

Lemma 7. *Let $P \in C_1(\mathbb{Q})$. Then $\rho_7(P) = \rho_7(\infty_+)$ or $\rho_7(P) = \rho_7(\infty_-)$.*

Proof. Let $G = \langle Q_1, Q_2\rangle$ be the subgroup of $J_1(\mathbb{Q})$ generated by the two points Q_1 and Q_2. We find that $\rho_7(G)$ has index 2 in $J_1(\mathbb{F}_7) \cong \mathbb{Z}/10\mathbb{Z} \oplus \mathbb{Z}/240\mathbb{Z}$. By Lemma 6, we know that $(J_1(\mathbb{Q}) : G)$ is odd, so we can deduce that $\rho_7(G) = \rho_7(J_1(\mathbb{Q}))$. The group $J_1(\mathbb{F}_7)$ surjects onto $(\mathbb{Z}/5\mathbb{Z})^2$. Since $\rho_7(J_1(G))$ has index 2 in $J_1(\mathbb{F}_7)$, $\rho_7(G) = \rho_7(J_1(\mathbb{Q}))$ also surjects onto $(\mathbb{Z}/5\mathbb{Z})^2$. This implies that the index of G in $J_1(\mathbb{Q})$ is not divisible by 5.

We determine the points $P \in C_1(\mathbb{F}_7)$ such that $\iota(P) \in \rho_7(2J_1(\mathbb{Q})) = 2\rho_7(G)$. We find the set

$$X_7 = \{\rho_7(\infty_+), \rho_7(\infty_-), (-2,2), (-2,-2)\}.$$

Note that for any $P \in J_1(\mathbb{Q})$, we must have $\rho_7(P) \in X_7$ by Lemma 4.

Now we look at $p = 13$. The image of G in $J_1(\mathbb{F}_{13}) \cong \mathbb{Z}/10\mathbb{Z} \oplus \mathbb{Z}/2850\mathbb{Z}$ has index 5. Since we already know that $(J_1(\mathbb{Q}) : G)$ is not a multiple of 5, this implies that $\rho_{13}(G) = \rho_{13}(J_1(\mathbb{Q}))$. As above for $p = 7$, we compute the set $X_{13} \subset C_1(\mathbb{F}_{13})$ of points mapping into $\rho_{13}(2J_1(\mathbb{Q}))$. We find

$$X_{13} = \{\rho_{13}(\infty_+), \rho_{13}(\infty_-)\}\,.$$

Now suppose that there is $P \in C_1(\mathbb{Q})$ with $\rho_7(P) \in \{(-2, 2), (-2, -2)\}$. Then $\iota(P)$ is in one of two specific cosets in $J_1(\mathbb{Q})/\ker\rho_7 \cong G/\ker\rho_7|_G$. On the other hand, we have $\rho_{13}(P) = \rho_{13}(\infty_\pm)$, so that $\iota(P)$ is in one of two specific cosets in $J_1(\mathbb{Q})/\ker\rho_{13} \cong G/\ker\rho_{13}|_G$. If we identify $G = \langle Q_1, Q_2\rangle$ with $\mathbb{Z}^2$, then we can find the kernels of ρ_7 and of ρ_{13} on G explicitly, and we can also determine the relevant cosets explicitly. It can then be checked that the union of the first two cosets does not meet the union of the second two cosets. This implies that such a point P cannot exist. Therefore, the only remaining possibilities are that $\rho_7(P) = \rho_7(\infty_\pm)$. □

Remark 8. The use of information at $p = 13$ to rule out residue classes at $p = 7$ in the proof above is a very simple instance of a method known as the *Mordell-Weil sieve*. For a detailed description of this method, see [4].

Now we need to find the space of holomorphic 1-forms on C_1, defined over $\mathbb{Q}_7$, that annihilate the Mordell-Weil group under the integration pairing, compare Sect. 3.4. We follow the procedure described in [14]. We first find two independent points in the intersection of $J_1(\mathbb{Q})$ and the kernel of reduction mod 7. In our case, we take $R_1 = 20Q_1$ and $R_2 = 5Q_1 + 60Q_2$. We represent these points in the form $R_j = [D_j - 4\infty_-]$ with effective divisors D_1, D_2 of degree 4. The coefficients of the primitive polynomial in $\mathbb{Z}[x]$ whose roots are the x-coordinates of the points in the support of D_1 have more than 100 digits and those of the corresponding polynomial for D_2 fill several pages, so we refrain from printing them here. (This indicates that it is a good idea to work with a small prime!) The points in the support of D_1 and D_2 all reduce to ∞_- modulo the prime above 7 in their fields of definition (which are degree 4 number fields totally ramified at 7). Expressing a basis of $\Omega^1_{C_1}(\mathbb{Q}_7)$ as power series in the uniformiser $t = 1/x$ at $P_0 = \infty_-$ times dt, we compute the integrals numerically. More precisely, the differentials

$$\eta_0 = \frac{dx}{2y}, \quad \eta_1 = \frac{x\,dx}{2y}, \quad \eta_2 = \frac{x^2\,dx}{2y} \quad \text{and} \quad \eta_3 = \frac{x^3\,dx}{2y}$$

form a basis of $\Omega^1_{C_1}(\mathbb{Q}_7)$. We get

$$\eta_j = t^{3-j}\Big(\frac{1}{2} - \frac{15}{2}t + 115t^2 - 1980t^3 + \frac{145385}{4}t^4 - \frac{2764899}{4}t^5 + \ldots\Big)\,dt$$

as power series in the uniformiser. Using these power series up to a precision of t^{20}, we compute the following 7-adic approximations to the integrals.

$$\left(\int_0^{R_j} \eta_i\right)_{0\le i\le 3, 1\le j\le 2} = \begin{pmatrix} -20\cdot 7 + O(7^4) & -155\cdot 7 + O(7^4) \\ -150\cdot 7 + O(7^4) & -13\cdot 7 + O(7^4) \\ -130\cdot 7 + O(7^4) & -83\cdot 7 + O(7^4) \\ -19\cdot 7 + O(7^4) & 163\cdot 7 + O(7^4) \end{pmatrix}$$

From this, it follows easily that the reductions mod 7 of the (suitably scaled) differentials that kill $J_1(\mathbb{Q})$ fill the subspace of $\Omega^1_{C_1}(\mathbb{F}_7)$ spanned by

$$\omega_1 = (1 + 3x - 2x^2)\frac{dx}{2y} \quad \text{and} \quad \omega_2 = (1 - x^2 + x^3)\frac{dx}{2y}\,.$$

Since ω_2 does not vanish at the points $\rho_7(\infty_\pm)$, this implies that there can be at most one rational point P on C_1 with $\rho_7(P) = \rho_7(\infty_+)$ and at most one point P with $\rho_7(P) = \rho_7(\infty_-)$ (see for example [15, Prop. 6.3]).

Proposition 9. *The only rational points on C_1 are ∞_+ and ∞_-.*

Proof. Let $P \in C_1(\mathbb{Q})$. By Lemma 7, $\rho_7(P) = \rho_7(\infty_\pm)$. By the argument above, for each sign $s \in \{+,-\}$, we have $\#\{P \in C_1(\mathbb{Q}) : \rho_7(P) = \rho_7(\infty_s)\} \le 1$. These two facts together imply that $\#C_1(\mathbb{Q}) \le 2$. Since we know the two rational points ∞_+ and ∞_- on C_1, there cannot be any further rational points. □

We can now prove Thm. 1.

Proof (of Thm. 1). The considerations in Sect. 2 imply that if (a^2, b^2, c^2, d^5) is an arithmetic progression in coprime integers, then there are coprime u and v, related to a, b, c, d by (3), such that $(u/v, a/v^5)$ is a rational point on one of the curves C_j with $-2 \le j \le 2$. By Prop. 3, there are no rational points on C_0 and C_2 and therefore also not on the curve C_{-2}, which is isomorphic to C_2. By Prop. 9, the only rational points on C_1 (and C_{-1}) are the points at infinity. This translates into $a = \pm 1$, $u = \pm 1$, $v = 0$, and we have $j = \pm 1$. We deduce $a^2 = 1$, $b^2 = g_1(\pm 1, 0)^2 = 1$, whence also $c^2 = d^5 = 1$. □

References

1. Bosma, W., Cannon, J., Playoust, C.: The Magma Algebra System I: The User Language. J. Symb. Comp. 24, 235–265 (1997), http://magma.maths.usyd.edu.au/magma
2. Bruin, N., Győry, K., Hajdu, L., Tengely, S.: Arithmetic progressions consisting of unlike powers. Indag. Math. 17, 539–555 (2006)
3. Bruin, N., Stoll, M.: 2-cover descent on hyperelliptic curves. Math. Comp. 78, 2347–2370 (2009)
4. Bruin, N., Stoll, M.: The Mordell-Weil sieve: Proving non-existence of rational points on curves. LMS J. Comput. Math. (to appear), arXiv:0906.1934v2 [math.NT]
5. Cantor, D.G.: Computing in the Jacobian of a hyperelliptic curve. Math. Comp. 48, 95–101 (1987)
6. Chabauty, C.: Sur les points rationnels des courbes algébriques de genre supérieur à l'unité. C. R. Acad. Sci. Paris 212, 882–885 (1941) (French)

7. Coleman, R.F.: Effective Chabauty. Duke Math. J. 52, 765–770 (1985)
8. Darmon, H., Merel, L.: Winding quotients and some variants of Fermat's last theorem. J. Reine Angew. Math. 490, 81–100 (1997)
9. Dickson, L.E.: History of the theory of numbers. Vol. II: Diophantine Analysis. Chelsea Publishing Co., New York (1966)
10. Evertse, J.-H., Tijdeman, R.: Some open problems about Diophantine equations from a workshop in Leiden in (May 2007), http://www.math.leidenuniv.nl/~evertse/07-workshop-problems.pdf
11. Hajdu, L.: Perfect powers in arithmetic progression. A note on the inhomogeneous case. Acta Arith. 113, 343–349 (2004)
12. Hajdu, L., Tengely, S.: Arithmetic progressions of squares, cubes and n-th powers. Funct. Approx. Comment. Math. 41, 129–138 (2009)
13. Poonen, B., Schaefer, E.F.: Explicit descent for Jacobians of cyclic covers of the projective line. J. Reine Angew. Math. 488, 141–188 (1997)
14. Stoll, M.: Rational 6-cycles under iteration of quadratic polynomials. LMS J. Comput. Math. 11, 367–380 (2008)
15. Stoll, M.: Independence of rational points on twists of a given curve. Compositio Math. 142, 1201–1214 (2006)
16. Stoll, M.: Implementing 2-descent for Jacobians of hyperelliptic curves. Acta Arith. 98, 245–277 (2001)

Sieving for Pseudosquares and Pseudocubes in Parallel Using Doubly-Focused Enumeration and Wheel Datastructures*

Jonathan P. Sorenson

Butler University, Indianapolis IN 46208, USA
sorenson@butler.edu
http://www.butler.edu/~sorenson

Abstract. We extend the known tables of pseudosquares and pseudocubes, discuss the implications of these new data on the conjectured distribution of pseudosquares and pseudocubes, and present the details of the algorithm used to do this work. Our algorithm is based on the space-saving wheel data structure combined with doubly-focused enumeration, run in parallel on a cluster supercomputer.

1 Introduction

It is well-known that testing for primality can be done in polynomial time [1,3]. However, the fastest known deterministic algorithms are conjectured to be the pseudosquares prime test of Lukes, Patterson, and Williams [6], and its generalizations, the pseudocube prime test of Berrizbeitia, Müller, and Williams [4], and the Eisenstein pseudocube test [13,15], all of which run in roughly cubic time, if a sufficiently large pseudosquare or pseudocube is available. In particular, the pseudosquares prime test is very useful in the context of finding all primes in an interval [10], where sieving can be used in place of trial division. This, then, motivates our search for larger and larger peudosquares and pseudocubes, and our attempts to predict their distribution. See, for example, Wooding and Williams [14] and also [7,12,8,2,11].

In this paper, we present extensions to the known tables of pseudosquares and pseudocubes in §2. We discuss the implications of this new data on the conjectured distribution of pseudosquares and pseudocubes in §3, and give a minor refinement of the current conjectures. Then we describe our parallel algorithm, based on Bernstein's doubly-focused enumeration [2], which is used in a way similar, but not identical to the work of Wooding and Williams [14], combined with the space-saving wheel data structure presented in [10, §4.1]. We then suggest ideas for future work in §5.

* Supported by a grant from the Holcomb Awards Committe, and computing resources provided by the Frank Levinson Supercomputing Center at Butler University.

G. Hanrot, F. Morain, and E. Thomé (Eds.): ANTS-IX 2010, LNCS 6197, pp. 331–339, 2010.

2 Computational Results

Let (x/y) denote the Legendre symbol [5]. For an odd prime p, let $L_{p,2}$, the *pseudosquare* for p, be the smallest positive integer such that

1. $L_{p,2} \equiv 1 \pmod 8$,
2. $(L_{p,2}/q) = 1$ for every odd prime $q \leq p$, and
3. $L_{p,2}$ is not a perfect square.

In other words, $L_{p,2}$ is a square modulo all primes up to p, but is not a square. We found the following new pseudosquares:

p	$L_{p,2}$
367	36553 34429 47705 74600 46489
373	42350 25223 08059 75035 19329
379	$> 10^{25}$

The two pseudosquares listed were found in 2008 in a computation that went up to 5×10^{24}, taking roughly 3 months wall time. The final computation leading to the lower bound of 10^{25} ran for about 6 months, in two 3-month pieces, the second of which finished on January 1st, 2010.

Wooding and Williams [14] had found a lower bound of $L_{367,2} > 120120 \times 2^{64} \approx 2.216 \times 10^{24}$. (Note: a complete table of pseudosquares, current as of this writing, is available at `http://cr.yp.to/focus.html` care of Dan Bernstein).

Note that 10^{25} may be used as a lower bound for $L_{379,2}$ in the pseudosquares prime test. Together with trial division to guarantee there are no divisors below, say, 10^{10}, this means the pseudosquares prime test is practical on integers of 35 decimal digits, especially in the context of a prime sieve [10].

Similarly, for an odd prime p, let $L_{p,3}$, the *pseudocube* for p, be the smallest positive integer such that

1. $L_{p,3} \equiv \pm 1 \pmod 9$,
2. $L_{p,3}^{(q-1)/3} \equiv 1 \pmod q$ for every prime $q \leq p$, $q \equiv 1 \pmod 3$,
3. $\gcd(L_{p,3}, q) = 1$ for every prime $q \leq p$, and
4. $L_{p,3}$ is not a perfect cube.

We found the following new pseudocubes (only listed for $p \equiv 1 \pmod 3$):

p	$L_{p,3}$
499	601 25695 21674 16551 89317
523,541	1166 14853 91487 02789 15947
547	41391 50561 50994 78852 27899
571,577	1 62485 73199 87995 69143 39717
601,607	2 41913 74719 36148 42758 90677
613	67 44415 80981 24912 90374 06633
619	$> 10^{27}$

These pseudocubes were found in about 6 months of total wall time in 2009. Wooding and Williams [14] had found a lower bound of $L_{499,3} > 1.45152 \times 10^{22}$. For a complete list of known pseudocubes, see [14,4,11].

3 The Distribution of Pseudosquares and Pseudocubes

Let p_i denote the ith prime, and q_i denote the ith prime such that $q_i \equiv 1 \pmod 3$. In [6] it was conjectured that, for a constant $c_2 > 0$, we have

$$L_{p_n,2} \approx c_2 2^n \log p_n. \tag{1}$$

Using similar methods, in [4] it was conjectured that, for a constant $c_3 > 0$, we have

$$L_{q_n,3} \approx c_3 3^n (\log q_n)^2. \tag{2}$$

In a desire to test the accuracy of these conjectures, for integers $n > 0$ let us define

$$c_2(n) := \frac{L_{p_n,2}}{2^n \log p_n}, \tag{3}$$

$$c_3(n) := \frac{L_{q_n,3}}{3^n (\log q_n)^2}. \tag{4}$$

We calculated $c_2(n)$ and $c_3(n)$ from known pseudosquares and pseudocubes. We present these computations in Table 1, for pseudosquares, and in Table 2, for pseudocubes, below.

From Table 1, we readily see that $c_2(n)$ appears to be bounded between roughly 5 and 162, with an average value near 45. There is no clear trend toward zero or infinity. Due to the common occurence of values of n where $L_{p_n,2} = L_{p_{n+1},2}$ (for example, $n = 56$), it should also be clear $c_2(n)$ does not have a limit.

Similarly for the pseudocubes, in Table 2 we see that $0.05 < c_3(n) < 6.5$ for $10 \le n \le 53$, with an average value of roughly 1.22. And again, there is no clear trend toward zero or infinity, nor can there be a limit for $c_3(n)$.

This leads us to the following refinements, if you will, of the conjectures (1),(2) above.

Conjecture. For the pseudosquares, we conjecture that

$$\liminf_{n\to\infty} \frac{L_{p_n,2}}{2^n \log p_n} > 0, \tag{5}$$

$$\limsup_{n\to\infty} \frac{L_{p_n,2}}{2^n \log p_n} < \infty. \tag{6}$$

Similarly, for the pseudocubes, we conjecture that

$$\liminf_{n\to\infty} \frac{L_{q_n,3}}{3^n (\log q_n)^2} > 0, \tag{7}$$

$$\limsup_{n\to\infty} \frac{L_{q_n,3}}{3^n (\log q_n)^2} < \infty. \tag{8}$$

Table 1. Values of $c_2(n)$ based on known pseudosquares

n	p_n	$L_{p_n,2}$	$c_2(n)$
2	3	73	16.61
3	5	241	18.72
4	7	1009	32.41
5	11	2641	34.42
6	13	8089	49.28
7	17	18001	49.64
8	19	53881	71.48
9	23	87481	54.49
10	29	117049	33.95
11	31	515761	73.34
12	37	1083289	73.24
13	41	3206641	105.41
14	43	3818929	61.97
15	47	9257329	73.38
16	53	22000801	84.55
17	59	48473881	90.70
18	61	48473881	44.98
19	67	175244281	79.49
20	71	427733329	95.70
21	73	427733329	47.54
22	79	898716289	49.04
23	83	2805544681	75.69
24	89	2805544681	37.25
25	97	2805544681	18.28
26	101	10310263441	33.29
27	103	23616331489	37.96
28	107	85157610409	67.89
29	109	85157610409	33.81
30	113	196265095009	38.67
31	127	196265095009	18.87
32	131	2871842842801	137.15
33	137	2871842842801	67.95
34	139	2871842842801	33.88
35	149	26250887023729	152.68
36	151	26250887023729	76.14
37	157	112434732901969	161.79
38	163	112434732901969	80.30
39	167	112434732901969	39.96
40	173	178936222537081	31.58
41	179	178936222537081	15.69
42	181	696161110209049	30.45
43	191	696161110209049	15.07
44	193	2854909648103881	30.84
45	197	6450045516630769	34.70
46	199	6450045516630769	17.32
47	211	11641399247947921	15.46
48	223	11641399247947921	7.65
49	227	190621428905186449	62.42
50	229	196640148121928601	32.14
51	233	712624335095093521	58.06
52	239	1773855791877850321	71.92
53	241	2327687064124474441	47.12
54	251	6384991873059836689	64.15
55	257	8019204661305419761	40.11
56	263	10198100582046287689	25.40
57	269	10198100582046287689	12.65
58	271	10198100582046287689	6.32
59	277	69848288320900186969	21.54
60	281	208936365799044975961	32.14
61	283	533552663339828203681	40.99
62	293	936664079266714697089	35.76
63	307	936664079266714697089	17.73
64	311	2142202860370269916129	20.23
65	313	2142202860370269916129	10.10
66	317	2142202860370269916129	5.04
67	331	13649154491558298803281	15.94
68	337	34594858801670127778801	20.14
69	347	9949294593047921334049	28.81
70	349	9949294593047921334049	14.39
71	353	295363187400900310880401	21.32
72	359	295363187400900310880401	10.63
73	367	365533442947705746004 6489	65.54
74	373	423502522308059750351 9329	37.86

Table 2. Values of $c_3(n)$ based on known pseudocubes

n	q_n	$L_{q_n,3}$	$c_3(n)$
10	79	7235857	6.42
11	97	8721539	2.35
12	103	8721539	0.764
13	109	91246121	2.6
14	127	91246121	0.813
15	139	98018803	0.281
16	151	1612383137	1.49
17	157	1612383137	0.488
18	163	7991083927	0.795
19	181	7991083927	0.254
20	193	7991083927	0.0827
21	199	20365764119	0.0695
22	211	2515598768717	2.8
23	223	6440555721601	2.34
24	229	29135874901141	3.49
25	241	29135874901141	1.14
26	271	29135874901141	0.365
27	277	406540676672677	1.69
28	283	406540676672677	0.558
29	307	406540676672677	0.181
30	313	406540676672677	0.0598
31	331	75017625272879381	3.61
32	337	75017625272879381	1.2
33	349	75017625272879381	0.394
34	367	996438651365898469	1.71
35	373	2152984914389968651	1.23
36	379	12403284862819956587	2.34
37	397	37605274105479228611	2.33
38	409	37605274105479228611	0.77
39	421	37605274105479228611	0.254
40	433	205830039006337114403	0.459
41	439	1845193818928603436441	1.37
42	457	7854338425385225902393	1.91
43	463	12904554928068268848739	1.04
44	487	13384809548521227517303	0.355
45	499	60125695216741655189317	0.527
46	523	116614853914870278915947	0.336
47	541	116614853914870278915947	0.111
48	547	4139150561509947885227899	1.31
49	571	16248573199879956914339717	1.69
50	577	16248573199879956914339717	0.56
51	601	24191374719361484275890677	0.274
52	607	24191374719361484275890677	0.0912
53	613	67444158098124912903740663 3	0.845

It has been pointed out, both by one of the referees and by Rich Schroeppel [9], that a value for $k > 0$ such that

$$L_{p_n,2} = L_{p_{n+1},2} = \cdots = L_{p_{n+k},2}$$

likely is not bounded. This applies to pseudocubes as well. It implies that we, most likely, cannot simultaneously have both (5) and (6), nor both of (7) and (8). This might be avoided if we, say, multiply our upper bounds by n and divide our lower bounds by n in our conjectures.

Our data also has implications on the relative efficiently of primality testing. In particular, several researchers have pointed out that if conjectures (1),(2) are true, then the running time of the pseudocube prime test, which depends on the value of $L^{2/3}_{q_n,3}$, should eventually outperform the pseudosquare prime test, whose running time depends on $L_{p_n,2}$. In particular, one infers from conjectures (1) and (2) that

$$\frac{L^{2/3}_{q_n,3}}{L_{p_n,2}} \quad \gg \quad \left(\frac{3^{2/3}}{2}\right)^n \quad > \quad 1 \tag{9}$$

for sufficiently large n (see [14, §9.1]). This inference follows from our refined conjectures as well.

We have our first specific value of n to support (9), namely with $n = 48$, where $L^{2/3}_{q_n,3} \approx 2.214 \cdot L_{p_n,2}$. However, given that $c_2(n)$ averages about 45, and $c_3(n)$ averages just over 1.2, we would reasonably expect (9) to largely be true only for n larger than about 75, under the assumption these averages are maintained. To test this, more pseudosquares and, in particular, more pseudocubes are needed.

4 Algorithm Details

We begin with a review of doubly-focused enumeration, explain how we employ parallelism, and how the space-saving wheel datastructure is utilized. We also discuss the details of our implementation, including the hardware platform and software used.

4.1 Doubly-Focused Enumeration

The main idea is that every integer x, with $0 \le x \le H$, can be written in the form

$$x = t_p M_n - t_n M_p \tag{10}$$

where

$$\gcd(M_p, M_n) = 1, \quad 0 \le t_p \le \frac{H + M_n M_p}{M_n}, \quad \text{and} \quad 0 \le t_n < M_n. \tag{11}$$

(See [2] or [14, Lemma 1].) This is an explicit version of the Chinese Remainder Theorem.

To find pseudosquares, we set M_n and M_p to be products of small odd primes and 8, choose t_p to be square modulo M_p, and $-t_n$ to be square modulo M_n. To be precise, in our implementation we set

$$\begin{aligned} M_p &= 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 53 \cdot 89 \\ &= 2057\,04617\,33829\,17717 \qquad \text{and} \\ M_n &= 8 \cdot 3 \cdot 5 \cdot 47 \cdot 59 \cdot 61 \cdot 67 \cdot 71 \cdot 73 \cdot 79 \cdot 83 \cdot 97 \\ &= 4483\,25952\,77215\,26840. \end{aligned}$$

Note that both $M_p, M_n < 2^{64}$, allowing us to work in 64-bit machine arithmetic.

To find pseudocubes, the same idea applies, only note that if $-t_n$ is a cube modulo M_n, so is t_n. We used only $2, 9$ and primes congruent to 1 (mod 3) for better filter rates:

$$\begin{aligned} M_p &= 2 \cdot 7 \cdot 13 \cdot 31 \cdot 43 \cdot 73 \cdot 79 \cdot 127 \cdot 139 \cdot 157 \cdot 181 \\ &= 701\,85635\,61110\,39402 \qquad \text{and} \\ M_n &= 9 \cdot 19 \cdot 37 \cdot 61 \cdot 67 \cdot 97 \cdot 103 \cdot 109 \cdot 151 \cdot 163 \\ &= 693\,11050\,43291\,92503 \end{aligned}$$

4.2 Parallelism and Main Loop

Each processor core was assigned an interval of t_p values to process by giving it values of H^- and H^+.

For finding pseudosquares, $H^+ - H^- \approx M_n \cdot 4.76 \times 10^{11}$. For finding pseudocubes, $H^+ - H^- \approx M_n \cdot 4.99 \times 10^{12}$.

Parallelism was achieved by having different processors working on different intervals simultaneously. Once all processors had finished their current intervals, the work was saved to disk (allowing restarts as needed) and new intervals were assigned.

To process an interval, each processor core did the following:

1. Using the wheel datastructure, generate all square or cube values of t_p with $H^- \le t_p M_n \le H^+$, and store these in an array `A[]`.
2. The wheel datastructure does not generate the t_p values in order, so sort `A[]` in memory using quicksort. Note that H^- and H^+ are chosen close enough together so that this array held no more than 40 million integers, using at most 320 megabytes of RAM per processor core.
3. Using the first and last entries in `A[]`, compute a range of valid t_n values to process, and then use a wheel datastructure to generate all t_n values in that range such that $-t_n$ is square modulo M_n for pseudosquares, or t_n is a cube modulo M_n for pseudocubes.

   We use an outer loop over t_n values in the order enumerated by the wheel data structure for M_n, and an inner loop over consecutive t_p values drawn from `A[]`.

4. For each t_n generated, we normalize sieve tables for the next 4 primes (101, 103, 107, 109 for pseudosquares, and 193, 199, 211, 223 for pseudocubes) to allow for constant-time table lookup to see if an x-value (see below) is a square/cube modulo these primes, indexed by t_p value.
   The number of primes to use for this depends on how many t_p values will be processed for each t_n – in our case, it was several hundred on average, so this step improves performance. If it were fewer, say 50, then normalizing the sieve tables would require more work than is saved by having constant-time lookup.
5. For each t_n generated, using binary search on `A[]` to find all the t_p values it can match with, generate an $x = t_p M_n - t_n M_p$ within our global search range. (For example, in our last run for pseudosquares, we searched for x values between 7.5×10^{24} and 10^{25}.)
   Note: at this point we do not actually compute the value of x.
6. Lookup each t_p value in the normalized tables mentioned above. If it fails any of the 4 sieve tests, move on to the next t_p value. For pseudosquares, a t_p values passes these tests with probability roughly $(1/2)^4 = 1/16$, and for pseudocubes, roughly $(1/3)^4 = 1/81$.
   Note that this step is the running time bottleneck of the algorithm.
7. The next batch of primes q have precomputed sieve tables that are not normalized, but we precompute M_p and M_n modulo each q so the we can compute x mod q without exceeding 64-bit arithmetic. Continue only if our t_p value passes all these sieve tests as well. The expected number of primes q used in this step is constant.
8. Finally, compute x using 128-bit hardware arithmetic, and see if it is a perfect square or perfect cube. If it passes this test, append x to the output file for this processor core.

We had two wheel datastructures, one each for M_p and M_n. For details on how this datastructure works, see [10]. We leave the details for how to modify the datastructure to handle cubes in place of squares to the reader.

4.3 Implementation Details

To compute the tables presented in §2, we used Butler University's cluster supercomputer, *BigDawg*, which has 24 compute nodes, each of which has four AMD Opteron 8354 quad-core CPUs at 2.2GHz with 512KB cache, for a total of 384 compute cores. As might be expected, we did not have sole access to this machine for over a year, so the code was designed, and ran, using anywhere from 10 to 24 nodes, or from 160 to 384 cores, depending on the needs of other users. This flexibility is one advantage of our parallelization method – by t_p intervals. In [14], they parallelized over residue classes, which restricts the CPU count to a fixed number (180 in their case).

BigDawg runs a Linux kernel on its head node and compute nodes, and the code was written in C++ using the gnu compiler (version 4.1.2) with MPI. It has both 10GB ethernet and Infiniband interconnect, but inter-processor communication was not a bottleneck for our programs.

We tested our code by first finding known pseudosquares (all but the highest few) and known pseudocubes, in the process verifying previous results.

5 Future Work

We plan to port our code to work with 8 NVidia GPUs recently added to Butler's supercomputer, giving it roughly 2-3 times the raw computing power. This will require a major restructuring of the code, and the removal of recursion in the wheel datastructure.

References

1. Agrawal, M., Kayal, N., Saxena, N.: PRIMES is in P. Ann. of Math. 160(2), 781–793 (2004), http://dx.doi.org/10.4007/annals.2004.160.781
2. Bernstein, D.J.: Doubly focused enumeration of locally square polynomial values. In: High Primes and Misdemeanours: Lectures in Honour of the 60th Birthday of Hugh Cowie Williams, Fields Inst. Commun., vol. 41, pp. 69–76. Amer. Math. Soc., Providence (2004)
3. Bernstein, D.J.: Proving primality in essentially quartic random time. Math. Comp. 76(257), 389–403 (2007), http://dx.doi.org/10.1090/S0025-5718-06-01786-8 (electronic)
4. Berrizbeitia, P., Müller, S., Williams, H.C.: Pseudocubes and primality testing. In: Buell, D.A. (ed.) ANTS 2004. LNCS, vol. 3076, pp. 102–116. Springer, Heidelberg (2004)
5. Hardy, G.H., Wright, E.M.: An Introduction to the Theory of Numbers, 5th edn. Oxford University Press, Oxford (1979)
6. Lukes, R.F., Patterson, C.D., Williams, H.C.: Some results on pseudosquares. Math. Comp. 65(213), 361–372, S25–S27 (1996)
7. Pomerance, C., Shparlinski, I.E.: On pseudosquares and pseudopowers. In: Combinatorial Number Theory, pp. 171–184. Walter de Gruyter, Berlin (2009)
8. Schinzel, A.: On pseudosquares. New Trends in Prob. and Stat. 4, 213–220 (1997)
9. Schroeppel, R.: Private communication (February 2010)
10. Sorenson, J.P.: The pseudosquares prime sieve. In: Hess, F., Pauli, S., Pohst, M. (eds.) ANTS 2006. LNCS, vol. 4076, pp. 193–207. Springer, Heidelberg (2006)
11. Stephens, A.J., Williams, H.C.: An open architecture number sieve. In: Number theory and cryptography (Sydney, 1989). London Math. Soc. Lecture Note Ser., vol. 154, pp. 38–75. Cambridge Univ. Press, Cambridge (1990)
12. Williams, H.C.: Édouard Lucas and primality testing. Canadian Mathematical Society Series of Monographs and Advanced Texts, vol. 22. John Wiley & Sons Inc, New York (1998), A Wiley-Interscience Publication
13. Wooding, K.: The Sieve Problem in One- and Two-Dimensions. Ph.D. thesis, The University of Calgary, Calgary, AB (April 2010) http://math.ucalgary.ca/~hwilliam/files/wooding10thesis.pdf
14. Wooding, K., Williams, H.C.: Doubly-focused enumeration of pseudosquares and pseudocubes. In: Hess, F., Pauli, S., Pohst, M. (eds.) ANTS 2006. LNCS, vol. 4076, pp. 208–221. Springer, Heidelberg (2006)
15. Wooding, K., Williams, H.C.: Improved primality proving with Eisenstein pseudocubes. In: Hanrot, G., Morain, F., Thomé, E. (eds.) ANTS-IX. LNCS, vol. 6197, pp. 372–384. Springer, Heidelberg (2010)

On the Extremality of an 80-Dimensional Lattice

Damien Stehlé[1,2] and Mark Watkins[2]

[1] CNRS and Macquarie University
[2] Magma Computer Algebra Group, School of Mathematics and Statistics, University of Sydney, NSW 2006, Australia
damien.stehle@gmail.com, watkins@maths.usyd.edu.au

Abstract. We show that a specific even unimodular lattice of dimension 80, first investigated by Schulze-Pillot and others, is extremal (i.e., the minimal nonzero norm is 8). This is the third known extremal lattice in this dimension. The known part of its automorphism group is isomorphic to $\mathbf{SL}_2(\mathbf{F}_{79})$, which is smaller (in cardinality) than the two previous examples. The technique to show extremality involves using the positivity of the Θ-series, along with fast vector enumeration techniques including pruning, while also using the automorphisms of the lattice.

1 Introduction

We show that a specific 80-dimensional even unimodular lattice is extremal, that is, that it has no (nonzero) vectors of norm less than 8. It follows that the kissing number of this lattice is 1 250 172 000.[1] Although two other even unimodular extremal lattices in dimension 80 are known [3], the one we describe has a construction related to coding theory, and has an automorphism group that contains $\mathbf{SL}_2(\mathbf{F}_{79})$.

In Section 2 we recall some facts and results about extremal lattices.

In Section 3 we follow the method of Schulze-Pillot [40] to construct our lattice $\mathbf{N}_{80}$ as a 2-neighbour of a lattice derived from a length 80 extended quadratic residue code over $\mathbf{F}_{19}$. The prime 19 here is not overly significant; the construction produces five unimodular lattices in correspondence with the class group of $\mathbf{Q}(\sqrt{-79})$, and the ideal class that yields $\mathbf{N}_{80}$ (the only extremal one among the five) has an ideal of norm 19 in it.[2] Alternatively, a variation (see [1]) on a method of Gross [18, §11] can be used to construct $\mathbf{N}_{80}$, and deals more directly with the ideals of this imaginary quadratic field. Via either method, it is fairly immediate that $\mathbf{N}_{80}$ has an automorphism group that contains $\mathbf{SL}_2(\mathbf{F}_{79})$.

In Section 4 we note that various choices of bases make the group action nice (doubly transitive as signed permutations on the coordinates), and then make a specific basis choice that relates directly to the construction in [1].

[1] We do not describe herein any features of these minimal vectors. In fact, the 2 555 orbits of these vectors under the known automorphisms were first found (without proof of completeness) by the authors of [1], with whom we started this project.

[2] We could also have chosen $l = 5$ (as indicated in [40, Example 3]), but for technical reasons (in lattice generation) wanted l not to be too small.

G. Hanrot, F. Morain, and E. Thomé (Eds.): ANTS-IX 2010, LNCS 6197, pp. 340–356, 2010.

In Section 5 we first briefly outline our method of proof that the lattice $\mathbf{N}_{80}$ is extremal. We need to show that $\mathbf{N}_{80}$ has no nonzero vectors of norm 6 or smaller. We can almost immediately eliminate vectors of norm 2, while a slightly more involved argument is necessary to show there are no vectors of norm 4. We then use the nonnegativity of the coefficients of the Θ-series of the lattice to reduce the problem of showing that there is no vector of norm 6 to the problem of finding (almost) all the vectors of norm 10. The latter is feasible due to the fact that we need only find one representative in each orbit class under the known automorphisms, whereas the more direct method of an exhaustive search for norm 6 vectors would be significantly more time-consuming. After first cataloguing the norm 10 orbits that have a nontrivial stabiliser, all the other vectors will have a full orbit under the known automorphisms, and so we can reduce the problem by a factor of approximately $\#\mathbf{SL}_2(\mathbf{F}_{79}) = 492\,960$. This leaves us with only 15.3 million orbits of norm 10 to find.

In Section 6 we describe our method to find all the norm 10 orbits. One principal idea is to prune the tree corresponding to the Kannan-Fincke-Pohst enumeration algorithm that finds all short lattice vectors [21,12]. Our tree pruning strategy, which generalizes that of [38, §7] and improves the one from [39], considers a truncated search domain that is much smaller but still finds a significant proportion of the desired vectors. Note that the pruning strategy we describe and its analysis have been independently discovered by Gama, Nguyen, and Regev [15, §4]. In our case, we need only find one vector in each orbit class, so the fact we miss some vectors when searching is unimportant. Another idea to speed the search is to periodically apply a random perturbation to the basis and re-apply lattice reduction (namely LLL with deep insertions [38]), before again searching with tree pruning. As our lattices are of quite high dimension, the new basis is very likely to be different than the previous ones. This can help in two ways: firstly, searching with a given lattice basis for short vectors, even with pruning available, tends to become less cost-effective over time, in terms of the number of vectors found per second; and secondly, and rather surprisingly to us, a "good basis" for searching can sometimes have many orbit classes which will not show up until quite deep in the search. We still do not understand this latter phenomenon, but it is easily overcome via the random perturbations.

Section 7 gives our results and verification methods, plus related questions.

Computations. All timings are given for 2.3Ghz Opteron 8356 processors. If otherwise unspecified, only one processor is used.

2 Extremal Lattices

The extremality of a lattice is typically defined using Θ-series, as for instance in [7, §7.4].[3] In particular, an extremal unimodular even lattice in dimension d with $8|d$ has a minimum nonzero vector norm of $2(1+\lfloor d/24 \rfloor)$, as this is twice the

[3] The precise notion of "extremal" seems to vary over time; for instance [6] is more demanding, asking that the minimum be at least $1 + \lfloor d/8 \rfloor$.

dimension of the associated space of modular forms. For odd lattices, shadow theory is typically used to obtain satisfactory bounds [8]. A relatively recent survey on extremality appears in [14].

In particular, there were already two extremal even unimodular lattices known in dimension 80, both due to Bachoc and Nebe [3] via a coding theory construction. The first lattice $\mathbf{L}_{80}$ has an automorphism group $2.A_7 \otimes_{\sqrt{-7}} 2.M_{22}.2$ of size $2^{12}\,3^4\,5^2\,7^2\,11 = 4\,470\,681\,600$, and this group is known to be a maximal finite subgroup of $\mathbf{GL}_{80}(\mathbf{Z})$ (see [3, Theorem 3.2]). The second extremal lattice $\mathbf{M}_{80}$ has known automorphisms [3, Lemma 4.11] of order $2^{12}3^4 5^2 = 8\,294\,400$. For comparison, the number of known automorphisms of our lattice is 492 960.

Our lattice $\mathbf{N}_{80}$ is isometric neither to $\mathbf{L}_{80}$ nor $\mathbf{M}_{80}$. The argument for $\mathbf{L}_{80}$ is immediate, as its automorphism group is known to be maximal but 79 does not divide the order. For $\mathbf{M}_{80}$ we can compute the minimal vectors in a few days, and perhaps argue via some property of them versus those for $\mathbf{N}_{80}$. We can also argue via Aschbacher's theorem on maximal subgroups of finite classical groups, and in an appendix, we sketch a proof along these lines, showing that $\mathrm{Aut}(\mathbf{N}_{80})$ is a maximal finite subgroup of $\mathrm{GL}_{80}(\mathbf{Z})$ up to a possible index of 4.

The idea of extremality can also be extended to include other lattices which are isomorphic to their dual(s). In this case, the full space of modular forms is typically replaced by the subspace that is fixed under the Atkin-Lehner involutions [36]. This then relates the question to a simultaneous maximisation of the minimum of a lattice and that of its shadow; see [13] and [32] for instance.

Finally, we note that [28] shows that there are only finitely many extremal lattices, though the most easily computed bound on maximal dimension still seems to be quite high.[4] In the other direction, King [22] classifies all (even) unimodular lattices in dimension 32 with no roots, and finds there to be at least 10^7 such; as the lack of roots implies that the lattices have no vectors of norm 2, it follows that each is extremal. Similarly, Peters [33] shows there are at least 10^{51} extremal lattices in dimension 40.

3 Construction of the Lattice $\mathbf{N}_{80}$

We follow the paper [40] of Schulze-Pillot on quadratic residue codes and cyclotomic lattices, which builds on works from Thompson, Feit [9], and Quebbemann [35, §3] about unimodular lattices with an automorphism of prime order.

[4] The proof therein is similar in flavour to the idea we exploit, that is, for sufficiently large dimension, the first form in a triangular basis will have coefficients that are negative, and thus positivity precludes the existence of an extremal lattice. See the recent [42, p. 36] for a brief sketch. Our computations give that the q^{n+2} term in the expansion is negative for $n \geq 6\,775, 6\,789, 6\,803$ for the respective 0, 8, 16 mod 24 classes, which gives an upper bound of $163\,264 = (6802 \cdot 24) + 16$ for the dimension of an even unimodular extremal lattice. Finally, Rains [37] has followed upon the work of Krasikov and Litsyn [27] to obtain that the minimal norm of a unimodular lattice is (asymptotically with dimension $d \to \infty$) smaller than the Siegel bound $\sim d/12$ by at least a *constant factor* (see $N = 1$ in the Remark after Theorem 4.2 in [37]).

The construction gives a unimodular lattice as a sublattice of index p in a (rescaled) direct sum of two lattices of dimensions 2 and $(p-1)$. In this, the 2-dimensional lattice T_2 can be taken as any integral lattice of determinant p. The lattice U_{p-1} of dimension $(p-1)$ comes about from an (unpublished) construction of Thompson (see [9, §9]). We let $E = \mathbf{Q}(\zeta_p)$ be cyclotomic, and take an ideal $\mathfrak{A} \subseteq \mathcal{O}_E$ such that $\mathfrak{A}\bar{\mathfrak{A}} = (d)$ with $d \in E^+$ totally positive. This ideal induces a (positive definite) lattice of dimension $(p-1)$ via a basis for the ring of integers $\mathbf{Z}[\zeta_p]$, with the quadratic form given by $Q_1(u) = \mathrm{tr}_{\mathbf{Q}}^E(u\bar{u}d^{-1})$. Via a computation (with the different as in [9, Theorem 9.3], or with a Vandermonde determinant) one can show that the lattice U_{p-1} has determinant p^{p-2}.

To obtain a unimodular lattice of dimension $(p+1)$, we start with the direct sum $T_2 \oplus U_{p-1}$, and take the sublattice of this consisting of all vectors whose norm is a multiple of p. Upon dividing the whole lattice by p, the result will be integral and unimodular, the latter since $(p \cdot p^{p-2}) \cdot p^2/p^{p+1} = 1$. We need to show that this actually yields a sublattice, that is, the resulting subset of the original lattice satisfies the group law, and this is most easily done via homomorphic projection maps. We take the lattice

$$\mathbf{N}(T_2, U_{p-1}) = \{(\boldsymbol{m}, \boldsymbol{u}) \in T_2 \oplus U_{p-1} \mid \pi(\boldsymbol{m}) = \rho(\boldsymbol{u})\}$$

under the quadratic form $Q\big((\boldsymbol{m}, \boldsymbol{u})\big) = \big(Q_0(\boldsymbol{m}) + Q_1(\boldsymbol{u})\big)/p$, with the projection maps being $\pi : T_2 \to R/\mathrm{rad}_{Q_0}(R)$ where $R = T_2/pT_2$, and $\tilde{\rho} : \mathfrak{A} \to \mathfrak{A}/(1-\zeta_p)\mathfrak{A}$ (here $\tilde{\rho}$ is on $\mathfrak{A}$, with ρ on U_{p-1}). Since $(1-\zeta_p)$ has norm p, both images will be vector spaces over $\mathbf{F}_p$ of dimension 1, and we can identify them (arbitrarily) by taking $\boldsymbol{m}_0 \in T_2$ and $u_0 \in \mathfrak{A}$ with $Q_0(\boldsymbol{m}_0) \equiv 1 \pmod{p}$ and $u_0\bar{u}_0 d^{-1} \equiv 1 \pmod{(1-\zeta_p)\mathcal{O}_E}$. The lattice $\mathbf{N}(T_2, U_{p-1})$ will be even if and only if T_2 is even.

3.1 An Odd Lattice

Rather than derive our desired even unimodular lattice directly, we again follow Schulze-Pillot, who first constructs an odd lattice for which the automorphism group can be determined via a relation to coding theory, and then passes to an even lattice via Kneser's neighbouring construction.

We let K be the imaginary quadratic field $\mathbf{Q}(\sqrt{-79})$, and $d = l = 19$ an auxiliary prime that splits. Writing $(l)\mathcal{O}_K = \mathfrak{l}\bar{\mathfrak{l}}$, the location of $\mathfrak{l}$ in the class group of K will have a determining factor on the lattice we derive in the end, and so the choice of l is not completely arbitrary. We let $\mathfrak{a}$ be the ideal of K generated by l and the twisted Gauss sum $\frac{1}{2}\big[1 - 33\sum_a \chi_p(a)\zeta_p^a\big]$ where χ_p is the quadratic character modulo p. Using the notation of Schulze-Pillot, we have $p = -j^2 + 8ml$ with $p = 79$, $j = 15$, $m = 2$, and $l = 19$, so that $yj \equiv 1 \pmod{l}$ together with $y \equiv 1 \pmod{4}$ yields $y = 33$.[5] Noting that $\mathfrak{a}\bar{\mathfrak{a}} = (l)$ and taking $E = \mathbf{Q}(\zeta_{79})$, we write $\mathfrak{A} = \mathfrak{a}\mathcal{O}_E$ so that $\mathfrak{A}\bar{\mathfrak{A}} = (19)$ in $\mathcal{O}_E$. Letting T_2 be the 2-dimensional lattice (in a basis $\{\boldsymbol{w}_1, \boldsymbol{w}_2\}$) of determinant 79 given by the

[5] The import of this numerology only becomes clear when proofs are included, as this choice of y for the scaling factor of the Gauss sum allows one to show that the cyclotomic and coding theory constructions agree.

Gram matrix $Q_0 = \begin{pmatrix} l & j \\ j & 8m \end{pmatrix} = \begin{pmatrix} 19 & 15 \\ 15 & 16 \end{pmatrix}$, we fix the gluing via $\pi(\boldsymbol{w}_1) = \rho(\lfloor l\zeta_p \rceil)$, where here $\lfloor\cdot\rceil$ gives the map from $\mathfrak{A}$ to U_{p-1}. We let $\mathbf{N}_o = \mathbf{N}(T_2, U_{p-1})$ with these choices, noting that $\mathbf{N}_o$ is odd.

3.2 Relation to Coding Theory

We can obtain the correspondence with coding theory by taking p coordinates as $\boldsymbol{e}_i = \boldsymbol{w}_1 \oplus \lfloor l\zeta_p^i \rceil$ for $0 \le i \le p-1$ and an additional one $\boldsymbol{e}_\infty = j\boldsymbol{w}_1 - l\boldsymbol{w}_2$, from which a computation shows that these $\boldsymbol{e}_i$ form a scaled root system of type $80A_1$ in $\mathbf{N}_o$, that is, each $\boldsymbol{e}_i$ has the same norm, and they are all mutually orthogonal. Indeed, for all $0 \le i \le p-1$ we have $\|\boldsymbol{e}_i\| = \big[Q_0(\boldsymbol{w}_1) + (p-1)\cdot(l^2/l)\big]/p = l$ since $Q_0(\boldsymbol{w}_1) = l$, while $\|\boldsymbol{e}_\infty\| = Q_0(j\boldsymbol{w}_1 - l\boldsymbol{w}_2)/p = l(8ml - j^2)/p = l$. For the inner products, we have

$$\begin{aligned}
\langle \boldsymbol{e}_i, \boldsymbol{e}_k \rangle &= \|\boldsymbol{e}_i + \boldsymbol{e}_k\| - \|\boldsymbol{e}_i\| - \|\boldsymbol{e}_k\| \\
&= \frac{1}{p}\Big(Q_0(2\boldsymbol{w}_1) + (l^2/l)\cdot \mathrm{tr}_{\mathbf{Q}}^E\big[(\zeta^i + \zeta^k)(\bar{\zeta}^i + \bar{\zeta}^k)\big]\Big) - 2l \\
&= \frac{1}{p}\Big(4l + l\cdot \mathrm{tr}_{\mathbf{Q}}^E\big[2 + \zeta^{i-k} + \zeta^{i+k}\big]\Big) - 2l \\
&= \frac{1}{p}\big(4l + l\cdot[2(p-1) - 1 - 1]\big) - 2l = 0
\end{aligned}$$

when $i \ne k$ and $i, k \ne \infty$, while for $i \ne \infty$ we have

$$\begin{aligned}
\langle \boldsymbol{e}_i, \boldsymbol{e}_\infty \rangle &= \|\boldsymbol{e}_i + \boldsymbol{e}_\infty\| - 2l \\
&= \frac{1}{p}\Big[Q_0\big((j+1)\boldsymbol{w}_1 - l\boldsymbol{w}_2\big) + (p-1)\cdot(l^2/l)\Big] - 2l \\
&= \frac{1}{p}\big[l(1 + 8ml - j^2) + l(p-1)\big] - 2l = 0.
\end{aligned}$$

Using this root system, it follows that the extended quadratic residue code $C \subseteq \mathbf{F}_l^{80}$ (or indeed, any self-dual code) gives an integral unimodular lattice via

$$\mathbf{N}_C = \left\{ \frac{1}{l}\sum_i a_i \boldsymbol{e}_i \;\middle|\; (\bar{a}_i) \in C \right\} \tag{1}$$

where the sum is over all 80 coordinates, and $\bar{a}_i$ is reduction mod l of a_i. The proof that $\mathbf{N}_C$ is the same lattice as our lattice $\mathbf{N}_o$ is given in [40, Proposition 1], using the generator matrix and idempotent of the code.[6] The appearance of the value $y = 33$ with the Gauss sum is of relevance therein.

[6] We have taken a sublattice of index l^{p+1} via the scaled root system, and then taken a superlattice of the same index via the construction from coding theory, and so just have to check that these operations are compatible.

One nicety of this re-visioning is that the code automorphism (of order 4) given by $a_\infty \to a_0$, $a_0 \to -a_\infty$, $a_i \to -\chi_p(i)a_j$, where $ij \equiv -1 \pmod p$, can be seen to lift to the lattice. Combined with the order p automorphism induced via ζ_p, which fixes a_∞ and cycles $a_0 \to a_1 \to \cdots \to a_{p-1} \to a_0$, this gives $\mathbf{SL}_2(\mathbf{F}_p)$ as a subgroup of the automorphism group $\mathrm{Aut}(\mathbf{N}_o)$ of the lattice.

In an appendix, we use the classification of finite simple groups to show that this realisation of $\mathbf{SL}_2(\mathbf{F}_{79})$ is within a factor of 4 of being a maximal finite subgroup of $\mathbf{GL}_{80}(\mathbf{Z})$, so that $[\mathrm{Aut}(\mathbf{N}_o) : \mathbf{SL}_2(\mathbf{F}_{79})] \leq 4$.

3.3 The Even 2-Neighbours

The above lattice $\mathbf{N}_o$ is odd, while we wish to get an even unimodular lattice. The method of passing to this is given by the neighbouring method of Kneser [26]. Again following Schulze-Pillot, we want to find $\boldsymbol{v} \in \mathbf{N}_o$ with $Q(\boldsymbol{v}) \in 4\mathbf{Z}$, and then take the lattice spanned by $\boldsymbol{v}/2$ and the sublattice of $\mathbf{N}_o$ whose inner product with $\boldsymbol{v}$ is even. Via linear algebra over $\mathbf{F}_2$, we find that there is a 2-dimensional space of such $\boldsymbol{v}$ satisfying the conditions (Schulze-Pillot notes this in general via genus theory). Obviously $\boldsymbol{v} = \mathbf{0}$ does not help us, while we also need $Q(\boldsymbol{v}) \in 8\mathbf{Z}$ if the resulting neighbouring lattice is to be even, and this eliminates another of the initial 4 possibilities. This leaves but 2 choices for $\boldsymbol{v}$, one of which gives a lattice with many vectors of norm 4 (note that $\boldsymbol{v}$ itself must have norm at least 32 if the new lattice is to have minimum 8) and the other of which is our desired lattice $\mathbf{N}_{80}$.

As in [40, Proposition 2], we could construct $\mathbf{N}_{80}$ directly using a different choice with T_2 in the cyclotomic construction, though the relation to coding theory then becomes less clear. For instance, [40, Example 3] takes $l = 5$ and $Q_0 = \begin{pmatrix} 8 & 1 \\ 1 & 10 \end{pmatrix}$ to get the same $\mathbf{N}_{80}$. Finally, the last Remark of [40] notes the automorphisms of $\mathbf{N}_o$ given by $\mathbf{SL}_2(\mathbf{F}_p)$ all transfer to $\mathbf{N}_{80}$. As noted above, we show in an appendix that $[\mathrm{Aut}(\mathbf{N}_{80}) : \mathbf{SL}_2(\mathbf{F}_{79})] \leq 4$ so that in particular $\mathbf{N}_{80}$ and $\mathbf{M}_{80}$ are not isometric, but our proof of extremality does not use this.

4 Nice Bases for $\mathbf{N}_{80}$

We next link $\mathbf{N}_{80}$ to the construction given in [1] that modifies the method of Gross. The authors of [1] construct the lattice from a representation that is irreducible away from 2. In particular, in the basis they obtain, all the coordinates are of the same parity. Furthermore, the automorphisms are given by a doubly transitive signed permutation action on the coordinates.

From our construction, we have a lattice $\mathbf{N}_{80}$ with automorphisms generated by two matrices O_{79} and O_4. We wish to transform this so that the automorphisms are generated by signed permutations σ_{79} and σ_4 (as in the end of Section 3.2), thus giving a doubly transitive coordinate action. One way to achieve this is just to solve the 80^2-dimensional linear algebra problem given by

equating the automorphisms, that is, solve $O_{79}X = X\sigma_{79}$ and $O_4X = X\sigma_4$ for the unknown matrix X (we try solving this with both σ_4 and σ_4^3).

It turns out that the resulting solution space is 2-dimensional, and if we write X_1 and X_2 for generators of it, then the determinant of the matrix $\left(X_1 t + X_2 u\right)$ is given by $2^{40} f(t,u)^{40}$ where f is a binary quadratic form of discriminant -79 corresponding to the ideal of above. To obtain the representation of [1] we choose the pair (t,u) so that $f(t,u) = 8$, so that the transform maps vectors of norm 10 in $\mathbf{N}_{80}$ to vectors of norm $16 \cdot 10$ in the resulting sublattice of $\mathbf{Z}^{80}$. The resulting basis has the property that every vector has coordinates all of the same parity. We denote this transform matrix from $\mathbf{N}_{80}$ to $\mathbf{Z}^{80}$ by T_{16}, and the resulting lattice basis by B_{80}.

4.1 Identifying Orbits

As noted above, the action of σ_{79} and σ_4 is doubly transitive, and we can exploit this to expedite the finding of a canonical representative for a given orbit. We first find the largest coordinate in absolute value, and move it to the front, and then cycle the latter 79 coordinates until the second largest is in the second position. This movement uses $80 \cdot 79$ elements of the group, and after modding out by the centre $\{\pm 1\}$, we only have 39 possibilities left to check for their 78 latter coordinates (we use a lexicographic ordering). Of course, we could have many ties amongst the two largest coordinates (this is basis-dependent, and we can map to another choice of (t,u) if desired), but this method will still be much faster than looping over all 492 960 possibilities.

5 Method of Proof

We now describe how we shall show that $\mathbf{N}_{80}$ is indeed extremal. Since the lattice $\mathbf{N}_{80}$ is even and unimodular, its Θ-series Θ_{80} lies in the vector space of modular forms of level 1 and weight 40 (see [30]). This space has dimension 4, and a triangular integral basis is:

$$\begin{aligned}
f_0 &= 1 + 1\,250\,172\,000\,q^4 + 7\,541\,401\,190\,400\,q^5 + O(q^6),\\
f_1 &= q + 19\,291\,168\,q^4 + 37\,956\,369\,150\,q^5 + O(q^6),\\
f_2 &= q^2 + 156\,024\,q^4 + 57\,085\,952\,q^5 + O(q^6),\\
f_3 &= q^3 + 168\,q^4 - 12\,636\,q^5 + O(q^6).
\end{aligned}$$

We thus know that $\Theta_{80} = f_0 + a_1 f_1 + a_2 f_2 + a_3 f_3$ for some integers a_i. We shall derive that $a_1 = a_2 = 0$ by showing that there are no vectors of norm 2 or 4 in the lattice. We will then have

$$\Theta_{80} = 1 + a_3 q^3 + (\cdots) q^4 + (7\,541\,401\,190\,400 - 12\,636\,a_3)\,q^5 + O(q^6).$$

By positivity we have $a_3 \geq 0$, and so by finding 7 541 401 190 400 vectors of norm 10 in the lattice, we deduce that $a_3 = 0$ so that $\mathbf{N}_{80}$ is extremal as claimed.

The reader might wonder why we do not simply search for norm 6 vectors, but instead aim to find all those of norm 10, as the latter (at first glance) seems much harder. However, the search in norm 6 has to be exhaustive, while with norm 10 it need not be: we find one vector in each orbit, and apply automorphisms to get the whole set. We estimate an exhaustive search for norm 6 vectors would take more than 1 000 times as much work as our method using norm 10 vectors.

5.1 The Lattice $\mathbf{N}_{80}$ Has No Vectors of Norm 2 or 4

As we noted above in Section 4, we can change the basis by a transform T_{16} so that each vector has its norm multiplied by 16, with the resulting basis having the property that all the coordinates of any vector will have the same parity. In particular, a vector of norm 2 or 4 will have the square-sum of its coordinates as 32 or 64, with necessarily all coordinates being even. Also, the inner product of any two vectors in this basis will need to be a multiple of 16, a fact we exploit below. Finally, the lattice automorphisms in this new basis are given by signed permutations, with the action doubly transitive.

No vectors of norm 2 (roots). One proof (from Elkies) first notes that the only root systems with compatible automorphisms are A_1^{80} and D_{80}. With the former, any automorphism of order 79 would necessarily fix at least one of the 160 roots, but the 2-dimensional sublattice of $\mathbf{N}_{80}$ fixed by a 79-cycle has no roots. The latter is similarly impossible; a 39-cycle must fix a root since $\gcd(39, 12\,640) = 1$, but the 4-dimensional sublattice therein lacks roots.

Another way (similar to a comment in [40, Example 3]) would be to use $l = 5$ and note that we must have $\sum_i a_i^2 = 2l = 10$ in (1), while the minimal distance[7] of the extended quadratic residue code of length 80 over $\mathbf{F}_5$ is > 10, though care needs to be made here when working with both $\mathbf{N}_{80}$ and the odd lattice L.

A direct computation also easily shows that $\mathbf{N}_{80}$ has no roots. After applying suitable reduction, the verification typically takes less than 30 minutes. We did not try a similar computation with norm 4, as we estimate that it would likely take a few months.

No vectors of norm 2 or 4. We let B_{80}^e be the sublattice of B_{80} given by vectors with even coordinates in the T_{16} basis, and map $B_{80}^e \to B_{80}^e/2 \to \mathbf{F}_2^{80}$ via the additive coordinate map generated by $\pm 2 \to \pm 1 \to 1$. The image in $\mathbf{F}_2^{80}$ is a binary code C_2, and this inherits the automorphisms from the lattice.

We have $16 | \langle \boldsymbol{v}, \boldsymbol{w} \rangle$ for any $\boldsymbol{v}, \boldsymbol{w} \in B_{80}^e$, which implies that C_2 is doubly-even, that is, each codeword has weight divisible by 4. Similarly, we see that $C_2 \subseteq C_2^\perp$, as the inner product between any two codewords is 0 (in $\mathbf{F}_2$). We then show equality here by finding enough vectors in B_{80}^e to show that $\dim(\mathrm{C}_2) \geq 40$.

As C_2 is self-dual and has automorphism group $\mathbf{PSL}_2(\mathbf{F}_{79})$, it follows from either [25, Theorem 6.2] or [24, Satz 3.4] that C_2 is equivalent to the extended

[7] It seems that showing the minimal distance exceeds 20 would take about 58 days, though the computation should parallelise.

binary quadratic residue code,[8] and thus has minimal weight of 16 with 97 565 minimal codewords which lie in 3 orbits under the automorphisms.[9]

We now check that the preimages of codewords of weight 0 and 16 in C_2 do not yield vectors of norm 2 or 4 in $\mathbf{N}_{80}$.[10] This is done using the explicit form of T_{16}^{-1}. For weight 0, we need to check that $T_{16}^{-1}\boldsymbol{w}$ is non-integral for

$$\boldsymbol{w} = \langle 8, 0, \ldots, 0\rangle, \langle 4, \pm 4, 0, \ldots, 0\rangle, \langle 4, \pm 4, (\ldots)\rangle$$

where in this third expression exactly two of the latter 78 coordinates have size 4. By the doubly transitive nature of the automorphism action, this suffices. There are thus $3 + 2^3\binom{78}{2} = 24\,027$ possibilities to check here.

For weight 16, we have 3 orbits of codewords. For each orbit we take a representative, and lift its nonzero coordinates in 2^{16} ways to every choice of sign for ± 2. We then apply T_{16}^{-1} to each, and note that none are integral. This completes the proof that there are no vectors of norm 2 or 4 in the lattice $\mathbf{N}_{80}$. Presumably we could similarly show that B_{80}^e has no vectors of norm 96, but extending our observations to odd-coordinate vectors in B_{80} looks more difficult.

5.2 Vectors with a Nontrivial Stabiliser

We now describe how to use the known automorphisms to reduce our vector-finding quota from 7.5 trillion vectors down to about 15.3 million. We make a separate computation of the norm 10 vectors that have nontrivial stabiliser. If a vector $\boldsymbol{v}$ has a nontrivial stabiliser under the above action of $G = \mathbf{SL}_2(\mathbf{F}_{79})$, there is some nontrivial element $g \in G$ such that the kernel of $\big(g - \mathrm{id}\big)$ contains $\boldsymbol{v}$. So we loop over nontrivial elements (or conjugacy classes) of G, compute this kernel (which is a sublattice), and then search for short vectors in it. The elements of order 3 give a kernel sublattice of dimension 28, for which it takes a few seconds to find the vectors of norm ≤ 10. These yield 465 orbit classes under the action. The elements of order 5, 39, and 79 give lattices of dimensions 16, 4, and 2, and yield 15, 2, and 1 orbits respectively. Upon computing the stabilisers, we obtain

- 1 orbit with stabiliser size $79 \cdot 39 = 3081$ (order 79),
- 2 orbits with stabiliser size 39 (order 39),
- 15 orbits with stabiliser size 5 (order 5),
- 465 orbits with stabiliser size 3 (order 3).

[8] We thank Elkies for recalling this fact, and J. Cannon for the Klemm reference.

[9] Here is an alternative method. Assume first that there is a codeword $\boldsymbol{w}$ of weight 4 or 8. Take a 79-cycle σ and note that since $(8-1)^2 < 79$ there is some iterate of σ such that $\boldsymbol{w}$ and $\sigma\boldsymbol{w}$ intersect only in the fixed coordinate. This implies that $\langle \boldsymbol{w}, \sigma\boldsymbol{w}\rangle = 1$, which contradicts that C_2 is self-dual. Since there are no codewords of weight 4 or 8, we can then apply Gleason's theorem [16] and get that the weight enumerator is of the form $q^0 + (a + 15\,200)\,q^{12} + (127\,965 + 2a)\,q^{16} + (11\,347\,488 - 101a)\,q^{20} + \ldots$ for some $a \in \mathbf{Z}$, and in an echo of our proof of lattice extremality, show code extremality (no codewords of weight 12) via finding 12 882 688 codewords of weight 20; for this, we find short vectors in the lattice, map to the code, and apply automorphisms.

[10] We do not explicitly need the fact that the code is extremal for this step, but only that we have all codewords of length 16 or less.

None of the other 78 nontrivial conjugacy classes of $\mathbf{SL}_2(\mathbf{F}_{79})$ yields an orbit with vectors of norm 10. We can also note that there no vectors of norm 6 with a nontrivial stabiliser (though this is not strictly necessary for our proof).

An accounting then tells us that there are presumably 7 541 323 277 280 vectors of norm 10 yet unfound, and dividing by $\#\mathbf{SL}_2(\mathbf{F}_{79}) = 492\,960$ predicts 15 298 043 orbits with trivial stabiliser. Via a standard coupon-collecting analysis [11, p. 213] we expect that about 250 million suitably random vectors of norm 10 should suffice to hit each orbit at least once.

In fact, for the purposes of proving the lattice extremal, we need only find $(15\,298\,043 - 12\,635)$ orbits (see the q^5 coefficient of f_3, and use the fact that $492\,960|a_3$ as we find no vectors of norm 6 with nontrivial stabiliser), and due to the lengthy final part of coupon-collecting,[11] this reduces the expected running time by about 55%. However, for completeness, we still chose to find all orbits.

6 General Search for Vectors of Norm 10

The general method to enumerate short vectors in a lattice is due to Kannan [21] and Fincke and Pohst [12]. This corresponds to the computation of the leaves of a huge tree. As noted by Schnorr and Euchner [38], this tree can be pruned to some extent. This can be thought of as searching first in the areas of the search region which are more likely to contain short vectors, or, equivalently, removing the tree nodes that are less likely to produce useful leaves. The initial pruning strategy was later improved in [39]. We describe below a further improvement.

6.1 The Full KFP Tree Search

The basic method iteratively looks at the projections to the span of the first i coordinates for decreasing i. We have a basis given by $\{\boldsymbol{b}_i\}$ and wish to solve the inequality $\|\sum_i x_i \boldsymbol{b}_i\|^2 \leq 10$. Borrowing the common notation for lattice reduction, we take the Gram-Schmidt orthogonalisation, and translate the x_i's by the $\mu_{j,i}$'s:

$$\boldsymbol{b}_i^\star = \boldsymbol{b}_i - \sum_{j<i} \mu_{i,j}\boldsymbol{b}_j^\star \text{ so that } \mu_{i,j} = \frac{\langle \boldsymbol{b}_i, \boldsymbol{b}_j^\star \rangle}{\|\boldsymbol{b}_j^\star\|^2} \text{ for } i > j, \text{ and } y_i = x_i + \sum_{j=i+1}^{d} \mu_{j,i} x_j.$$

Here d is the dimension. By substituing y_i for x_i, we get $\sum_i y_i^2 \|\boldsymbol{b}_i^\star\|^2 \leq 10$, which by positivity leads to the series of inequalities:

$$\begin{aligned} y_d^2 \|\boldsymbol{b}_d^\star\|^2 &\leq 10, \\ y_{d-1}^2 \|\boldsymbol{b}_{d-1}^\star\|^2 &\leq 10 - y_d^2 \|\boldsymbol{b}_d^\star\|^2, \\ &\cdots \\ y_1^2 \|\boldsymbol{b}_1^\star\|^2 &\leq 10 - \sum_{i=2}^{d} y_i^2 \|\boldsymbol{b}_i^\star\|^2. \end{aligned}$$

[11] The comparison is between $\sum_{n=1}^{N} \frac{N}{n}$ and $\sum_{n=12636}^{N} \frac{N}{n}$ for $N = 15\,298\,043$.

Note that for all i, the variable x_i is an integer, while y_i is a shift of x_i by a fixed amount (once $x_{i+1}, \ldots, x_d$ have been chosen). The KFP method proceeds by looking at all y_d's satisfying the first inequality, then all pairs (y_{d-1}, y_d) satisfying the second, etc. In particular, the vectors with $y_i \approx 0$ for all i up to a given point will be found most easily (and these often correspond to small x_i's). Also, to find more short vectors earlier in the search procedure, it is useful to run over the different possible x_i's from the centre of the interval implied by the inequality $y_i^2 \|\boldsymbol{b}_i^*\|^2 \leq 10 - \sum_{j>i} y_j^2 \|\boldsymbol{b}_j^*\|^2$: the variable x_i will run across the integers by decreasing proximity to $-\sum_{j>i} \mu_{j,i} x_j$. This "zig-zag" strategy, introduced by Schnorr and Euchner [38], allows one to split the search of the tree in different stages: in the first stage, we have $x_j = 0$ for all $j > 1$; then in the second stage we have $x_j = 0$ for all $j > 2$ but $x_2 \neq 0$; etc. We call stage i the period of time during which $x_j = 0$ for all $j > i$ but $x_i \neq 0$. Stage i means that we have already reached level i in the KFP tree but not yet been in level $i + 1$ (level 1 corresponding to the leaves).

The arithmetic operations corresponding to Gram-Schmidt orthogonalisation computations can be quite slow. The Magma [5] implementation of the KFP tree search replaces them by double precision floating-point arithmetic operations, in a fully reliable way (using [34]).

6.2 Tree Pruning

Our pruning strategy consists in restricting the above inequalities by a "pruning factor" that depends on the level. So the above inequalities become

$$\sum_{i=j}^{d} y_i^2 \|\boldsymbol{b}_i^\star\|^2 \leq 10 \cdot P_j, \ \forall j$$

where P_j is the jth pruning factor. A version of this with a specific choice of P_j appears in [38, §7], and the general description as well as its analysis below have been independently obtained in [15, §4]. In the latter, the authors also introduce the concept of "extreme pruning", which resembles but differs from our bases switching strategy (see subsection below).

The "best" choice for the pruning factors appears to be something like $P_j = (d-j+1)/d$. We happened to choose $P_j = 1 - (j-1)/100$ in practise. The idea here can be phrased as follows: we have a given quantity of "norm" (here 10) to spend on a vector; if we spend a lot on the coordinates x_j to x_d, there will then be a lesser chance that we can form an integral vector via some possible choice of the other coordinates, due to positivity and the fact that most coordinates will have at least some nonzero contribution.

Efficacy of pruning. To give an idea of the efficacy of pruning, we can use the notion, from [19], of expected enumeration cost for a given lattice basis $\{\boldsymbol{b}_i\}$ and for vectors of norm A (a function `EnumerationCost` is available in Magma [5]):

$$\sum_{j=1}^{d} \frac{\sqrt{\pi^{d-j+1} \prod_{k=j}^{d} A / \|\boldsymbol{b}_k^\star\|^2}}{\Gamma\big(1 + (d-j+1)/2\big)}. \tag{2}$$

A typical enumeration cost for our bases with $\mathbf{N}_{80}$ was around 10^{23}. This is the expected number of nodes of the KFP tree. For comparison, the implementation in Magma [5] has a traversal rate of about 7.5 million nodes per second.

By comparing this enumeration cost estimate to the expected $7.5 \cdot 10^{12}$ vectors of norm 10, we find that more than 10^{10} nodes are expected to be searched for each vector found. In the case of the pruned enumeration, the jth summand in (2) should be multiplied by the volume of the truncated hypersphere $\{(z_j, \ldots, z_d) : \forall i \geq j, \sum_{k \geq i} z_k^2 \leq P_i\}$. By estimating these volumes with a Monte-Carlo rejection method (uniformly sampling points in the full hypersphere and counting how many belong to the truncation), we expect our pruning to gain a factor of around 10^4 here, at the cost of missing about 60% of the short vectors. These speedup and miss ratios are not constant across all levels of the search: they seem to be closer to 100 and 25% respectively for the levels of our interest (due to the early abort and perturbation strategy described below).

6.3 Switching Bases

The early stages of the tree search can have a significantly better chance of providing short vectors, due primarily to the relative paucity of "uninteresting" branches that tend to become more numerous at higher levels. In practice, we would find 10^5 vectors in about 30 minutes, for a ratio of about 150 000 nodes searched for each vector found, more than an order of magnitude lower than the above estimate, even with the pruning included.

Every 15-30 minutes we would switch the basis by applying a random permutation to the coordinates of the current basis, and then multiplying by a random upper triangular matrix with ones on the diagonal and off-diagonal entries in $\{-1, 0, +1\}$. We then re-apply LLL (with a δ-value nearly 1) to the perturbed basis, and then LLL with deep insertions [38]. Overall, this takes only a few seconds. This basis switching also makes parallelisation essentially trivial.

A second reason for periodically changing the basis is that (a phenomenon we found experimentally) there are some bases which "hide" many of the orbits, in the sense that every vector in such an orbit would not be found until we reach one of the latter stages. We currently have no explanation of this.

7 Conclusion and Related Work

We implemented the above in a combination of Magma [5] and `C`. As we typically found 10^5 vectors of norm 10 in about 30 minutes, the estimated time was around 52 days. Using 14 processors in parallel, it took us about 4 days in April 2009.

7.1 Software to Check Our Data

A verification of our proof can be done in much less time than the computation itself. We provide software[12] that takes less than 10 hours to verify that $\mathbf{N}_{80}$

[12] The code is `checkit80.c` (to be run with arguments "10 ⟨filename⟩") and the data is `LAT80.n10.sc16.bz2` in the directory `http://magma.maths.usyd.edu.au/~watkins`

is indeed extremal. The input consists of 15 298 526 entries that correspond to coordinate vectors in the T_{16} basis of Section 4. The following checks are run:

- Each entry lexicographically follows its predecessor,
- Each entry has norm 160 and is integral when multiplied by T_{16}^{-1},
- Each entry is lexicographically the first in its orbit.

The first condition ensures that all entries are distinct, while the last ensures that each corresponds to a distinct orbit, with the middle condition implying that the vectors have norm 10 and are in $\mathbf{N}_{80}$. We can also list the 483 orbits with nontrivial stabiliser, whose provenance can be checked separately.

7.2 Three Lattices of Dimension 72

The work in progress [1] investigates three lattices of dimension 72. Two of these are 2-neighbours of a lattice constructed via the extended quadratic residue code over $\mathbf{F}_3$, and the other involves a code over $\mathbf{Z}/4\mathbf{Z}$. None of these turned out to be extremal (minimal norm of 8), and indeed, we know of no extremal lattice of this dimension. In fact, a recent preprint of Griess [17] claims to be the first to prove a minimal norm as large as 6 for an even unimodular lattice of dimension 72.

7.3 Other Candidate Lattices for Extremality in Dimension 80

In [3], the authors note three other candidates for extremality amongst even unimodular lattices in dimension 80. One candidate comes from a cyclo-quaternionic construction given in [31, Remark 5.2], and its automorphism group contains $\mathbf{SL}_2(\mathbf{F}_{41}) \otimes \tilde{S}_3$, which is of comparable size to our $\mathbf{SL}_2(\mathbf{F}_{79})$. We do not see how to facilitate the calculation of canonical orbit representatives as readily as in our case, but the fact that canonicalising took only about 5% of our running time indicates that our methods could work in this case, with sufficient effort.

The other two candidates come from a cyclotomic construction explored in [4], and have an automorphism group containing the general affine linear group $\mathbf{F}_{41}^{+} \rtimes \mathbf{F}_{41}^{*}$. Our initial opinion is that the automorphism group (even if augmented by an order 4 element) is too small for our method to work well here.

Acknowledgments. We thank the authors of [1], with whom we started this research, and S. R. Donnelly who shared some of his ideas with us. We also thank the anonymous reviewers for their recommendation to add a proof that the automorphism group of $\mathbf{N}_{80}$ differs from those of $\mathbf{L}_{80}$ and $\mathbf{M}_{80}$. The present work is part of the Australian Research Council Discovery Project DP0880724 "Integral lattices and their theta series".

References

1. Abel, Z., Elkies, N.D., Kominers, S.D.: On 72-dimensional lattices (in preparation)
2. Aschbacher, M.: On the maximal subgroups of the finite classical groups. Invent. Math. 76(3), 469–514 (1984), http://dx.doi.org/10.1007/BF01388470

3. Bachoc, C., Nebe, G.: Extremal lattices of minimum 8 related to the Mathieu group M_{22}. J. Reine Angew. Math. 494, 155–171 (1998), http://dx.doi.org/10.1515/crll.1998.004
4. Batut, C., Quebbemann, H.-G., Scharlau, R.: Computations of cyclotomic lattices. Experiment. Math. 4(3), 177–179 (1995), http://www.expmath.org/restricted/4/4.3/batut.ps
5. Bosma, W., Cannon, J., Playoust, C.: The Magma algebra system. I. The user language. In: Cannon, J., Holt, D. (eds.) Computational algebra and number theory, Proceedings of the 1st Magma Conference held at Queen Mary and Westfield College, London, August 23-27, 1993, pp. 235–265. Elsevier Science B.V, Amsterdam (1997); Cross-referenced as J. Symbolic Comput. 24(3-4), 235–265 (1997), http://magma.maths.usyd.edu.au
6. Conway, J.H., Odlyzko, A.M., Sloane, N.J.A.: Extremal self-dual lattices exist only in dimensions 1 to 8, 12, 14, 15, 23, and 24. Mathematika 25(1), 36–43 (1978), http://dx.doi.org/10.1112/S0025579300009244
7. Conway, J.H., Sloane, N.J.A.: Sphere packings, lattices and groups. In: Grundlehren der Mathematischen Wissenschaften. Fundamental Principles of Mathematical Sciences, vol. 290, xxviii+663 pp. Springer, New York (1988)
8. Conway, J.H., Sloane, N.J.A.: A new upper bound on the minimal distance of self-dual codes. IEEE Trans. Inform. Theory 36(6), 1319–1333 (1990), http://dx.doi.org/10.1109/18.59931
9. Feit, W.: On integral representations of finite groups. Proc. London Math Soc. 29(3), 633–683 (1974), http://plms.oxfordjournals.org/cgi/reprint/s3-29/4/633
10. Feit, W.: Orders of finite linear groups. In: Foguel, T., Minty, J. (eds.) Proceedings of the First Jamaican Conference on Group Theory and its Applications 1996, University of the West Indies, Mona Campus, Kinsgston, Jamaica , January 9-12, pp. 9–11 (1997)
11. Feller, W.: Introduction to Probability Theory, vol. I. John Wiley & Sons, New York (1950)
12. Fincke, U., Pohst, M.: A procedure for determining algebraic integers of given norm. In: van Hulzen, J.A. (ed.) Proceedings of the European computer algebra conference (EUROCAL), Computer Algebra, London. LNCS, vol. 162, pp. 194–202. Springer, Berlin (1983), http://dx.doi.org/10.1007/3-540-12868-9_103
13. Gaborit, P.: A bound for certain s-extremal lattices and codes. Arch. Math. (Basel) 89(2), 143–151 (2007), http://dx.doi.org/10.1007/s00013-006-1164-5
14. Gaborit, P.: Construction of new extremal unimodular lattices. Eur. J. Combin. 25(4), 549–564 (2004), http://dx.doi.org/10.1016/j.ejc.2003.07.005
15. Gama, N., Nguyen, P.Q., Regev, O.: Lattice Enumeration Using Extreme Pruning. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110. Springer, Heidelberg (to appear, 2010)
16. Gleason, A.M.: Weight polynomials of self-dual codes and the MacWilliams identities. In: Proceedings of the International Congress of Mathematicians, Actes du Congrès International des Mathématiciens (Nice, 1970), Tome 3, Gauther-Villars, Paris, pp. 211–215 (1971)
17. Griess Jr., R.L.: Rank 72 high minimum norm lattices (preprint), http://arxiv.org/abs/0910.2055
18. Gross, B.H.: Group representations and lattices. J. Amer. Math. Soc. 3(4), 929–960 (1990), http://dx.doi.org/10.2307/1990907

19. Hanrot, G., Stehlé, D.: Improved Analysis of Kannan's Shortest Lattice Vector Algorithm. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 390–405. Springer, Heidelberg (2007), http://dx.doi.org/10.1007/978-3-540-74143-5_10
20. Hiss, G., Malle, G.: Low-dimensional representations of quasi-simple groups. LMS J. Comput. Math. 4, 22–63 (2001); Corrigenda: LMS J. Comput. Math. 5, 95–126 (2002), http://www.lms.ac.uk/jcm/4/lms2000-014/sub/lms2000-014.pdf, http://www.lms.ac.uk/jcm/5/lms2002-025/sub/lms2002-025.pdf
21. Kannan, R.: Improved algorithms for integer programming and related lattice problems. In: Proceedings of the fifteenth annual ACM symposium on the Theory of computing, STOC 1983, Boston, MA, pp. 99–108 (1983). ACM order #508830, http://doi.acm.org/10.1145/800061.808749
22. King, O.: A mass formula for unimodular lattices with no roots. Math. Comp. 72(242), 839–863 (2003), Available online from the publisher (the AMS) via, http://www.ams.org/mcom/2003-72-242/S0025-5718-02-01455-2
23. Kleidman, P.B., Liebeck, M.W.: The subgroup structure of the finite classical groups. London Mathematical Society Lecture Note Series, vol. 129, x+303 pp. Cambridge University Press, Cambridge (1990)
24. Klemm, M.: Kennzeichnung der erweiterten quadrate-codes durch ihre $\mathbf{PSL}(2, q)$-zulässigkeit. (German). Characterising the extended quadratic-codes by their $\mathbf{PSL}(2, q)$-admissibility. Communications in Algebra 11(18), 2051–2068 (1983), http://dx.doi.org/10.1080/00927878308822949
25. Knapp, W., Schmid, P.: Codes with prescribed permutation group. J. Algebra 67, 415–435 (1980), http://dx.doi.org/10.1016/0021-8693(80)90169-6
26. Kneser, M.: Klassenzahlen definiter quadratischer Formen. (German) [Class numbers of definite quadratic forms]. Arch. Math. 8, 241–250 (1957), http://dx.doi.org/10.1007/BF01898782
27. Krasikov, I., Litsyn, S.: An improved upper bound on the minimum distance of doubly-even self-dual codes. IEEE Trans. Inform. Theory 46(1), 274–278 (2000), http://dx.doi.org/10.1109/18.817527
28. Mallows, C.L., Odlyzko, A.M., Sloane, N.J.A.: Upper bounds for modular forms, lattices, and codes. J. Algebra 36(1), 68–76 (1975), http://dx.doi.org/10.1016/0021-8693(75)90155-6
29. Minkowski, H.: Zur Theorie der positiven quadratischen Formen (German) [On the Theory of positive quadratic Forms]. J. reine angew. Math. 101, 196–202 (1887), http://resolver.sub.uni-goettingen.de/purl?GDZPPN002160390
30. Miyake, T.: Modular Forms. Springer, Berlin (1989)
31. Nebe, G.: Some cyclo-quaternionic lattices. J. Algebra 199(2), 472–498 (1998), http://dx.doi.org/10.1006/jabr.1997.7163
32. Nebe, G., Schindelar, K.: S-extremal strongly modular lattices. J. Théor. Nombres Bordeaux 19(3), 683–701 (2007), http://jtnb.cedram.org/item?id=JTNB_2007__19_3_683_0
33. Peters, M.: Definite unimodular 48-dimensional quadratic forms. Bull. London Math. Soc. 15(1), 18–20 (1983), http://blms.oxfordjournals.org/cgi/content/citation/15/1/18
34. Pujol, X., Stehlé, D.: Rigorous and efficient short lattice vectors enumeration. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 390–405. Springer, Heidelberg (2008), http://www.springerlink.com/content/978-3-540-89254-0

35. Quebbemann, H.-G.: Zur Klassifikation unimodularer Gitter mit Isometrie von Primzahlordnung (German) [On the classification of unimodular lattices with an isometry of prime order]. J. Reine Angew. Math. 326, 158–170 (1981), http://resolver.sub.uni-goettingen.de/purl?GDZPPN002198681; Quebbemann, H.-G.: Unimodular lattices with isometries of large prime order. II. Math. Nachr. 156, 219–224 (1992), http://dx.doi.org/10.1002/mana.19921560114
36. Quebbemann, H.-G.: Atkin-Lehner eigenforms and strongly modular lattices. Enseign. Math. 43(1-2), 55–65 (1997), http://retro.seals.ch/digbib/view?rid=ensmat-001:1997:43::263
37. Rains, E.M.: New asymptotic bounds for self-dual codes and lattices. IEEE Trans. Inform. Theory 49(5), 1261–1274 (2003), http://dx.doi.org/10.1109/TIT.2003.810623
38. Schnorr, C.P., Euchner, M.: Lattice Basis Reduction: Improved Practical Algorithms and Solving Subset Sum Problems. Math. Program. 66, 181–191 (1994), http://dx.doi.org/10.1007/BF01581144
39. Schnorr, C.P., Hörner, H.H.: Attacking the Chor-Rivest cryptosystem by improved lattice reduction. In: Guillou, L.C., Quisquater, J.-J. (eds.) EUROCRYPT 1995. LNCS, vol. 921, pp. 1–12. Springer, Heidelberg (1995), http://dx.doi.org/10.1007/3-540-49264-X_1
40. Schulze-Pillot, R.: Quadratic residue codes and cyclotomic lattices. Arch. Math. (Basel) 60(1), 40–45 (1993), http://dx.doi.org/10.1007/BF01194237
41. Weisfeiler, B.: On the size of structure of finite linear groups. Notes from 1984, Parts 1-17, A1-A10, totalling 91 typewritten and 63 handwritten pages, http://weisfeiler.com/boris/papers/papers.html
42. Zagier, D.: Elliptic modular forms and their applications. In: Ranestad, K. (ed.) The 1-2-3 of modular forms. Lectures from the Summer School on Modular Forms and their Applications held in Nordfjordeid, Universitext, June 2004, x+266 pp. Springer, Berlin (2008)

A Appendix: Proof That $\mathbf{M}_{80}$ and $\mathbf{N}_{80}$ Are Not Isometric

We wish to show that $\mathbf{M}_{80}$ is not isometric to our lattice $\mathbf{N}_{80}$. Bachoc and Nebe list a subgroup of $\mathrm{Aut}(\mathbf{M}_{80})$ of order $2^{12}3^45^2$, while we have $S \cong \mathbf{SL}_2(\mathbf{F}_{79})$ as a subgroup of $\mathrm{Aut}(\mathbf{N}_{80})$. We wish to show that there is no finite matrix group in $\mathbf{GL}_{80}(\mathbf{Z})$ that is a supergroup of both of these (possibly after conjugation).

We let G be such a putative supergroup, and note that $[G : S] \geq 2^7 3^3 5$. From a classical theorem of Minkowski [29] on the modular reduction of matrix groups, we have injective maps $\iota_p : G \hookrightarrow \mathbf{GL}_{80}(\mathbf{F}_p)$ for all odd primes p. By taking a gcd over all odd p this gives a bound of

$$\#G \mid 2^{198}3^{58}5^{24}7^{14}11^8 13^6 17^5 19^4 23^3 29^2 31^2 37^2 41^2 \cdot 43 \cdot 47 \cdot 53 \cdot 59 \cdot 61 \cdot 67 \cdot 71 \cdot 73 \cdot 79,$$

though here we really only need such a divisibility result at a specific prime.[13]

We write $H = \iota_7\big(G \cap \mathbf{SL}_{80}(\mathbf{Z})\big)$, and since every matrix in $S \cong \mathbf{SL}_2(\mathbf{F}_{79})$ has determinant 1 we have $\iota_7(S) \subseteq H$. As every matrix in G has determinant ± 1,

[13] We note in passing that the best upper bound on the size of a finite matrix group is due to Feit [10], relying on unpublished notes of Weisfeiler [41].

we get $[\iota_7(G) : H] \leq 2$, and since $[G : S] > 4$ and ι_7 is injective, this implies that $[H : \iota_7(S)] > 2$. The use of a theorem of Aschbacher (see below) now implies that $7^{780} \mid \#H$, which contradicts the above bound. Thus G cannot exist, and so $\mathbf{M}_{80}$ and $\mathbf{N}_{80}$ are not isometric. Indeed, this argument almost shows that S is maximal finite in $\mathbf{GL}_{80}(\mathbf{Z})$, though a low-index extension could still exist.

We now use Aschbacher's theorem [2] on maximal subgroups of finite classical groups (see also [23]). Let l be an odd prime (to be specified below) and suppose that $\iota_l(S) \subset H \subseteq \mathbf{SL}_{80}(\mathbf{F}_l)$. We note that S splits into a pair of conjugate absolutely irreducible unitary 40-dimensional representations defined over $\mathbf{Q}(\sqrt{-79})$.

We know that H lies in some maximal (proper) subgroup of $\mathbf{SL}_{80}(\mathbf{F}_l)$, and the theorem of Aschbacher lists the possibilities. For any inert prime l that does not divide $\#S$, we can eliminate class 1 of Aschbacher since $\iota_l(S)$ acts irreducibly (we could consider split primes also, but choosing an inert prime simplifies the argument slightly). Classes 2 and 4-7 are not possible simply because 79 must divide $\#H$. This leaves subgroups of class 3 (splitting as above) or class 8 (inclusions of classical groups), or class 9 (other simple groups, handled below). The inclusions of classical groups give us $\mathbf{G}_{80}(\mathbf{F}_l)$ for $\mathbf{G} = \mathbf{Sp}, \mathbf{SO}^{\pm}$ and $\mathbf{SU}_{40}(\mathbf{F}_l)$, while the splitting of class 3 yields $\mathbf{SL}_{40}(\mathbf{F}_{l^2}).2$. where the notation indicates that we have a 2-extension – in this case, we continue the analysis after replacing H by $H \cap \mathbf{SL}_{40}(\mathbf{F}_{l^2})$, where this subgroup has index at most 2 in H.

We iteratively apply Aschbacher's theorem to each classical group obtained; either H is isomorphic to this classical group, or is contained in a maximal subgroup of it. We again use $79|\#H$, and find that the only possible maximal subgroup of $\mathbf{Sp}_{80}(\mathbf{F}_l)$ that could contain H is $\mathbf{SU}_{40}(\mathbf{F}_l).2$, and similarly with the others. Any maximal subgroup chain of classical groups must end here, since H contains $\iota_l(S)$ and $S \to \mathbf{SU}_{40}(\mathbf{F}_l)$ is absolutely irreducible.

So we end in one of the following cases: H is isomorphic to one of

$$\mathbf{SU}_{40}(\mathbf{F}_l).\epsilon \text{ or } \mathbf{SL}_{40}(\mathbf{F}_{l^2}).\epsilon \text{ with } \epsilon = 1, 2, \text{ or } \mathbf{G}_{80}(\mathbf{F}_l) \text{ with } \mathbf{G} = \mathbf{Sp}, \mathbf{SO}^{\pm}, \mathbf{SL};$$

or $[H : \iota_l(S)] = 2$, in correspondence to a 2-extension as above; or (sometimes called "class 9" for Aschbacher) we have $\mathbf{PSL}_2(\mathbf{F}_{79}) \subset K \subset \mathbf{P}$, where K is simple and $\mathbf{P}$ is the associated simple group of one of the above classical groups.

There is sundry general knowledge for this latter situation, but for us a case-by-case analysis (with $l = 7$ for concreteness) using the known orders of the finite simple groups is sufficient to show that no such K can exist.[14] We conclude that either $[H : \iota_7(S)] = 2$, or that H contains a copy of $\mathbf{SU}_{40}(\mathbf{F}_7)$ and so $7^{780} \mid \#H$.

[14] One can also proceed via degrees of representations, and D. F. Holt indicated to us that the tables of Hiss and Malle [20] should suffice for this.

Computing Automorphic Forms on Shimura Curves over Fields with Arbitrary Class Number

John Voight

Department of Mathematics and Statistics
University of Vermont
16 Colchester Ave
Burlington, VT 05401, USA
jvoight@gmail.com

Abstract. We extend methods of Greenberg and the author to compute in the cohomology of a Shimura curve defined over a totally real field with arbitrary class number. Via the Jacquet-Langlands correspondence, we thereby compute systems of Hecke eigenvalues associated to Hilbert modular forms of arbitrary level over a totally real field of odd degree. We conclude with two examples which illustrate the effectiveness of our algorithms.

The development and implementation of algorithms to compute with automorphic forms has emerged as a major topic in explicit arithmetic geometry. The first such computations were carried out for elliptic modular forms, and now very large and useful databases of such forms exist [2,13,14]. Recently, effective algorithms to compute with Hilbert modular forms over a totally real field F have been advanced. The first such method is due to Dembélé [4,5], who worked initially under the assumption that F has even degree $n = [F : \mathbb{Q}]$ and strict class number 1. Exploiting the Jacquet-Langlands correspondence, systems of Hecke eigenvalues can be identified inside spaces of automorphic forms on $B^\times$, where B is the quaternion algebra over F ramified precisely at the infinite places of F—whence the assumption that n is even. Dembélé then provides a computationally efficient theory of Brandt matrices associated to B. This method was later extended (in a nontrivial way) to fields F of arbitrary class number by Dembélé and Donnelly [6].

When the degree n is odd, a different algorithm has been proposed by Greenberg and the author [8], again under the assumption that F has strict class number 1. This method instead locates systems of Hecke eigenvalues in the (degree one) cohomology of a Shimura curve, now associated to the quaternion algebra B ramified at all but one real place and no finite place. This method uses in a critical way the computation of a fundamental domain and a reduction theory for the associated quaternionic unit group [16]; see Section 1 for an overview. In this article, we extend this method to the case where F has arbitrary (strict) class number. Our main result is as follows; we refer the reader to Sections 1 and 2 for precise definitions and notation.

G. Hanrot, F. Morain, and E. Thomé (Eds.): ANTS-IX 2010, LNCS 6197, pp. 357–371, 2010.

Theorem 1. *There exists an (explicit) algorithm which, given a totally real field F of degree $n = [F : \mathbb{Q}]$, a quaternion algebra B over F ramified at all but one real place, an ideal $\mathfrak{N}$ of F coprime to the discriminant $\mathfrak{D}$ of B, and a weight $k \in (2\mathbb{Z}_{>0})^n$, computes the system of eigenvalues for the Hecke operators $T_{\mathfrak{p}}$ with $\mathfrak{p} \nmid \mathfrak{D}\mathfrak{N}$ and the Atkin-Lehner involutions $W_{\mathfrak{p}^e}$ with $\mathfrak{p}^e \parallel \mathfrak{D}\mathfrak{N}$ acting on the space of quaternionic modular forms $S_k^B(\mathfrak{N})$ of weight k and level $\mathfrak{N}$ for B.*

In other words, there exists an explicit finite procedure which takes as input the field F, its ring of integers $\mathbb{Z}_F$, a quaternion algebra B over F, an ideal $\mathfrak{N} \subset \mathbb{Z}_F$, and the vector k encoded in bits (each in the usual way), and outputs a finite set of number fields $E_f \subset \overline{\mathbb{Q}}$ and sequences $(a_f(\mathfrak{p}))_{\mathfrak{p}}$ encoding the Hecke eigenvalues for each cusp form constituent f in $S_k^B(\mathfrak{N})$, with $a_f(\mathfrak{p}) \in E_f$.

From the Jacquet-Langlands correspondence, applying the above theorem to the special case where $\mathfrak{D} = (1)$ (and hence $n = [F : \mathbb{Q}]$ is odd), we have the following corollary.

Corollary 2. *There exists an algorithm which, given a totally real field F of odd degree $n = [F : \mathbb{Q}]$, an ideal $\mathfrak{N}$ of F, and a weight $k \in (2\mathbb{Z}_{>0})^n$, computes the system of eigenvalues for the Hecke operators $T_{\mathfrak{p}}$ and Atkin-Lehner involutions $W_{\mathfrak{p}^e}$ acting on the space of Hilbert modular cusp forms $S_k(\mathfrak{N})$ of weight k and level $\mathfrak{N}$.*

This corollary is not stated in its strongest form: in fact, our methods overlap with the methods of Dembélé and his coauthors whenever there is a prime $\mathfrak{p}$ which exactly divides the level; see Remark 5 for more detail. Combining these methods, Donnelly and the author [7] are systematically enumerating tables of Hilbert modular forms, and the details of these computations (including the dependence on the weight, level, and class number, as well as a comparison of the runtime complexity of the steps involved) will be reported there [7], after further careful optimization.

A third technique to compute with automorphic forms, including Hilbert modular forms, has been advanced by Gunnells and Yasaki [9]. They instead use the theory of Voronoĭ reduction and sharbly complexes; their work is independent of either of the above approaches.

This article is organized as follows. In Section 1, we give an overview of the basic algorithm of Greenberg and the author which works over fields F with strict class number 1. In Section 2, using an adelic language we address the complications which arise over fields of arbitrary class number, and in Section 3 we make this theory concrete and provide the explicit algorithms announced in Theorem 1. Finally, in Section 4, we consider two examples, one in detail; our computations are performed in the computer system Magma [1].

The author would like to thank Steve Donnelly and Matthew Greenberg for helpful discussions as well as the referees for their comments. The author was supported by NSF Grant No. DMS-0901971.

1 An Overview of the Algorithm for Strict Class Number 1

In this section, we introduce the basic algorithm of Greenberg and the author [8] with a view to extending its scope to base fields of arbitrary class number; for further reading, see the references contained therein.

Let F be a totally real field of degree $n = [F : \mathbb{Q}]$ with ring of integers $\mathbb{Z}_F$. Let $F_+^\times$ be the group of totally positive elements of F and let $\mathbb{Z}_{F,+}^\times = \mathbb{Z}_F^\times \cap F_+^\times$. Let B be a quaternion algebra over F of discriminant $\mathfrak{D}$. Suppose that B is split at a unique real place v_1, corresponding to an embedding $\iota_\infty : B \hookrightarrow B \otimes \mathbb{R} \cong \mathrm{M}_2(\mathbb{R})$, and ramified at the other real places $v_2, \ldots, v_n$. Let $\mathcal{O}(1) \subset B$ be a maximal order and let

$$\mathcal{O}(1)_+^\times = \{\gamma \in \mathcal{O}(1)^\times : v_1(\mathrm{nrd}(\gamma)) > 0\} = \{\gamma \in \mathcal{O}(1) : \mathrm{nrd}(\gamma) \in \mathbb{Z}_{F,+}^\times\}$$

denote the group of units of $\mathcal{O}(1)$ with totally positive reduced norm. Let

$$\Gamma(1) = \iota_\infty(\mathcal{O}(1)_+^\times / \mathbb{Z}_F^\times) \subset \mathrm{PGL}_2(\mathbb{R})^+,$$

so that $\Gamma(1)$ acts on the upper half-plane $\mathcal{H} = \{z \in \mathbb{C} : \mathrm{Im}(z) > 0\}$ by linear fractional transformations. Let $\mathfrak{N} \subset \mathbb{Z}_F$ be an ideal coprime to $\mathfrak{D}$, let $\mathcal{O} = \mathcal{O}_0(\mathfrak{N})$ be an Eichler order of level $\mathfrak{N}$, and let $\Gamma = \Gamma_0(\mathfrak{N}) = \iota_\infty(\mathcal{O}_0(\mathfrak{N})_+^\times / \mathbb{Z}_F^\times)$.

Let $k = (k_1, \ldots, k_n) \in (2\mathbb{Z}_{>0})^n$ be a weight vector; for example, the case $k = (2, \ldots, 2)$ of parallel weight 2 is of significant interest. Let $S_k^B(\mathfrak{N})$ denote the finite-dimensional $\mathbb{C}$-vector space of quaternionic modular forms of weight k and level $\mathfrak{N}$ for B. Roughly speaking, a form $f \in S_k^B(\mathfrak{N})$ is an analytic function $f : \mathcal{H} \to W_k(\mathbb{C})$ which is invariant under the weight k action by the group $\gamma \in \Gamma$, where $W_k(\mathbb{C})$ is an explicit right $B^\times$-module [8, (2.4)] and $W_k(\mathbb{C}) = \mathbb{C}$ when k is parallel weight 2. The space $S_k^B(\mathfrak{N})$ comes equipped with the action of Hecke operators $T_\mathfrak{p}$ for primes $\mathfrak{p} \nmid \mathfrak{D}\mathfrak{N}$ and Atkin-Lehner involutions $W_{\mathfrak{p}^e}$ for prime powers $\mathfrak{p}^e \parallel \mathfrak{D}\mathfrak{N}$.

The Jacquet-Langlands correspondence [8, Theorem 2.9] (see Hida [10, Proposition 2.12]) gives an isomorphism of Hecke modules

$$S_k^B(\mathfrak{N}) \xrightarrow{\sim} S_k(\mathfrak{D}\mathfrak{N})^{\mathfrak{D}\text{-new}},$$

where $S_k(\mathfrak{D}\mathfrak{N})^{\mathfrak{D}\text{-new}}$ denotes the space of Hilbert modular cusp forms of weight k and level $\mathfrak{D}\mathfrak{N}$ which are new at all primes dividing $\mathfrak{D}$. Therefore, as Hecke modules one can compute equivalently with Hilbert cusp forms or with quaternionic modular forms.

We compute with the Hecke module $S_k^B(\mathfrak{N})$ by identifying it as a subspace in the degree one cohomology of $\Gamma(1)$, as follows. Let $V_k(\mathbb{C})$ be the subspace of the algebra $\mathbb{C}[x_1, y_1, \ldots, x_n, y_n]$ consisting of those polynomials q which are homogeneous in (x_i, y_i) of degree $w_i = k_i - 2$. Then $V_k(\mathbb{C})$ has a right action of the group $B^\times$ given by

$$q^\gamma(x_1, y_1, \ldots, x_n, y_n) = \left(\prod_{i=1}^n (\det \gamma_i)^{-w_i/2}\right) q((x_1\ y_1)\overline{\gamma}_1, \ldots, (x_n\ y_n)\overline{\gamma}_n) \quad (1)$$

for $\gamma \in B^\times$, where $\bar{}$ denotes the standard involution (conjugation) on B and $\gamma_i = v_i(\gamma) \in \mathrm{M}_2(\mathbb{C})$. By the theorem of Eichler and Shimura [8, Theorem 3.8], we have an isomorphism of Hecke modules

$$S_k^B(\mathfrak{N}) \xrightarrow{\sim} H^1\big(\Gamma, V_k(\mathbb{C})\big)^+$$

where the group cohomology H^1 denotes the (finite-dimensional) $\mathbb{C}$-vector space of crossed homomorphisms $f : \Gamma \to V_k(\mathbb{C})$ modulo coboundaries and $+$ denotes the $+1$-eigenspace for complex conjugation. By Shapiro's lemma [8, §6], we then have a further identification

$$S_k^B(\mathfrak{N}) \xrightarrow{\sim} H^1\big(\Gamma, V_k(\mathbb{C})\big)^+ \cong H^1(\Gamma(1), V(\mathbb{C}))^+, \tag{2}$$

where $V(\mathbb{C}) = \mathrm{Coind}_\Gamma^{\Gamma(1)} V_k(\mathbb{C})$.

In the isomorphism (2), the Hecke operators act as follows. Let $\mathfrak{p}$ be a prime of $\mathbb{Z}_F$ with $\mathfrak{p} \nmid \mathfrak{D}\mathfrak{N}$ and let $\mathbb{F}_\mathfrak{p}$ denote the residue class field of $\mathfrak{p}$. Since F has strict class number 1, by strong approximation [15, Theorème III.4.3] there exists $\pi \in \mathcal{O}$ such that $\mathrm{nrd}\,\pi$ is a totally positive generator for $\mathfrak{p}$. It follows that there are elements $\gamma_a \in \mathcal{O}_+^\times$, indexed by $a \in \mathbb{P}^1(\mathbb{F}_\mathfrak{p})$, such that

$$\mathcal{O}_+^\times \pi \mathcal{O}_+^\times = \bigsqcup_{a \in \mathbb{P}^1(\mathbb{F}_\mathfrak{p})} \mathcal{O}_+^\times \alpha_a \tag{3}$$

where $\alpha_a = \pi\gamma_a$.

Let $f : \Gamma(1) \to V(\mathbb{C})$ be a crossed homomorphism, and let $\gamma \in \Gamma(1)$. The decomposition (3) extends to $\mathcal{O}(1)$ as

$$\mathcal{O}(1)_+^\times \pi \mathcal{O}(1)_+^\times = \bigsqcup_{a \in \mathbb{P}^1(\mathbb{F}_\mathfrak{p})} \mathcal{O}(1)_+^\times \alpha_a.$$

Thus, there are elements $\delta_a \in \mathcal{O}(1)_+^\times$ for $a \in \mathbb{P}^1(\mathbb{F}_\mathfrak{p})$ and a unique permutation γ^* of $\mathbb{P}^1(\mathbb{F}_\mathfrak{p})$ such that

$$\alpha_a \gamma = \delta_a \alpha_{\gamma^* a} \tag{4}$$

for all a. We then define $f \mid T_\mathfrak{p} : \Gamma(1) \to V(\mathbb{C})$ by

$$(f \mid T_\mathfrak{p})(\gamma) = \sum_{a \in \mathbb{P}^1(\mathbb{F}_\mathfrak{p})} f(\delta_a)^{\alpha_a}. \tag{5}$$

The space $S_k^B(\mathfrak{N})$ similarly admits an action of Atkin-Lehner operators $W_{\mathfrak{p}^e}$ for primes $\mathfrak{p}^e \parallel \mathfrak{D}\mathfrak{N}$.

From this description, we see that the Hecke module $H^1(\Gamma(1), V(\mathbb{C}))^+$ is amenable to explicit computation. First, we compute a finite presentation for $\Gamma(1)$ with a minimal set of generators G and a solution to the word problem for the computed presentation using an algorithm of the author [16]. Given such a set of generators and relations, one can explicitly find a basis for the $\mathbb{C}$-vector space $H^1(\Gamma(1), V(\mathbb{C}))$ [8, §5].

We then compute the action of the Hecke operator $T_{\mathfrak{p}}$ on $H^1(\Gamma(1), V(\mathbb{C}))$. We first compute a splitting $\iota_{\mathfrak{p}} : \mathcal{O} \hookrightarrow \mathrm{M}_2(\mathbb{Z}_{F,\mathfrak{p}})$. The elements α_a in (4) are then generators with totally positive reduced norm of the left ideals

$$I_a = \mathcal{O}\iota_{\mathfrak{p}}^{-1} \begin{pmatrix} x & y \\ 0 & 0 \end{pmatrix} + \mathcal{O}\mathfrak{p} \tag{6}$$

and are obtained by principalizing the ideals I_a; here again we use strong approximation and the hypothesis that F has strict class number 1. Then for each $a \in \mathbb{P}^1(\mathbb{F}_p)$ and each $\gamma \in G$, we compute the permutation γ^* [8, Algorithm 5.8] and the element $\delta_a = \alpha_a \gamma \alpha_{\gamma^* a}^{-1} \in \Gamma(1)$ as in (4). Using the solution to the word problem, we then write δ_a as a word in the generators G for $\Gamma(1)$, and then for a basis of crossed homomorphisms f we compute $f \mid T_{\mathfrak{p}}$ by computing $(f \mid T_{\mathfrak{p}})(\gamma) \in V(\mathbb{C})$ for each $\gamma \in G$ as in (5). In a similar way, we compute the action of complex conjugation and the Atkin-Lehner involutions. We then decompose the space $H^1(\Gamma, V(\mathbb{C}))$ under the action of these operators into Hecke irreducible subspaces, and from this we compute the systems of Hecke eigenvalues using linear algebra.

2 The Indefinite Method with Arbitrary Class Number

In this section, we show how to extend the method introduced in the previous section to the case where F has arbitrary class number [8, Remark 3.11]. We refer the reader to Hida [11] for further background.

2.1 Setup

We carry over the notation from Section 1. Recall that $\mathcal{O} = \mathcal{O}_0(\mathfrak{N})$ is an Eichler order of level $\mathfrak{N}$ in the maximal order $\mathcal{O}(1) \subset B$.

Let $\mathcal{H}^{\pm} = \{z \in \mathbb{C} : \mathrm{Im}(z) \neq 0\} = \mathbb{C} \setminus \mathbb{R}$ be the union of the upper and lower half-planes. Then via ι_∞, the group $B^\times$ acts on $\mathcal{H}^{\pm}$ by linear fractional transformations.

In this generality, we find it most elucidating to employ adelic notation. Let $\widehat{\mathbb{Z}} = \varprojlim_n \mathbb{Z}/n\mathbb{Z}$ and let $\widehat{\ }$ denote tensor with $\widehat{\mathbb{Z}}$ over $\mathbb{Z}$. Consider the double coset

$$X(\mathbb{C}) = B^\times \backslash (\mathcal{H}^{\pm} \times \widehat{B}^\times / \widehat{\mathcal{O}}^\times),$$

where $B^\times$ acts on $\widehat{B}^\times / \widehat{\mathcal{O}}^\times$ by left multiplication via the diagonal embedding. Then $X(\mathbb{C})$ has the structure of a complex analytic space [3] which fails to be compact if and only if $B \cong \mathrm{M}_2(\mathbb{Q})$, corresponding to the classical case of elliptic modular forms—higher class number issues do not arise in this case, so from now we assume that B is a division ring.

We again write $S_k^B(\mathfrak{N})$ for the finite-dimensional $\mathbb{C}$-vector space of quaternionic modular forms of weight k and level $\mathfrak{N}$: here, again roughly speaking, a quaternionic modular form of weight $k \in (2\mathbb{Z}_{>0})^n$ and level $\mathfrak{N}$ for B is an analytic function

$$f : \mathcal{H}^{\pm} \times \widehat{B}^\times / \widehat{\mathcal{O}}^\times \to W_k(\mathbb{C})$$

which is invariant under the weight k action of $B^\times$, with $W_k(\mathbb{C})$ as in Section 1.

2.2 Decomposing the Double Coset Space

By Eichler's theorem of norms, we have $\mathrm{nrd}(B^\times) = F^\times_{(+)}$ where

$$F^\times_{(+)} = \{a \in F^\times : v_i(a) > 0 \text{ for } i = 2, \dots, n\}$$

is the subgroup of elements of F which are positive at all real places which are ramified in B. In particular, $B^\times/B^\times_+ \cong \mathbb{Z}/2\mathbb{Z}$, where

$$B^\times_+ = \{\gamma \in B^\times : v_1(\mathrm{nrd}(\gamma)) > 0\} = \{\gamma \in B : \mathrm{nrd}(\gamma) \in F^\times_+\}.$$

The group $B^\times_+$ acts on the upper half-plane $\mathcal{H}$, therefore we may identify

$$X(\mathbb{C}) = B^\times_+ \backslash (\mathcal{H} \times \widehat{B}^\times / \widehat{\mathcal{O}}^\times).$$

Now we have a natural (continuous) projection map

$$X(\mathbb{C}) \to B^\times_+ \backslash \widehat{B}^\times / \widehat{\mathcal{O}}^\times,$$

and by strong approximation [15, Theorème III.4.3] the reduced norm gives a bijection

$$\mathrm{nrd} : B^\times_+ \backslash \widehat{B}^\times / \widehat{\mathcal{O}}^\times \xrightarrow{\sim} F^\times_+ \backslash \widehat{F}^\times / \widehat{\mathbb{Z}}^\times_F \cong \mathrm{Cl}^+ \mathbb{Z}_F, \tag{7}$$

where $\mathrm{Cl}^+ \mathbb{Z}_F$ denotes the strict class group of $\mathbb{Z}_F$, i.e. the ray class group of $\mathbb{Z}_F$ with modulus equal to the product of all real (infinite) places of F.

The space $X(\mathbb{C})$ is therefore the disjoint union of Riemann surfaces indexed by $\mathrm{Cl}^+ \mathbb{Z}_F$, which we identify explicitly as follows. Let the ideals $\mathfrak{b} \subset \mathbb{Z}_F$ form a set of representatives for $\mathrm{Cl}^+ \mathbb{Z}_F$, and let $\widehat{b} \in \widehat{\mathbb{Z}}_F$ be such that $\widehat{b}\widehat{\mathbb{Z}}_F \cap \mathbb{Z}_F = \mathfrak{b}$. For expositional simplicity, choose $\mathfrak{b} = \mathbb{Z}_F$ and $\widehat{\beta} = \widehat{1}$ for the representatives of the trivial class. By strong approximation (7), there exists $\widehat{\beta} \in \widehat{B}^\times$ such that $\mathrm{nrd}(\widehat{\beta}) = \widehat{b}$. Therefore

$$X(\mathbb{C}) = \bigsqcup_{[\mathfrak{b}]} B^\times_+ (\mathcal{H} \times \widehat{\beta}\widehat{\mathcal{O}}^\times). \tag{8}$$

We have a map

$$B^\times_+ (\mathcal{H} \times \widehat{\beta}\widehat{\mathcal{O}}^\times) \to \mathcal{O}^\times_{\widehat{\beta},+} \backslash \mathcal{H}$$
$$(z, \widehat{\beta}\widehat{\mathcal{O}}^\times) \mapsto z$$

where $\mathcal{O}_{\widehat{\beta}} = \widehat{\beta}\widehat{\mathcal{O}}\widehat{\beta}^{-1} \cap B$ and $\mathcal{O}^\times_{\widehat{\beta},+} = \mathcal{O}^\times_{\widehat{\beta}} \cap B^\times_+$, so that $\mathcal{O}_{\widehat{1}} = \mathcal{O}$.

For each $\widehat{\beta}$, let $\Gamma_{\widehat{\beta}} = \iota_\infty\big(\mathcal{O}^\times_{\widehat{\beta},+}/\mathbb{Z}^\times_F\big) \subset \mathrm{PGL}_2(\mathbb{R})^+$. Then the Eichler-Shimura isomorphism on each component in (8) gives an identification of Hecke modules

$$S^B_k(\mathfrak{N}) \xrightarrow{\sim} \bigoplus_{\widehat{\beta}} H^1(\Gamma_{\widehat{\beta}}, V_k(\mathbb{C}))^+, \tag{9}$$

where $^+$ denotes the $+1$-eigenspace for complex conjugation. For each $\widehat{\beta}$, let $\mathcal{O}(1)_{\widehat{\beta}} = \widehat{\beta}\mathcal{O}(1)\widehat{\beta}^{-1} \cap B$ be the maximal order containing the Eichler order $\mathcal{O}_{\widehat{\beta}}$,

and let $\Gamma(1)_{\widehat{\beta}} = \iota_\infty(\mathcal{O}(1)^\times_{\widehat{\beta},+}/\mathbb{Z}_F^\times)$. Further, let $V_{\widehat{\beta}}(\mathbb{C}) = \mathrm{Coind}_{\Gamma_{\widehat{\beta}}}^{\Gamma(1)_{\widehat{\beta}}} V_k(\mathbb{C})$. Then Shapiro's lemma applied to each summand in (9) gives

$$S_k^B(\mathfrak{N}) \xrightarrow{\sim} \bigoplus_{\widehat{\beta}} H^1(\Gamma(1)_{\widehat{\beta}}, V_{\widehat{\beta}}(\mathbb{C}))^+. \tag{10}$$

2.3 Hecke Operators

In the description (10), the Hecke operators $T_\mathfrak{p}$ act on $\bigoplus_{\widehat{\beta}} H^1(\Gamma(1)_{\widehat{\beta}}, V_{\widehat{\beta}}(\mathbb{C}))$ in the following way. Let $\mathfrak{p}$ be a prime ideal of $\mathbb{Z}_F$ with $\mathfrak{p} \nmid \mathfrak{D}\mathfrak{N}$, and let $\widehat{p} \in \widehat{\mathbb{Z}}_F$ be such that $\widehat{p}\,\widehat{\mathbb{Z}}_F \cap \mathbb{Z}_F = \mathfrak{p}$. We consider the $\widehat{\beta}'$-summand in (10), corresponding to the ideal class $[\mathfrak{b}']$. Let $f : \Gamma(1)_{\widehat{\beta}'} \to V_{\widehat{\beta}'}(\mathbb{C})$ be a crossed homomorphism: we will then obtain a new crossed homomorphism $f \mid T_\mathfrak{p} : \Gamma(1)_{\widehat{\beta}} \to V_{\widehat{\beta}}(\mathbb{C})$, where $\widehat{\beta}$ corresponds to the ideal class of $[\mathfrak{p}\mathfrak{b}']$ among the explicit choices made above.

Let $\widehat{\varpi} \in \widehat{\mathcal{O}}_{\widehat{\beta}}$ be such that $\mathrm{nrd}(\widehat{\varpi}) = \widehat{p}$. Then there are elements $\widehat{\gamma}_a \in \widehat{\mathcal{O}}_{\widehat{\beta}}$, indexed by $a \in \mathbb{P}^1(\mathbb{F}_\mathfrak{p})$, such that

$$\widehat{\mathcal{O}}^\times_{\widehat{\beta}} \widehat{\varpi} \widehat{\mathcal{O}}^\times_{\widehat{\beta}} = \bigsqcup_{a \in \mathbb{P}^1(\mathbb{F}_\mathfrak{p})} \widehat{\mathcal{O}}^\times_{\widehat{\beta}} \widehat{\alpha}_a \tag{11}$$

where $\widehat{\alpha}_a = \widehat{\varpi}\widehat{\gamma}_a$.

Let $\gamma \in \Gamma_{\widehat{\beta}}$. Extending (11) to $\widehat{\mathcal{O}}(1)^\times_{\widehat{\beta}}$, we conclude that there exist unique elements $\widehat{\delta}_a \in \widehat{\mathcal{O}}(1)^\times_{\widehat{\beta}}$ and a unique permutation γ^* of $\mathbb{P}^1(\mathbb{F}_\mathfrak{p})$ such that

$$\widehat{\alpha}_a \gamma = \widehat{\delta}_a \widehat{\alpha}_{\gamma^* a}$$

for $a \in \mathbb{P}^1(\mathbb{F}_\mathfrak{p})$. Thus we have

$$(\widehat{\beta}'\widehat{\beta}^{-1}\widehat{\alpha}_a)\gamma = (\widehat{\beta}'\widehat{\beta}^{-1})\widehat{\delta}_a \widehat{\alpha}_{\gamma^* a} = \widehat{\delta}'_a (\widehat{\beta}'\widehat{\beta}^{-1}\widehat{\alpha}_{\gamma^* a}).$$

where $\widehat{\delta}'_a = (\widehat{\beta}'\widehat{\beta}^{-1})\widehat{\delta}_a(\widehat{\beta}'\widehat{\beta}^{-1})^{-1}$.

Recall that $\widehat{\beta}'\widehat{\mathcal{O}}$ has left order $\widehat{\mathcal{O}}_{\widehat{\beta}'}$ and similarly $\widehat{\mathcal{O}}\widehat{\beta}^{-1}$ has right order $\widehat{\mathcal{O}}_{\widehat{\beta}}$. Therefore, we may consider the left $\widehat{\mathcal{O}}_{\widehat{\beta}'}$-ideal

$$\widehat{\mathcal{O}}_{\widehat{\beta}'}\widehat{\beta}'\widehat{\mathcal{O}}\widehat{\beta}^{-1}\widehat{\mathcal{O}}_{\widehat{\beta}}\widehat{\alpha}_a \tag{12}$$

noting that the left and right orders in each case match up, so the product is compatible. Next, recall that the elements $\widehat{\beta}'$, $\widehat{\beta}$, $\widehat{\varpi}$ have reduced norms corresponding to the ideal classes $[\mathfrak{b}']$, $[\mathfrak{p}\mathfrak{b}']$, and $[\mathfrak{p}]$, respectively. Thus the reduced norm of the left ideal (12) has a trivial ideal class. Therefore, by strong approximation (applied now to left ideals of the order $\mathcal{O}_{\widehat{\beta}'}$), for each $a \in \mathbb{P}^1(\mathbb{F}_\mathfrak{p})$, there exist elements $\pi'_a \in \mathcal{O}_{\widehat{\beta}'} \cap B^\times_+$ such that

$$\widehat{\mathcal{O}}_{\widehat{\beta}'}\widehat{\beta}'\widehat{\beta}^{-1}\widehat{\alpha}_a \cap B = \mathcal{O}_{\widehat{\beta}'}\pi'_a.$$

Hence there exists a unique permutation γ^* of $\mathbb{P}^1(\mathbb{F}_\mathfrak{p})$ such that

$$\pi'_a \gamma = \delta'_a \pi'_{\gamma^* a}$$

with $\delta_a \in \mathcal{O}^\times_{\widehat{\beta}',+}$. The new crossed homomorphism $f \mid T_\mathfrak{p} : \Gamma_{\widehat{\beta}} \to V_{\widehat{\beta}}(\mathbb{C})$ is then defined by the formula

$$(f \mid T_\mathfrak{p})(\gamma) = \sum_{a \in \mathbb{P}^1(\mathbb{F}_\mathfrak{p})} f(\delta'_a)^{\pi'_a}$$

for $\gamma \in \Gamma_{\widehat{\beta}}$.

2.4 Complex Conjugation and Atkin-Lehner Involutions

We now define an operator W_∞ which acts by complex conjugation. Let $\mathrm{Cl}^{(+)}\,\mathbb{Z}_F$ denote the ray class group of $\mathbb{Z}_F$ with modulus equal to the real (infinite) places of F which are ramified in B. Then we have a natural map $\mathrm{Cl}^+\,\mathbb{Z}_F \to \mathrm{Cl}^{(+)}\,\mathbb{Z}_F$; this map is an isomorphism if and only if there exists a unit $u \in \mathbb{Z}_F^\times$ which satisfies $v_1(u) < 0$ and $v_i(u) > 0$ for the other real places v_i $(i = 2, \ldots, n)$ of F, otherwise the kernel of this map is isomorphic to $\mathbb{Z}/2\mathbb{Z}$. Let $[\mathfrak{m}] \in \mathrm{Cl}^+\,\mathbb{Z}_F$ generate the kernel of this map.

Let $f : \Gamma(1)_{\widehat{\beta}'} \to V_{\widehat{\beta}'}$ be a crossed homomorphism, and let $\widehat{\beta}$ correspond to the ideal class $[\mathfrak{b}'\mathfrak{m}^{-1}]$; we will define the complex conjugate crossed homomorphism $(f \mid W_\infty) : \Gamma(1)_{\widehat{\beta}} \to V_{\widehat{\beta}}(\mathbb{C})$. The left $\mathcal{O}_{\widehat{\beta}'}$-ideal $\widehat{\mathcal{O}}_{\widehat{\beta}'}\widehat{\beta}'\widehat{\mathcal{O}}\widehat{\beta}^{-1} \cap B$ has reduced norm corresponding to the ideal class $[\mathfrak{m}] \in \mathrm{Cl}^+\,\mathbb{Z}_F$, so there exists a generator $\mu' \in \mathcal{O}_{\widehat{\beta}'}$ of this ideal such that $v_1(\mathrm{nrd}(\mu')) < 0$ but $v_i(\mathrm{nrd}(\mu')) > 0$ for $i = 2, \ldots, n$. Then given $\gamma \in \Gamma(1)_{\widehat{\beta}}$, we define

$$(f \mid W_\infty)(\gamma) = f(\mu'\gamma\mu'^{-1})^{\mu'}.$$

Finally, we define the Atkin-Lehner involutions $W_{\mathfrak{p}^e}$ for $\mathfrak{p}^e \parallel \mathfrak{D}\mathfrak{M}$. Let $\mathfrak{p}$ correspond to $\widehat{p} \in \widehat{\mathbb{Z}}_F$. Then there exists an element $\widehat{\pi} \in \mathcal{O}_{\widehat{\beta}}$ which generates the unique two-sided ideal of $\mathcal{O}_{\widehat{\beta}}$ of reduced norm generated by $\widehat{p}^e$. The element $\widehat{\pi}$ normalizes $\mathcal{O}_{\widehat{\beta}}$ and $\widehat{\pi}^2 \in \mathcal{O}^\times_{\widehat{\beta}}\widehat{F}^\times$. Let $\widehat{\beta}$ correspond to the ideal class $[\mathfrak{p}\mathfrak{b}']$. Then as above, by strong approximation there exists an element $\mu' \in \mathcal{O}_{\widehat{\beta}'} \cap B^\times_+$ such that $\mathcal{O}_{\widehat{\beta}'}\widehat{\beta}'\widehat{\beta}\widehat{\pi} \cap B = \mathcal{O}_{\widehat{\beta}'}\mu'$. Given $f : \Gamma(1)_{\widehat{\beta}'} \to V_{\widehat{\beta}'}$, we then define $(f \mid W_{\mathfrak{p}^e}) : \Gamma(1)_{\widehat{\beta}} \to V_{\widehat{\beta}}(\mathbb{C})$ by

$$(f \mid W_\mathfrak{p})(\gamma) = f(\mu'\gamma\mu'^{-1})^{\mu'}$$

for $\gamma \in \Gamma(1)_{\widehat{\beta}}$.

3 Algorithmic Methods

In this section, we take the adelic description of Section 2 and show how to compute with it explicitly, proving Theorem 1.

Our algorithm takes as input a totally real field F of degree $[F : \mathbb{Q}] = n$, a quaternion algebra B over F split at a unique real place, an ideal $\mathfrak{N} \subset \mathbb{Z}_F$ coprime to the discriminant $\mathfrak{D}$ of B, a vector $k \in (2\mathbb{Z}_{>0})^n$, and a prime $\mathfrak{p} \nmid \mathfrak{D}\mathfrak{N}$, and outputs the matrix of the Hecke operator $T_{\mathfrak{p}}$ acting on the space $H = \bigoplus_{\widehat{\beta}} H^1\big(\Gamma(1)_{\widehat{\beta}}, V_{\widehat{\beta}}(\mathbb{C})\big)^+$ (in the notation of Section 2) with respect to some fixed basis which does not depend on $\mathfrak{p}$. From these matrices, one decomposes the space H into Hecke-irreducible subspaces by the techniques of basic linear algebra.

Our algorithm follows the form given in the overview in Section 1, so we describe our algorithm in steps, with a description of each step along the way.

Step 1 (Compute a splitting field): Let $K \hookrightarrow \mathbb{C}$ be a Galois number field containing F which splits B: for example, we can take the normal closure of any quadratic field contained in B. Since all computations then occur inside $K \subset \mathbb{C}$, we may work then with coefficient modules over K using exact arithmetic. (This step is only necessary if k is not parallel weight 2, for otherwise the action of $B^\times$ factors through $K = \mathbb{Q}$.)

Step 2 (Compute ideal class representatives): Compute a set of representatives $[\mathfrak{b}]$ for the strict class group $\mathrm{Cl}^+ \mathbb{Z}_F$ with each $\mathfrak{b}$ coprime to $\mathfrak{p}\mathfrak{D}\mathfrak{N}$. (See Remark 4 below.)

Compute a maximal order $\mathcal{O}(1) \subset B$. For each representative ideal $\mathfrak{b}$, compute a right $\mathcal{O}(1)$-ideal $J_{\mathfrak{b}}$ such that $\mathrm{nrd}(J_{\mathfrak{b}}) = \mathfrak{b}$ and let $\mathcal{O}(1)_{\mathfrak{b}}$ be the left order of $J_{\mathfrak{b}}$. (In the notation of Section 2, the right $\mathcal{O}(1)$-ideals $J_{\mathfrak{b}}$ represent the elements $\widehat{\beta}$, and $\mathcal{O}(1)_{\mathfrak{b}} = \mathcal{O}(1)_{\widehat{\beta}}$.)

Step 3 (Compute presentations for the unit groups): Compute an embedding $\iota_\infty : B \hookrightarrow \mathrm{M}_2(\mathbb{R})$ corresponding to the unique split real place.

For each $\mathfrak{b}$, compute a finite presentation for $\Gamma(1)_{\mathfrak{b}} = \iota_\infty(\mathcal{O}(1)^\times_{\mathfrak{b},+}/\mathbb{Z}_F^\times)$ consisting of a (minimal) set of generators $G_{\mathfrak{b}}$ and relations $R_{\mathfrak{b}}$ together with a solution to the word problem for the computed presentation [16]. (Note that the algorithm stated therein [16, Theorem 3.2] is easily extended from units of reduced norm 1 to totally positive units.)

For efficiency, we start by computing such a presentation with generators G associated to the order $\mathcal{O}(1)$ and then for each order $\mathcal{O}(1)_{\mathfrak{b}}$ we begin with the elements in hand formed by short products of elements in G which happen to lie in $\mathcal{O}(1)_{\mathfrak{b}}$ (to aid in the search for units [16, Algorithm 3.2]; note that $\mathcal{O}(1) \cap \mathcal{O}(1)_{\mathfrak{b}}$ is an Eichler order of level $\mathfrak{b}$ in $\mathcal{O}(1)_{\mathfrak{b}}$).

Step 4 (Compute splitting data): Compute a splitting

$$\iota_{\mathfrak{N}} : \mathcal{O}(1) \hookrightarrow \mathcal{O}(1) \otimes_{\mathbb{Z}_F} \mathbb{Z}_{F,\mathfrak{N}} \cong \mathrm{M}_2(\mathbb{Z}_{F,\mathfrak{N}}).$$

Note that since $\mathfrak{b}$ is coprime to $\mathfrak{N}$, we have $\mathcal{O}(1) \otimes \mathbb{Z}_{F,\mathfrak{N}} = \mathcal{O}(1)_{\mathfrak{b}} \otimes \mathbb{Z}_{F,\mathfrak{N}}$ for all $\mathfrak{b}$, so $\iota_{\mathfrak{N}}$ also gives rise to a splitting for each $\mathcal{O}(1)_{\mathfrak{b}}$. For each $\mathfrak{b}$, compute the Eichler order $\mathcal{O}_{\mathfrak{b}} \subset \mathcal{O}(1)_{\mathfrak{b}}$ of level $\mathfrak{N}$ with respect to $\iota_{\mathfrak{N}}$.

Next, for each $\mathfrak{b}$, compute representatives for the left cosets of the group $\Gamma_{\mathfrak{b}} = \iota_\infty(\mathcal{O}^\times_{\mathfrak{b},+}/\mathbb{Z}_F^\times)$ inside $\Gamma(1)_{\mathfrak{b}}$ [8, Algorithm 6.1]. Finally, identify

$$V(K)_{\mathfrak{b}} = \mathrm{Coind}_{\Gamma_{\mathfrak{b}}}^{\Gamma(1)_{\mathfrak{b}}} V_k(K)$$

as a K-vector space given by copies of $V_k(K)$ indexed by these cosets, and compute the permutation action of the representatives of these cosets on this space.

In practice, it is more efficient to identify the above coset representatives with elements of $\mathbb{P}^1(\mathbb{Z}_F/\mathrm{N})$ and thereby work directly with the coefficient module $V(K)_{\mathfrak{b}} \cong K[\mathbb{P}^1(\mathbb{Z}_F/\mathfrak{N})] \otimes V_k(K)$.

Step 5 (Compute a basis for cohomology): Identify the space of crossed homomorphisms $\bigoplus_{\mathfrak{b}} Z^1(\Gamma(1)_{\mathfrak{b}}, V(K)_{\mathfrak{b}})$ with its image under the inclusion

$$\begin{aligned} Z^1(\Gamma_{\mathfrak{b}}, V(K)_{\mathfrak{b}}) &\to \bigoplus_{g \in G_{\mathfrak{b}}} V(K)_{\mathfrak{b}} \\ f &\mapsto (f(g))_{g \in G_{\mathfrak{b}}} \end{aligned}$$

consisting of those $f \in \bigoplus_{g \in G_{\mathfrak{b}}} V(K)_{\mathfrak{b}}$ which satisfy the relations $f(r) = 0$ for $r \in R_{\mathfrak{b}}$. Compute the space of principal crossed homomorphisms $B^1(\Gamma(1)_{\mathfrak{b}}, V(K)_{\mathfrak{b}})$ in a similar way, and thereby compute using linear algebra a K-basis for the quotient $H^1(\Gamma(1)_{\mathfrak{b}}, V(K)_{\mathfrak{b}}) = Z^1(\Gamma(1)_{\mathfrak{b}}, V(K)_{\mathfrak{b}})/B^1(\Gamma(1)_{\mathfrak{b}}, V(K)_{\mathfrak{b}})$ for each $\mathfrak{b}$.

Let $H = \bigoplus_{\mathfrak{b}} H^1(\Gamma(1)_{\mathfrak{b}}, V(K)_{\mathfrak{b}})$.

Step 6 (Compute representatives for left ideal classes): Compute a splitting $\iota_{\mathfrak{p}} : \mathcal{O}(1) \hookrightarrow \mathrm{M}_2(\mathbb{Z}_{F,\mathfrak{p}})$. For each ideal $\mathfrak{b}'$, perform the following steps.

First, compute the ideal $\mathfrak{b}$ with ideal class $[\mathfrak{b}] = [\mathfrak{p}\mathfrak{b}']$. Compute the left ideals

$$I_a = \mathcal{O}\iota_{\mathfrak{p}}^{-1} \begin{pmatrix} x & y \\ 0 & 0 \end{pmatrix} + \mathcal{O}\mathfrak{p}$$

indexed by the elements $a = (x : y) \in \mathbb{P}^1(\mathbb{F}_{\mathfrak{p}})$ and then compute the left $\mathcal{O}_{\mathfrak{b}'}$-ideals $I'_a = J_{\mathfrak{b}'}\overline{J}_{\mathfrak{b}} I_a$.

Compute totally positive generators $\pi'_a \in \mathcal{O}_{\mathfrak{b}'} \cap B_+^\times$ for $\mathcal{O}_{\mathfrak{b}'}\pi'_a = I'_a$ [12].

Now, for each $\gamma \in G_{\mathfrak{b}}$, compute the permutation γ^* of $\mathbb{P}^1(\mathbb{F}_{\mathfrak{p}})$ [8, Algorithm 5.8] and then the elements $\delta'_a = \pi'_a \gamma \pi'^{-1}_{\gamma^* a}$ for $a \in \mathbb{P}^1(\mathbb{F}_{\mathfrak{p}})$; write each such element δ'_a as a word in $G'_{\mathfrak{b}}$ and from the formula

$$(f \mid T_{\mathfrak{p}})(\gamma) = \sum_{a \in \mathbb{P}^1(\mathbb{F}_{\mathfrak{p}})} f(\delta'_a)^{\pi'_a}$$

with f in a basis for the $\mathfrak{b}'$-component of cohomology as in Step 5 compute the induced crossed homomorphism $f \mid T_{\mathfrak{p}}$ in the $\mathfrak{b}$-component.

Step 7 (Compute the blocks of the intermediate matrix): Assemble the matrix T with rows and columns indexed as in Step 5 with blocks in the $(\mathfrak{b}, \mathfrak{b}')$ position given by the output of Step 6: this matrix describes the action of $T_{\mathfrak{p}}$ on H.

Step 8 (Decompose H into $\pm$-eigenspaces for complex conjugation): Determine the representative ideal $\mathfrak{m}$ (among the ideals $\mathfrak{b}$) which generates the kernel of the map $\mathrm{Cl}^+ \mathbb{Z}_F \to \mathrm{Cl}^{(+)} \mathbb{Z}_F$.

For each ideal $\mathfrak{b}'$, perform the following steps. Compute the ideal $\mathfrak{b}$ such that $[\mathfrak{b}] = [\mathfrak{b}'\mathfrak{m}^{-1}]$, and compute a generator μ' with $\mathcal{O}_{\mathfrak{b}'}\mu' = J_{\mathfrak{b}'}\overline{J}_{\mathfrak{b}}$ such that $v(\mathrm{nrd}(\mu')) < 0$. For each $\gamma \in G_{\mathfrak{b}}$, from the formula

$$(f \mid W_\infty)(\gamma) = f(\mu'\gamma\mu'^{-1})^{\mu'},$$

for f in a basis for the $\mathfrak{b}'$-component of cohomology as in Step 5 compute the induced crossed homomorphism $f \mid T_{\mathfrak{p}}$ in the $\mathfrak{b}$-component.

Assemble the matrix with blocks in the $(\mathfrak{b}, \mathfrak{b}')$ position given by this output: this matrix describes the action of complex conjugation W_∞ on H. Compute a K-basis for the $+1$-eigenspace H^+ of H for W_∞. Finally, compute the matrix T^+ giving the action of $T_{\mathfrak{p}}$ restricted to H^+ and return T^+.

This completes the description of the algorithm.

In a similar way, one computes the Atkin-Lehner involutions, replacing Step 6 with the description given in Section 2.4, similar to the computation of complex conjugation in Step 8.

Remark 3. Note that Steps 1 through 3 do not depend on the prime $\mathfrak{p}$ nor the level $\mathfrak{N}$ and Steps 4, 5, and 8 do not depend on the prime $\mathfrak{p}$, so these may be precomputed for use in tabulation.

Remark 4. To arrange uniformly that the ideals $\mathfrak{b}$ representing the classes in $\mathrm{Cl}^+ \mathbb{Z}_F$ are coprime to the prime $\mathfrak{p}$ in advance for many primes $\mathfrak{p}$, one has several options. One possibility is to choose suitable ideals $\mathfrak{b}$ of large norm in advance. Another option is to make suitable modifications "on the fly": if $\mathfrak{p}$ is not coprime to $\mathfrak{b}$, we simply choose a different ideal $\mathfrak{c}$ coprime to $\mathfrak{p}$ with $[\mathfrak{b}] = [\mathfrak{c}]$, a new ideal $J_{\mathfrak{c}}$ with $\mathrm{nrd}(J_{\mathfrak{c}}) = \mathfrak{c}$, and compute an element $\nu \in \mathcal{O}_{\mathfrak{b}}$ such that $\nu\mathcal{O}_{\mathfrak{b}}\nu^{-1} = \mathcal{O}_{\mathfrak{c}}$. Conjugating by ν where necessary, one can then transport the computations from one order to the other so no additional computations need to take place.

4 Examples

In this section, we compute with two examples to demonstrate the algorithm outlined in Section 3. Throughout, we use the computer system Magma [1].

Our first and most detailed example is concerned with the smallest totally real cubic field F with the property that the dimension of the space of Hilbert cusp forms of parallel weight 2 and level (1) is greater than zero and the strict class number of F is equal to 2. This field is given by $F = \mathbb{Q}(w)$ where w satisfies the equation $f(w) = w^3 - 11w - 11 = 0$. The discriminant of F is equal to $2057 = 11^2 17$, and $\mathbb{Z}_F = \mathbb{Z}[w]$. The roots of f in $\mathbb{R}$ are $-2.602\ldots$, $-1.131\ldots$, and $3.73\ldots$, and we label the real places v_1, v_2, v_3 of F into $\mathbb{R}$ according to this ordering.

We define the *sign* of $a \in F$ to be the triple $\mathrm{sgn}(a) = (\mathrm{sgn}(v_i(a)))_{i=1}^3 \in \{\pm 1\}^3$. The unit group of F is generated by the elements -1, $w + 1$ with $\mathrm{sgn}(w + 1) = (1, -1, -1)$, and the totally positive unit $-w^2 + 2w + 12$.

We begin by finding a quaternion algebra B with $\mathfrak{D} = \mathbb{Z}_F$ which is ramified at all but one real place [8, Algorithm 4.1]. We find the algebra $B = \left(\frac{w+1,-1}{F}\right)$ ramified only at v_1 and v_2, generated by i, j subject to $i^2 = w+1$, $j^2 = -1$, and $ji = -ij$.

For forms of parallel weight 2, **Step 1** is trivial: we can take $K = \mathbb{Q}$.

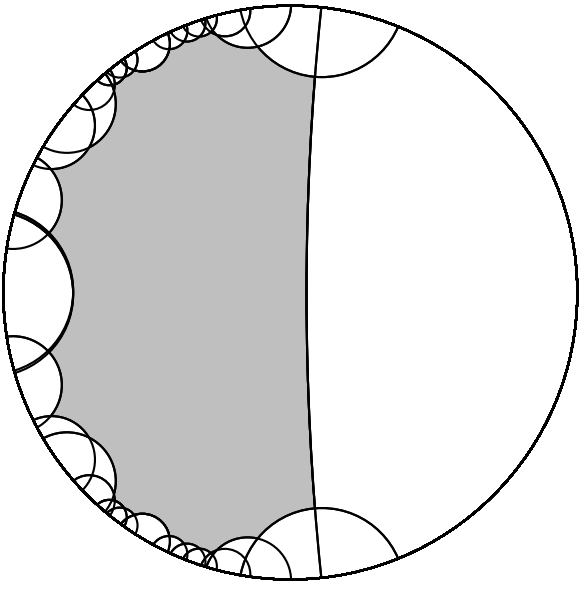
Next, in **Step 2** we compute ideal class representatives. The nontrivial class in $\mathrm{Cl}^+(\mathbb{Z}_F)$ is represented by the ideal $\mathfrak{b} = (w^2 - 2w - 6)\mathbb{Z}_F$, which is principal but does not possess a totally positive generator, since $\mathrm{sgn}(-w^2 + 2w + 6) = (-1, 1, -1)$ and there is no unit of $\mathbb{Z}_F$ with this sign. We note that $\mathrm{N}(\mathfrak{b}) = 7$.

Next, we compute a maximal order $\mathcal{O} = \mathcal{O}(1)$; it is generated over $\mathbb{Z}_F$ by i and the element $k = (1 + (w^2+1)i + ij)/2$. Next, we find that the right $\mathcal{O}$-ideal $J_\mathfrak{b}$ generated by $w^2 - 2w - 6$ and the element $(5 + (w^2+5)i + ij)/2 = 2 + 2i + k$ has $\mathrm{nrd}(J_\mathfrak{b}) = \mathfrak{b}$.

Next, in **Step 3** we compute presentations for the unit groups. We take the splitting

$$\begin{aligned} B &\hookrightarrow \mathrm{M}_2(\mathbb{R}) \\ i, j &\mapsto \begin{pmatrix} s & 0 \\ 0 & -s \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \end{aligned}$$

where $s = \sqrt{v_3(w+1)}$. We then compute a fundamental domain for $\Gamma = \Gamma(1)$ [16], given below.



We find that $\Gamma = \Gamma(1)$ is the free group on the generators $\alpha, \beta, \gamma_1, \ldots, \gamma_7$ subject to the relations

$$\gamma_1^2 = \gamma_2^2 = \gamma_3^3 = \gamma_4^2 = \gamma_5^3 = \gamma_6^2 = \gamma_7^2 = \alpha\beta\alpha^{-1}\beta^{-1}\gamma_1 \cdots \gamma_7 = 1.$$

For example, we have

$$2\alpha = (w^2 - 14) + (2w^2 - 4w - 13)i + (-2w^2 + 5w + 9)j + (-4w^2 + 8w + 26)ij.$$

The groups Γ and $\Gamma_{\mathfrak{b}}$ have isomorphic presentations. In particular, we note that both Γ and $\Gamma_{\mathfrak{b}}$ have genus 1, so we conclude that $\dim S_2(1) = 1 + 1 = 2$.

We illustrate the computation of Hecke operators with the primes $\mathfrak{p}_3 = (w + 2)\mathbb{Z}_F$ of norm 3 and $\mathfrak{p}_5 = (w + 3)\mathbb{Z}_F$ of norm 5. Note that $\mathfrak{p}_3$ is nontrivial in $\mathrm{Cl}^+(\mathbb{Z}_F)$ whereas $\mathfrak{p}_5$ is trivial.

Step **Step 4** requires no work, since we work with forms of level (1). In **Step 5** we compute with a basis for cohomology, and here we see directly that

$$H^1(\Gamma, \mathbb{Q}) \cong \mathrm{Hom}(\Gamma, \mathbb{Q}) \cong \mathbb{Z}f_\alpha \oplus \mathbb{Z}f_\beta$$

where f_α, f_β are the characteristic functions for α and β. We have a similar description for $H^1(\Gamma_{\mathfrak{b}}, \mathbb{Q})$.

Next, in **Step 6** we compute representatives of the left ideal classes. For $\mathfrak{p}_3$, for example, for $I_{[1:0]} \subset \mathcal{O}$ we find that $J_{\mathfrak{b}} I_{[1:0]} = \mathcal{O}_{\mathfrak{b}}((w + 1) + i + ij)$ and for $I_{[1:1]} \subset \mathcal{O}_{\mathfrak{b}}$ we have $\overline{J_{\mathfrak{b}}} I_{[1:1]} = \mathcal{O}(w + 1 - i + ij)$; we thereby find elements π_a, π'_a for $a \in \mathbb{P}^1(\mathbb{F}_{\mathfrak{p}_3})$. For the generators $\gamma = \alpha, \beta$ of $\mathcal{O}$ and $\mathcal{O}_{\mathfrak{b}}$, we compute the permutations γ^* of $\mathbb{P}^1(\mathbb{F}_{\mathfrak{p}_3})$; we find for example that α^* is the identity and

$$\pi'_{[1:0]}\alpha = \delta'_{[1:0]}\pi'_{[1:0]}$$

with $\delta'_{[1:0]} \in \mathcal{O}_{\mathfrak{b}}$, namely,

$$\begin{aligned} 14\delta'_{[1:0]} = (7w^2 - 98) + (-23w^2 + 40w + 167)i + \\ (-25w^2 + 59w + 103)j + (-2w^2 + 5w + 20)ij. \end{aligned}$$

We then write $\delta'_{[1:0]}$ as a word in the generators for $\Gamma'_{\mathfrak{b}}$ of length 23. Repeating these steps (reducing a total of 64 units), we assemble the block matrix in **Step 7** as the matrix

$$T_{\mathfrak{p}_3} \mid H = \begin{pmatrix} 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \\ 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \end{pmatrix}.$$

In a similar way, we find that $T_{\mathfrak{p}_5}$ is the identity matrix.

Finally, in **Step 8** we compute the action of complex conjugation. Here we have simply $\mu = i$ (whereas $\mu_{\mathfrak{b}}$ is more complicated), and thereby compute that

$$W_\infty \mid H = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & -1 \end{pmatrix}.$$

We verify that W_∞ commutes with $T_{\mathfrak{p}_3}$ (and $T_{\mathfrak{p}_5}$). We conclude that $T_{\mathfrak{p}_3} \mid H^+ = \begin{pmatrix} 0 & 2 \\ 2 & 0 \end{pmatrix}$ and $T_{\mathfrak{p}_5} \mid H^+ = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.

We then diagonalize the space H^+, which breaks up into two one-dimensional eigenforms f and g, and compute several more Hecke operators: we list in Table 1 below a generator for the prime $\mathfrak{p}$, its norm $\mathrm{N}\,\mathfrak{p}$, and the Hecke eigenvalues $a_{\mathfrak{p}}(f)$ and $a_{\mathfrak{p}}(g)$ for the cusp forms f, g.

Table 1. Hecke eigenvalues for the Hilbert cusp forms for $F = \mathbb{Q}(w)$ with $w^3 - 11w - 11 = 0$ of level (1) and parallel weight 2

$\mathfrak{p}$	$\mathrm{N}\mathfrak{p}$	$a_{\mathfrak{p}}(f)$	$a_{\mathfrak{p}}(g)$
$w+2$	3	2	-2
$w+3$	5	1	1
2	8	-5	-5
$2w+7$	9	-2	2
w	11	0	0
w^2-w-8	17	-5	5
$w-3$	17	-5	-5
$2w^2-5w-10$	23	2	-2
w^2-3w-2	25	-9	-9
w^2-6	29	9	-9
$w+4$	31	-2	-2
$2w^2-3w-16$	37	-3	3
w^2-2w-9	41	-5	5
w^2+w-3	49	-10	10

We note that the primes generated by w and $w-3$ are ramified in F.

By work of Deligne [3], the curves $X = X(1)$ and $X_{\mathfrak{b}}$ are defined over the strict class field F^+ of F, and $\mathrm{Gal}(F^+/F)$ permutes them. We compute that $F^+ = F(\sqrt{-3w^2+8w+12})$. Therefore the Jacobian J_f, corresponding to the cusp form f, is a modular elliptic curve over F^+ with $\#J(\mathbb{F}_{\mathfrak{p}}) = \mathrm{N}\,\mathfrak{p} + 1 - a_f(\mathfrak{p})$ with everywhere good reduction. The form g is visibly a quadratic twist of f by the character corresponding to the extension F^+/F.

Unfortunately, this curve does not have any apparent natural torsion structure which would easily allow for its identification as an explicit curve given by a sequence of coefficients [6, §4].

As a second and final example, we compute with a quaternion algebra defined over a quadratic field and therefore ramified at a finite prime. We take $F = \mathbb{Q}(\sqrt{65})$, with $\mathbb{Z}_F = \mathbb{Z}[(1+\sqrt{65})/2]$. The field F has $\#\,\mathrm{Cl}(F) = \#\,\mathrm{Cl}^+(F) = 2$. We compute the space $S = S_2(\mathfrak{p}_5)^{\mathfrak{p}_5\text{-new}}$ of Hilbert cuspidal new forms of parallel weight 2 and level $\mathfrak{p}_5$, where $\mathfrak{p}_5$ is the unique prime in $\mathbb{Z}_F$ of norm 5.

We compute that $\dim S = 10$, and that the space S decomposes into Hecke-irreducible subspaces of dimensions $2, 2, 3, 3$. For example, the characteristic polynomial of $T_{\mathfrak{p}_2}$ for $\mathfrak{p}_2$ either prime above 2 factors as

$$(T^2 - 2T - 1)(T^2 + 2T - 1)(T^6 + 11T^4 + 31T^2 + 9).$$

Remark 5. By the Jacquet-Langlands correspondence, the space $S_2(\mathfrak{p}_5)^{\mathfrak{p}_5\text{-new}}$ also occurs in the space of quaternionic modular forms for an Eichler order of

level $\mathfrak{p}_5$ in the definite quaternion algebra ramified at the the two real places of F and no finite place, and therefore is amenable to calculation by the work of Dembélé and Donnelly. We use this overlap to duplicate their computations (as well as ours) and thereby give some compelling evidence that the results are correct since they are computed in entirely different ways.

References

1. Bosma, W., Cannon, J., Playoust, C.: The Magma algebra system. I. The user language. J. Symbolic Comput. 24(3-4), 235–265 (1997)
2. Cremona, J.: The elliptic curve database for conductors to 130000. In: Hess, F., Pauli, S., Pohst, M. (eds.) ANTS 2006. LNCS, vol. 4076, pp. 11–29. Springer, Heidelberg (2006)
3. Deligne, P.: Travaux de Shimura. Séminaire Bourbaki, Lecture notes in Math. 244(389), 123–165
4. Dembélé, L.: Explicit computations of Hilbert modular forms on $\mathbb{Q}(\sqrt{5})$. Experiment. Math. 14(4), 457–466 (2005)
5. Dembélé, L.: Quaternionic Manin symbols, Brandt matrices and Hilbert modular forms. Math. Comp. 76(258), 1039–1057 (2007)
6. Dembélé, L., Donnelly, S.: Computing Hilbert modular forms over fields with nontrivial class group. In: van der Poorten, A.J., Stein, A. (eds.) ANTS-VIII 2008. LNCS, vol. 5011, pp. 371–386. Springer, Heidelberg (2008)
7. Donnelly, S., Voight, J.: Tables of Hilbert modular forms and elliptic curves over totally real fields (in preparation)
8. Greenberg, M., Voight, J.: Computing systems of Hecke eigenvalues associated to Hilbert modular forms. Math. Comp. (accepted)
9. Gunnells, P., Yasaki, D.: Hecke operators and Hilbert modular forms. In: van der Poorten, A.J., Stein, A. (eds.) ANTS-VIII 2008. LNCS, vol. 5011, pp. 387–401. Springer, Heidelberg (2008)
10. Hida, H.: On abelian varieties with complex multiplication as factors of the Jacobians of Shimura curves. American Journal of Mathematics 103(4), 727–776 (1981)
11. Hida, H.: Hilbert modular forms and Iwasawa theory. Clarendon Press, Oxford (2006)
12. Kirschmer, M., Voight, J.: Algorithmic enumeration of ideal classes for quaternion orders. SIAM J. Comput. (SICOMP) 39(5), 1714–1747 (2010)
13. Stein, W.A.: Modular forms database (2004), http://modular.math.washington.edu/Tables
14. Stein, W.A., Watkins, M.: A database of elliptic curves—first report. In: Fieker, C., Kohel, D.R. (eds.) ANTS 2002. LNCS, vol. 2369, pp. 267–275. Springer, Heidelberg (2002)
15. Vignéras, M.-F.: Arithmétique des algèbres de quaternions. LNM, vol. 800. Springer, Berlin (1980)
16. Voight, J.: Computing fundamental domains for cofinite Fuchsian groups. J. Théorie Nombres Bordeaux 21(2), 467–489 (2009)

Improved Primality Proving with Eisenstein Pseudocubes

Kjell Wooding and H.C. Williams

Institute for Security, Privacy and Information Assurance, University of Calgary,
2500 University Dr. NW, Calgary, Alberta, T2N 1N4, Canada
kjell@ispia.ca,
williams@math.ucalgary.ca

Abstract. In August 2002, Agrawal, Kayal, and Saxena described an unconditional, deterministic algorithm for proving the primality of an integer N. Though of immense theoretical interest, their technique, even incorporating the many improvements that have been proposed since its publication, remains somewhat slow for practical application. This paper describes a new, highly efficient method for certifying the primality of an integer $N \equiv 1 \pmod 3$, making use of quantities known as Eisenstein pseudocubes. This improves on previous attempts, including the peudosquare-based approach of Lukes *et al.*, and the pseudosquare improvement proposed by Berrizbeitia, *et al.*

1 Motivation

In [1], Lukes *et al.*, building on the ideas of Hall [2], Shanks [3, p. 414], and Selfridge and Weinberger [4], described a highly efficient method for proving the primality of an integer N using quantities known as *pseudosquares*. Their test requires a table of least pseudosquares, denoted $M_{2,x}$, of sufficient size to ensure that $N < M_{2,x}$. If such a table is available, their method certifies the primality of an integer N using only $(\log N)^{3+o(1)}$ operations.

In [5], Berrizbeitia *et al.* introduced a conjecturally more efficient test, relying on quantities they termed *pseudocubes*, denoted $M_{3,x}$. Though expected to outperform the pseudosquare-based method asymptotically, this test required a table of pseudocubes of suffient size to ensure that $N < M_{3,x}^{2/3}$. In [6], we provided numerical data to support the conjectured asymptotic improvement. In the same paper, however, we pointed out that it is unlikely we will obtain pseudocubes large enough to realize the theoretical gains. Recent results of Sorenson [7] further support both the asymptotic benefit and the practical limitations of this method.

In this paper, we propose an alternate definition of pseudocube — the Eisenstein pseudocube — with a conjectured growth rate better than that of the pseudosquares. Furthermore, we propose an algorithm for proving primality of integers $N \equiv 1 \pmod 3$ that eliminates the troublesome 2/3 exponent of Berrizbeitia's method. In the process, we supply numerical evidence to support the argument that, both asymptotically and practically, proving primality using

G. Hanrot, F. Morain, and E. Thomé (Eds.): ANTS-IX 2010, LNCS 6197, pp. 372–384, 2010.

Eisenstein pseudocubes will soon be more efficient than the pseudosquare test for primes $N \equiv 1 \pmod 3$.

2 Eisenstein Pseudocubes

Let ω be a primitive cube root of unity; *i.e.* $\omega = \frac{-1+\sqrt{3}i}{2}$, and consider the ring of Eisenstein integers, $\mathbb{Z}[\omega]$. Recall [8, Chap. 9] that $\mathbb{Z}[\omega]$ is a unique factorization domain with a norm given by $N(\alpha) = \alpha\overline{\alpha}$, and six units: $\pm 1, \pm\omega, \pm\omega^2$. There are three types of primes in $\mathbb{Z}[\omega]$: $(1-\omega)$, which lies over 3; the inert rational primes $q \equiv -1 \pmod 3$ with norm q^2; and the primes π of norm $\pi\overline{\pi} = p \equiv 1 \pmod 3$ where p is prime in $\mathbb{Z}$. We say that an element $\alpha \in \mathbb{Z}[\omega]$ is *primary* if $\alpha \equiv -1 \pmod 3$.[1] It is straightforward to show that every prime in $\mathbb{Z}[\omega]$ except $(1-\omega)$ has exactly one primary associate.

For any $\alpha, \pi \in \mathbb{Z}[\omega]$ with π prime, $N(\pi) \neq 3$, we can define the *cubic residue character* of α modulo π, denoted $\left(\frac{\alpha}{\pi}\right)_3$, as follows:

1. $\left(\frac{\alpha}{\pi}\right)_3 = 0$ if $\pi \mid \alpha$
2. $\left(\frac{\alpha}{\pi}\right)_3 \equiv \alpha^{(N(\pi)-1)/3} \pmod \pi$ otherwise, where $\left(\frac{\alpha}{\pi}\right)_3 \in \{1, \omega, \omega^2\}$.

The properties of this symbol are well-known. See, for example [8].

We can extend the notion of cubic residue character to include non-primes as follows. If $\alpha, \tau \in \mathbb{Z}[\omega]$ with $3 \nmid N(\tau)$, we define

$$\left(\frac{\alpha}{\tau}\right)_3 = \begin{cases} 1 \text{ if } \tau \text{ is a unit of } \mathbb{Z}[\omega], \\ \prod_{i=1}^{k} \left(\frac{\alpha}{\pi_i}\right)_3 \text{ otherwise} \end{cases}$$

where $\tau = \prod_{i=1}^{k} \pi_i$ and all $\pi_i \in \mathbb{Z}[\omega]$ are prime.

Finally, recall the Cubic Reciprocity Law (CRL), as it applies to to the cubic Jacobi symbol [5, §2.3]:

Theorem 1. *(Cubic Reciprocity) Let α, β be primary in $\mathbb{Z}[\omega]$ and of coprime norm $\neq 3$. Then $\left(\frac{\alpha}{\beta}\right)_3 = \left(\frac{\beta}{\alpha}\right)_3$.* □

We are now in a position to define an Eisenstein pseudocube.

Definition 1. *Let p be a fixed rational prime. Define $\mu_p = a + b\omega \in \mathbb{Z}[\omega]$, $a, b \in \mathbb{Z}$ to be an element of $\mathbb{Z}[\omega]$ of* minimal norm *such that:*

1. *μ_p is primary*
2. *$\gcd(a, b) = 1$*
3. *$\left(\frac{q}{\mu_p}\right)_3 = 1$ for all rational primes $q \in \mathbb{Z}$, $q \leq p$*
4. *μ_p not a cube in $\mathbb{Z}[\omega]$.*

We will call μ_p a minimal Eisenstein pseudocube *(or simply an* Eisenstein pseudocube*) for the prime p.*

[1] That is to say, if we write $\alpha = a + b\omega$, $a \equiv -1 \pmod 3$ and $3 \mid b$.

3 Congruence Criteria for Eisenstein Pseudocubes

One technique for efficiently computing a table of Eisenstein pseudocubes $\mu_p = x_p + y_p\omega$, is that of congruential sieving. In order to use this technique, we must first establish a set of acceptable residue conditions $\mathcal{S}_q$ on μ_p for each of the primes $q \leq p$ corresponding to the requirements of Definition 1. There are 3 cases to consider, one for each type of prime in the Eisenstein integers.

3.1 Case 1: $q \equiv -1 \pmod 3$

In this case, q is inert and primary. Since μ_p is by definition primary, we can invoke cubic reciprocity: $1 = \left(\frac{q}{\mu_p}\right)_3 = \left(\frac{\mu_p}{q}\right)_3$, and obtain the desired residue conditions by simply computing $\mu_p \equiv (m + n\omega)^3 \pmod q$ for all $0 \leq m, n < q$; *i.e.* the residue classes given by

$$\begin{aligned} x_p &\equiv m^3 - 3mn^2 + n^3 \pmod q \\ y_p &\equiv 3mn(m-n) \pmod q. \end{aligned}$$

There are

$$\frac{q^2 - 1}{3} \tag{1}$$

such solutions modulo q.

Example 1. The set of acceptable residues for Eisenstein pseudocubes modulo 5 is given by

$$\begin{aligned} \mathcal{S}_5 = \{&(1+0\omega), (2+0\omega), (3+0\omega), (4+0\omega), \\ &(3+1\omega), (1+2\omega), (4+3\omega), (2+4\omega)\}. \end{aligned}$$

3.2 Case 2: $q = 3$

Observe that $-3\omega = (1-\omega)^2$. By the bimultiplicity of the cubic Jacobi symbol,

$$\left(\frac{3}{\mu_p}\right)_3 = \left(\frac{\omega(1-\omega)}{\mu_p}\right)_3^2.$$

Write $\mu_p = x_p + y_p\omega = (-1)^{k-1}\prod_{i=1}^{k}\alpha_i$ where $\alpha_i = r_i + s_i\omega$ are primary primes; *i.e.* $3 \mid s_i$ and $r_i \equiv -1 \pmod 3$.

From the properties of the cubic Jacobi symbol, we know that $\left(\frac{1-\omega}{\alpha_i}\right)_3 = \omega^{\frac{2(r_i+1)}{3}}$, and $\left(\frac{\omega}{\alpha_i}\right)_3 = \omega^{\frac{r_i+1+s_i}{3}}$ giving

$$\left(\frac{1-\omega}{\mu_p}\right)_3 = \prod_{i=1}^{k}\omega^{\frac{2(r_i+1)}{3}} = \omega^{2\sum_{i=1}^{k}(r_i+1)/3},$$

$$\left(\frac{\omega}{\mu_p}\right)_3 = \prod_{i=1}^{k} \omega^{\frac{r_i+1+s_i}{3}} = \omega^{\sum_{i=1}^{k}(r_i+1)/3+\sum_{i=1}^{k} s_i/3},$$

and hence $\left(\frac{\omega(1-\omega)}{\mu_p}\right)_3 = \omega^{\sum_{i=1}^{k} s_i/3}$.

Thus

$$\left(\frac{3}{\mu_p}\right)_3 = \omega^{\frac{2}{3}\sum_{i=1}^{k} s_i}. \tag{2}$$

Lemma 1. *Let $\mu_p = x_p + y_p\omega = (-1)^{n-1}\prod_{i=1}^{n}\alpha_i$ where $\alpha_i = r_i + s_i\omega$ are primary primes. Then $x_p \equiv (-1)^{n-1}\prod_{i=1}^{n} r_i \pmod 9$ and $y_p \equiv \sum_{i=1}^{n} s_i \pmod 9$.*

Proof. If $n = 1$, the statement is trivially true.

Let $\alpha_j = r_j + s_j\omega, \alpha_k = r_k + s_k\omega$ be primary; *i.e.* $r_j \equiv r_k \equiv -1 \pmod 3$ and $s_j \equiv s_k \equiv 0 \pmod 3$. Writing $s_i = 3S_i$, $r_i = -1 + 3R_i$ for some $S_i, R_i \in \mathbb{Z}$, observe that

$$\begin{aligned} -(r_k + s_k\omega)(r_j + s_j\omega) &= -(r_k + 3S_k\omega)(r_j + 3S_j\omega) \\ &\equiv -r_k r_j - 3(S_j r_k + S_k r_j)\omega \\ &\equiv -r_k r_j - 3(-S_k + 3R_j S_k - S_j + 3R_k S_j)\omega \\ &\equiv -r_k r_j + (s_k + s_j)\omega \pmod 9 \end{aligned}$$

which is again primary. Thus, by induction, $(-1)^{n-1}\prod_{i=1}^{n}(r_i + s_i\omega) \equiv (-1)^{n-1}\prod_{i=1}^{n} r_i + \sum_{i=1}^{n} s_i\omega \pmod 9$, so writing $\mu_p = x_p + y_p\omega = (-1)^{n-1}\prod_{i=1}^{n}\alpha_i$ where $\alpha_i = r_i + s_i\omega$ are primary primes

$$\begin{aligned} x_p &\equiv (-1)^{n-1}\prod_{i=1}^{n} r_i \pmod 9, \\ y_p &\equiv \sum_{i=1}^{n} s_i \pmod 9 \end{aligned}$$

as desired. □

From Lemma 1, $y_p \equiv \sum_{i=1}^{k} s_i \pmod 9$, so $y_p/3 \equiv \sum_{i=1}^{k} s_i/3 \pmod 3$. Combining these facts with Equation 2, we obtain $\left(\frac{3}{\mu_p}\right)_3 = \omega^{\frac{2}{3}\sum_{i=1}^{k} s_i} = \omega^{2y_p/3}$. Clearly, $\left(\frac{3}{\mu_p}\right)_3 = 1 \iff 3 \mid \frac{y_p}{3}$ which, when combined with the requirement that μ_p be primary, gives the requisite congruence conditions:

$$\left(\frac{3}{\mu_p}\right)_3 = 1 \iff 9 \mid y_p \text{ and } x_p \equiv -1 \pmod 3.$$

Example 2. The set of acceptable residues for Eisenstein pseudocubes modulo 9 is given by

$$\mathcal{S}_9 = \{(2 + 0\omega), (5 + 0\omega), (8 + 0\omega)\}.$$

3.3 Case 3: $q \equiv 1 \pmod 3$

We can write $q = \pi_q \overline{\pi_q}$ where $\pi_q = a + b\omega$ and π_q is primary. Of course, $\overline{\pi_q}$ is also primary.

Lemma 2. *Let q be a rational prime, $\left(\frac{q}{\mu_p}\right)_3 = 1$ and $q = \pi_q\overline{\pi_q}$ with $\pi_q \in \mathbb{Z}[\omega]$ prime and primary, then $\left(\frac{q}{\mu_p}\right)_3 = 1$ if and only if $\left(\frac{\mu_p}{\pi_q}\right)_3 = \left(\frac{\overline{\mu_p}}{\pi_q}\right)_3$.*

Proof. Recall $q = \pi_q\overline{\pi_q}$, and that $\pi_q, \overline{\pi_q}$, and μ_p are all primary. From cubic reciprocity and the properties of the cubic Jacobi symbol [8, §9.3] we have that

$$\left(\frac{q}{\mu_p}\right)_3 = \left(\frac{\pi_q}{\mu_p}\right)_3 \left(\frac{\overline{\pi_q}}{\mu_p}\right)_3 = \left(\frac{\mu_p}{\pi_q}\right)_3 \left(\frac{\mu_p}{\overline{\pi_q}}\right)_3 = \left(\frac{\mu_p}{\pi_q}\right)_3 \overline{\left(\frac{\overline{\mu_p}}{\pi_q}\right)_3} = \left(\frac{\mu_p}{\pi_q}\right)_3 \left(\frac{\overline{\mu_p}}{\pi_q}\right)_3^{-1}$$

And thus it is clear that $\left(\frac{q}{\mu_p}\right)_3 = 1$ if and only if $\left(\frac{\mu_p}{\pi_q}\right)_3 = \left(\frac{\overline{\mu_p}}{\pi_q}\right)_3$. □

If $\left(\frac{q}{\mu_p}\right)_3 = 1$, then from Lemma 2 and the properties of the cubic reciprocity symbol, $\mu_p^{\frac{q-1}{3}} \equiv \overline{\mu_p}^{\frac{q-1}{3}} \pmod{\pi_q}$. By complex conjugation, we have also that $\left(\frac{\overline{\mu_p}}{\overline{\pi_q}}\right)_3 = \left(\frac{\mu_p}{\overline{\pi_q}}\right)_3$, and hence $\mu_p^{\frac{q-1}{3}} \equiv \overline{\mu_p}^{\frac{q-1}{3}} \pmod{\overline{\pi_q}}$. Combining these facts, we obtain

$$\left(\frac{q}{\mu_p}\right)_3 = 1 \iff \mu_p^{\frac{q-1}{3}} \equiv \overline{\mu_p}^{\frac{q-1}{3}} \pmod{q}. \tag{3}$$

Writing $\mu_p = x_p + y_p\omega$, we will now endeavour to reduce (3) to a set of congruence conditions on x_p and y_p. Note that when q is small, these congruence conditions can be computed by exhaustion. A more elegant algorithm, however, can be obtained from the theory of Lucas sequences.

First, observe that if $q \mid y_p$ then (3) reduces to the trivial $x_p \equiv x_p \pmod q$; *i.e.* $x + 0\omega \subset \mathcal{S}_q$ for $x = 1, \ldots, q-1$. For the remaining case, consider the recurring sequences $S_n(x,y), T_n(x,y) \in \mathbb{Z}[x,y]$ given by:[2]

$$\begin{aligned} S_1(x,y) &= x \\ T_1(x,y) &= y \\ S_n + T_n\omega &= (S_1 + T_1\omega)^n \end{aligned}$$

with $S_n, T_n \in \mathbb{Z}$. Clearly, we have also that $S_n + T_n\omega^2 = (S_1 + T_1\omega^2)^n$. By subtraction, $(\omega - \omega^2)T_n = (S_1 + T_1\omega)^n - (S_1 + T_1\omega^2)^n$, and thus writing $\alpha = \mu_p = x_p + y_p\omega$, $\beta = \overline{\mu_p} = x_p + y_p\omega^2$, we have

$$T_n = \frac{\alpha^n - \beta^n}{\omega - \omega^2}, \tag{4}$$

a recurrent sequence whose properties are described in [9]. We may parameterize this recurrence by writing $G = \alpha + \beta$, $H = \alpha\beta$, and observing that $T_n(G,H)$ is

[2] For simplicity, we will usually write S_n and T_n for $S_n(x,y)$ and $T_n(x,y)$, respectively.

given by the second-order recurrence: $T_{n+2} = GT_{n+1} - HT_n$. From (3), $\left(\frac{q}{\mu_p}\right)_3 = 1$ if and only if $q \mid (\alpha^{\frac{q-1}{3}} - \beta^{\frac{q-1}{3}})$ and hence from (4),

$$\left(\frac{q}{\mu_p}\right)_3 = 1 \iff q \mid T_{\frac{q-1}{3}}(G, H). \tag{5}$$

Since only the case $q \nmid y_p$ remains, we can rewrite (5) in terms of a single variable by defining $z_p \equiv x_p y_p^{-1} \pmod q$. Now $(x_p + y_p\omega)^{(q-1)/3} \equiv (x_p + y_p\omega^2)^{(q-1)/3} \pmod q$ if and only if $(z_p + \omega)^{(q-1)/3} \equiv (z_p + \omega^2)^{(q-1)/3} \pmod q$. Setting $\alpha = z_p + \omega$, $\beta = z_p + \omega^2$ in (4), we obtain

$$\left(\frac{q}{\mu_p}\right)_3 = 1 \iff q \mid T_{\frac{q-1}{3}}(G', H') \tag{6}$$

where $G' = 2z_p - 1$, $H' = z_p^2 - z_p + 1$. Since this relationship involves only one variable, we are in effect considering polynomials $T_n(x)$ where

$$T_0(x) = 0, \quad T_1(x) = 1$$
$$T_{n+1}(x) = (2x-1)T_n x - (x^2 - x + 1)T_{n-1}(x)$$

for a fixed $x \in \mathbb{Z}$. By induction, we see that $T_n(x)$ is a polynomial over $\mathbb{Z}$ with coefficients of degree $n-1$ and leading coefficient n.

In fact, $T_n(x) = U_n(G', H')$ where U_n is the Lucas function, $U_n = \frac{\alpha^n - \beta^n}{\alpha - \beta}$, $G' = \alpha + \beta = 2x - 1$, $H' = \alpha\beta = x^2 - x + 1$, and hence $\alpha = (x + \omega), \beta = (x + \omega^2)$. By drawing on the rich theory of Lucas functions, we can obtain both an efficient algorithm for computing the acceptable congruence conditions on $x_p, y_p \pmod q$, and the number of acceptable residues for the prime q.

To obtain the candidate solutions z_p satisfying (6), compute $T_{\frac{q-1}{3}}(x)$ for all $0 \le x < q$ by the method described in [3, §4.4], retaining solutions for which $T_{\frac{q-1}{3}}(x) \equiv 0 \pmod q$. Each z_p obtained in this fashion can then be used to produce $(q-1)$ acceptable values of μ_p by evaluating $x_p = 1, 2, \ldots, q-1$ and computing the corresponding $y_p = x_p z_p \pmod q$—a procedure illustrated in Example 3.

To obtain a count of these solutions, observe that in (6), we can write $\Delta = (\alpha - \beta)^2 = (2x-1)^2 - 4(x^2 - x + 1) = -3$. If q is a prime $\equiv 1 \pmod 3$ then $\epsilon = \left(\frac{\Delta}{q}\right) = 1$. Thus if $x \in \mathbb{Z}$ and $q \nmid x^2 - x + 1$ then $q \mid T_{q-\epsilon}(x)$ [3, Equation 4.3.3]. It follows that the polynomial $T_{q-1}(x)$ of degree $q-2$ has precisely $q-2$ distinct zeros modulo q. Now $T_{\frac{q-1}{3}}(x) \in \mathbb{Z}[x]$, and so it divides $T_{q-1}(x)$ as, from the theory of Lucas functions [3, Equation 4.2.45], we have

$$T_{3n}(2x-1, x^2 - x + 1) = 3T_n((x^2 - x + 1)^n - T_n^2).$$

It follows that $T_{\frac{q-1}{3}}(x)$ has exactly $\frac{q-1}{3} - 1$ distinct zeros modulo q.

By combining the cases when $q \nmid y_p$ and $q \mid y_p$, we see that there are

$$\left(\frac{q-1}{3} - 1\right)(q-1) + (q-1) = \frac{(q-1)^2}{3} \tag{7}$$

acceptable residues for a prime $q \equiv 1 \pmod 3$.

Example 3. Consider the case $q = 7$. We can derive the acceptable residue conditions on μ_p as follows.

If $q \mid y_p$, then $(x + 0\omega)$ is acceptable for $x = 1, \ldots, (q-1)$.

If $q \nmid y_p$ then from (6), we have that $\left(\frac{7}{\mu_p}\right)_3 = 1 \iff 7 \mid T_{\frac{7-1}{3}}(G', H') = T_2(G', H')$. Further, $T_2(G', H') = G'T_1(G', H') - HT_0(G', H') = G' - 0 = 2z_p - 1$ and hence,

$$\left(\frac{7}{\mu_p}\right)_3 = 1 \iff 7 \mid 2z_p - 1.$$

Thus, $z_p \equiv 4 \pmod 7$. Since we defined $z_p = x_p y_p^{-1} \pmod q$, $x_p \equiv 4y_p \pmod 7$, and we can obtain all solutions by running x_p through all nonzero residue classes (modulo 7) and computing $y_p \equiv 4^{-1}x_p \equiv 2x_p \pmod 7$; *i.e.*

x_p	1	2	3	4	5	6
$y_p \equiv 2x_p \pmod 7$	2	4	6	1	3	5

Combining these solutions with the trivial case ($q \mid y_p$), we obtain a complete set of solutions (modulo 7):

$$\begin{aligned}\mathcal{S}_7 = \{&(1 + 0\omega), (2 + 0\omega), (3 + 0\omega), (4 + 0\omega), (5 + 0\omega), (6 + 0\omega),\\ &(4 + 1\omega), (1 + 2\omega), (5 + 3\omega), (2 + 4\omega), (6 + 5\omega), (3 + 6\omega)\}.\end{aligned}$$

4 Eisenstein Pseudocubes and Primality Testing

Eisenstein pseudocubes may be employed to prove primality for integers $N \equiv 1 \pmod 3$ via the following theorem [10].

Theorem 2. *(Berrizbeitia, 2003, personal correspondence) Let $\nu = a + b\omega$ be a primary element of $\mathbb{Z}[\omega]$, where $\gcd(a, b) = 1$, ν is not a unit, prime, or perfect power in $\mathbb{Z}[\omega]$, and $N(\nu) < N(\mu_p)$. Then there must exist a rational prime $q \leq p$ such that $\left(\frac{q}{\nu}\right)_3 \not\equiv q^{(N(\nu)-1)/3} \pmod \nu$.* □

Recall that if $N \equiv 1 \pmod 3$ and N is a prime in $\mathbb{Z}$, then $N = \nu\overline{\nu}$, where ν is a primary prime in $\mathbb{Z}[\omega]$. Furthermore, if q is any rational prime, then

$$\left(\frac{q}{\nu}\right)_3 \equiv q^{\frac{N-1}{3}} \pmod \nu.$$

If we have a table of Eisenstein pseudocubes available to us, Berrizbeitia's result gives us a means to certify the primality of $N \equiv 1 \pmod 3$; *i.e.*

1. Test that N is not a perfect power; *e.g.* via [11].
2. Find a primary $\nu \in \mathbb{Z}[\omega]$ such that $N(\nu) = N$. This can be done efficiently using Cornacchia's algorithm [12, §1.5.2] via the method of Williams [13, §5]. If this step fails, then N is composite.[3]

[3] Cornacchia's algorithm requires the evaluation of a square root modulo N, and hence, usually requires a factorization of N. For our purposes, however, we simply assume that N is prime in this step. If Cornacchia fails, it is because N was composite, which is exactly what we set out to determine.

3. From a precomputed table of Eisenstein pseudocubes, choose $\mu_p \in \mathbb{Z}[\omega]$ of minimal norm such that $N < N(\mu_p)$.
4. For each prime $q \leq p$, test $\left(\frac{q}{\nu}\right)_3 \equiv q^{\frac{N-1}{3}} \pmod{\nu}$. If the test succeeds for all q, then N is prime.

Step 1 of this algorithm requires $(\log N)^{1+o(1)}$ operations. Cornacchia's algorithm (Step 2) essentially consists of a GCD computation ($(\log N)^{2+o(1)}$ operations), and the computation of a square root modulo a prime ($(\log N)^{3+o(1)}$). Step 3 is a merely a table lookup. Step 4 appears to be the most computationally intensive component of the algorithm, requiring a series of modular exponentiations (each requiring $(\log N)^{2+o(1)}$ operations). The precise number of exponentiations is dependent on the expected growth rate of the Eisenstein pseudocubes, something which we will now attempt to estimate.

5 Eisenstein Pseudocube Growth Rate

Let p_i denote the i^{th} prime ($p_1 = 2$), and let $\mathcal{S}_p$ denote the set of acceptable residues modulo p for the Eisenstein pseudocubes as developed in Section 3. Writing $p = p_n$, and denoting by (a, b) the Eisenstein integer $a + b\omega$, we know that

$$\mathcal{S}_2 = \{(1,0)\},$$
$$\mathcal{S}_9 = \{(2,0),(5,0),(8,0)\}, \text{ and}$$
$$\mathcal{S}_p = \left\{ (a,b) \in \mathbb{Z} \times \mathbb{Z} \,\middle|\, \left(\frac{p}{a+b\omega}\right)_3 = 1, \quad -\frac{p-1}{2} \leq a, b \leq \frac{p-1}{2} \right\} \text{ for } p > 3$$

Recall from Equations (1) and (7) that we expect

$$|\mathcal{S}_p| = \begin{cases} \dfrac{(p-1)^2}{3} & \text{if } p \equiv 1 \pmod 3 \\ \dfrac{(p^2-1)}{3} & \text{if } p \equiv 2 \pmod 3 \end{cases}$$

acceptable residues modulo p. Writing

$$S_1 = \prod_{p \equiv 1 \pmod 3} \frac{(p-1)^2}{3} \qquad H_1 = \prod_{p \equiv 1 \pmod 3} p$$
$$S_2 = \prod_{p \equiv 2 \pmod 3} \frac{(p^2-1)}{3} \qquad H_2 = \prod_{p \equiv 2 \pmod 3} p$$

for primes $p \leq p_n$, and invoking the Chinese Remainder Theorem we see that there are $S = 3S_1S_2$ solutions satisfying the congruence criteria of the Eisenstein pseudocubes in the region $-H/2 \leq a, b < H/2$, where $H = 9H_1H_2$.

Assume the S solutions $\mu = a + b\omega$ are equidistributed in the region $-H/2 \leq a, b < H/2$. By a similar argument to that of Lukes *et al.* [1], we expect the solution of *minimal norm*, denoted by μ_p, to be given by $a \approx b \approx \frac{H}{\sqrt{S}}$; *i.e.*

$$N(\mu_p) \approx \frac{H^2}{S}. \tag{8}$$

Consider the primes $p = p_n$ as $n \to \infty$. Making an assumption that the primes are distributed equally between $p \equiv 1 \pmod 3$ and $p \equiv 2 \pmod 3$, we can approximate H^2/S as follows. Write

$$\frac{H_1^2}{S_1} = \prod_{\substack{p \equiv 1 \pmod 3 \\ p \leq x}} \frac{3p^2}{(p-1)^2}, \text{ and} \tag{9}$$

$$\frac{H_2^2}{S_2} = \prod_{\substack{p \equiv 2 \pmod 3 \\ p \leq x}} \frac{3p^2}{(p^2-1)}. \tag{10}$$

From Mertens's Theorem [14, p. 351], $\prod_{p \leq x} \frac{1}{1-1/p} \sim \frac{e^{-\gamma}}{\log x}$ as $x \to \infty$, so (9) becomes

$$\begin{aligned} \frac{H_1^2}{S_1} &\approx 3^{\pi(x)/2} \prod_{\substack{p \equiv 1 \pmod 3 \\ p \leq x}} \left(\frac{p}{p-1}\right)^2 \\ &\approx 3^{\pi(x)/2} \prod_{p \leq x} \frac{p}{p-1} \\ &\sim e^{\gamma} 3^{\pi(x)/2} \log x. \end{aligned}$$

For (10), recall that $\prod_{p \leq x} \left(1 - \frac{1}{p^2}\right) = \zeta(2) = \frac{\pi^2}{6}$ as $x \to \infty$.[4] Hence

$$\frac{H_2^2}{S_2} \sim 3^{\pi(x)/2} \sqrt{\frac{6}{\pi^2}}$$

Putting these together, and writing $n = \pi(x)$, $c = \frac{27e^{\gamma}\sqrt{6}}{\pi}$, we obtain

$$N(\mu_{p_n}) \approx \frac{(9H_1H_2)^2}{3S_1S_2} \sim c3^n \log p_n$$

as $n \to \infty$. Thus, we expect $(\log N)^{1+o(1)}$ exponentiations in Step 4 of our primality proving algorithm, for a combined (randomized) complexity of $(\log N)^{3+o(1)}$ operations.[5]

[4] See, for example, [15, Theorem 1.4.1].

[5] The randomized nature of the algorithm stems solely from the requirement for a quadratic nonresidue in Cornacchia's algorithm. Finding this quadratic nonresidue requires, on average, two evaluations of a Jacobi symbol.

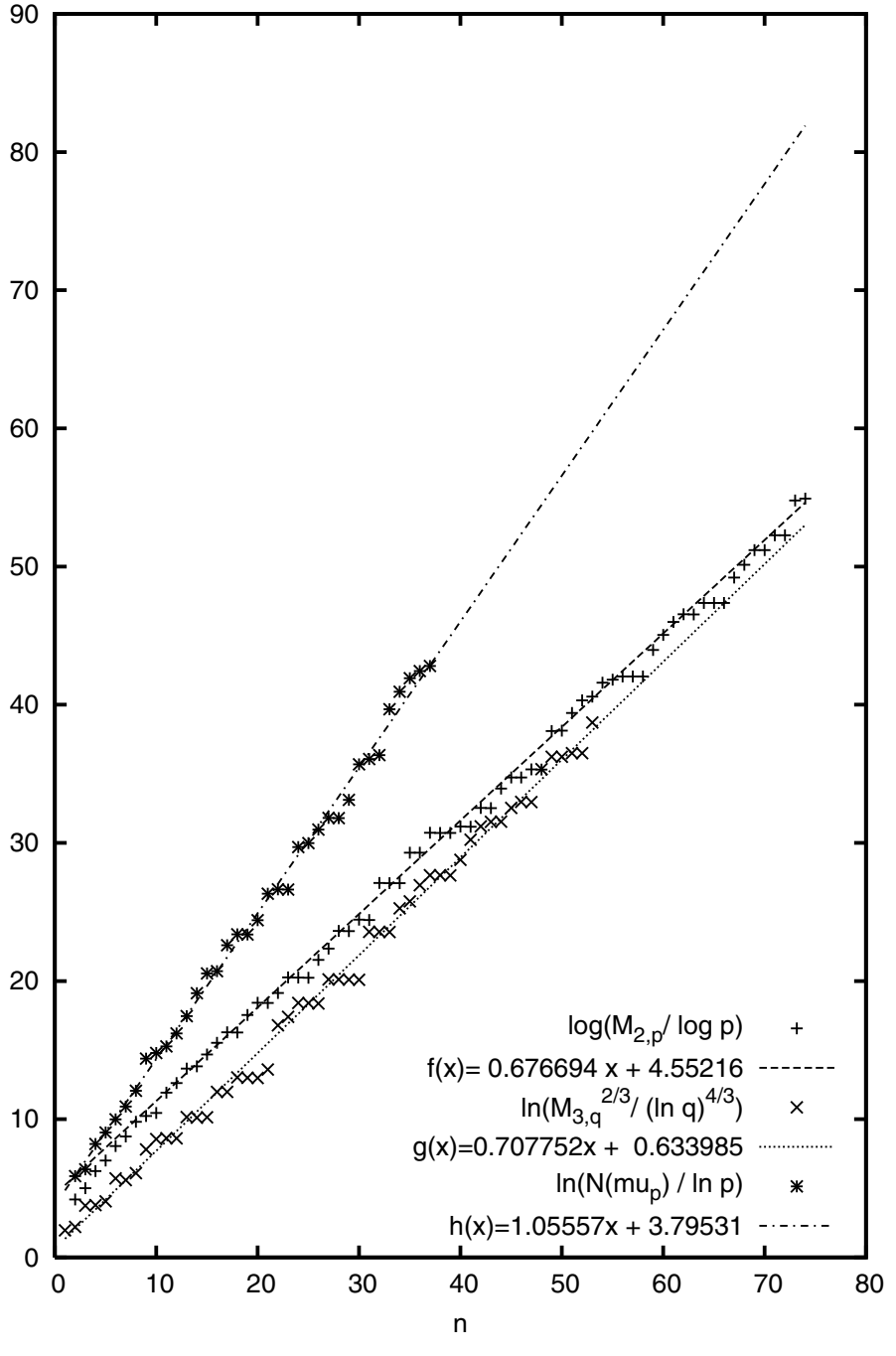


Fig. 1. Growth Rates

6 Experimental Results

Our experiment followed the same basic approach as [6]. To test our hypotheses, a table of Eisenstein pseudocubes was developed using the Calgary Scalable Sieve (CASSIE), a software toolkit for congruential sieving on the University of Calgary's Advanced Cryptography Laboratory (ACL) Beowulf cluster [6]. First, a series of small, non-normalized runs were performs in order to obtain Eisenstein pseudocubes for values of $p \leq 109$. Once these runs were completed, a large parallel job was executed. This larger job evaluated all candidate solutions with $N(\mu_p) \leq 2^{64}$. To parallelize this job, the 11520 acceptable residues formed by

Table 1. Eisenstein Pseudocube Results

p	$N(\mu_p)$	μ_p
18	247	$11 + 18\omega$
5	643	$29 + 18\omega$
7	5113	$71 + 72\omega$
11	13507	$23 + 126\omega$
13	39199	$227 + 90\omega$
17	1 07803	$-181 + 198\omega$
19	3 60007	$653 + 126\omega$
23	39 04969	$443 + 2160\omega$
29	61 07191	$-1669 + 1170\omega$
31	103 18249	$3617 + 2520\omega$
37	273 33067	$6023 + 3366\omega$
41	991 79467	$4973 + 11466\omega$
43	5329 97833	$-15451 + 11088\omega$
47	22785 22747	$54017 + 17514\omega$
53	27417 02809	$47477 + 56160\omega$
59	1 85007 66499	$66887 + 156510\omega$
61, 67	4 15475 53813	$235061 + 107172\omega$
71	11 94233 48797	$-139813 + 253764\omega$
73	82 46210 13649	$-267733 + 744120\omega$
79, 83	115 18103 60731	$1227419 + 761670\omega$
89	2507 90827 69801	$5052689 + 4961880\omega$
97	3393 26375 28481	$-2127709 + 4462200\omega$
101	9175 67688 29893	$10322861 + 8601732\omega$
103, 107	21408 90619 32079	$3056387 + 15918570\omega$
109	81221 66151 53761	$-27791551 + 1366560\omega$
113	10 70670 04348 13749	$109364777 + 13014540\omega$
127	15 84695 56547 47279	$-114717193 + 19952010\omega$
131	21 44850 97583 41459	$160585853 + 126202050\omega$
137	596 03669 06441 31739	$845355437 + 667764090\omega$
139	2127 62708 04110 19739	$-724036477 + 954969030\omega$
149	5736 34194 93471 77659	$696254903 + 2666049750\omega$
151	9708 82344 17235 68077	$2979509543 + 3236384556\omega$
157	14102 28178 31706 25921	$3671532959 + 3833807040\omega$

combining the solution candidates for moduli 18, 5, 7, and 11 were each used as a normalization modulus.[6] Each of these jobs required approximately 8000 CPU-seconds. Using 250 processing nodes, the complete job required approximately 4.25 days to complete, obtaining Eisenstein pseudocubes μ_p for $p \leq 157$. These results are summarized in Table 1.

7 Analysis and Conclusions

In Figure 1, Eisenstein pseudocube growth is shown as a function of n, where p_n is the n^{th} prime. The straight line represents the least squares line fitted to this data, and is given by:

$$y = 1.05557x + 3.79531$$

a result that is remarkably consistent with the slope predicted by the argument of Section 5; *i.e.* $\log 3 = 1.09861$. As a basis for comparison, classical pseudocube and pseudosquare results (including the recent work of Sorenson [7]) are also shown.

Two conclusions may be drawn from these results. First, even with the relatively modest amount of computing power used to compute our table of Eisenstein pseudocubes, we have already produced a test that is more efficient than the pseudocube method originally proposed by Berrizbeitia, *et al.* Second, we would expect that with a reasonable amount of computational investment, the Eisenstein pseudocube primality proving method will eventually be more efficient than existing methods involving the pseudosquares.

8 Summary

In this paper, we have adapted a theorem of Berrizbeitia to produce a highly efficient primality proving algorithm for integers $N \equiv 1 \pmod 3$, making use of quantities known as Eisenstein pseudocubes. In addition to theoretical contributions, we have compiled a table of these quantities using an extensive two-dimensional sieve calculation, and offered numerical evidence for a conjectured growth rate: $N(\mu_{p_n}) \sim c3^n \log p_n$ as $n \to \infty$.

References

1. Lukes, R.F., Patterson, C.D., Williams, H.C.: Some results on pseudosquares. Mathematics of Computation 65(213), S25–S27, 361–372 (1996)
2. Hall, M.: Quadratic residues in factorization. Bulletin of the American Mathematical Society 39, 758–763 (1933)

[6] The normalization optimization, first proposed by Lehmer in [16], is described in some detail in [6, §3.2].

3. Williams, H.C.: Édouard Lucas and Primality Testing. Canadian Mathematical Society Series of Monographs and Advanced Texts, vol. 22. Wiley Interscience, Hoboken (1998)
4. Williams, H.C.: Primality testing on a computer. Ars Combinatoria 5, 127–185 (1978)
5. Berrizbeitia, P., Müller, S., Williams, H.C.: Pseudocubes and primality testing. In: Buell, D.A. (ed.) ANTS 2004. LNCS, vol. 3076, pp. 102–116. Springer, Heidelberg (2004)
6. Wooding, K., Williams, H.C.: Doubly-focused enumeration of pseudosquares and pseudocubes. In: Hess, F., Pauli, S., Pohst, M. (eds.) ANTS 2006. LNCS, vol. 4076, pp. 208–221. Springer, Heidelberg (2006)
7. Sorenson, J.P.: Sieving for pseudosquares and pseudocubes in parallel using doubly-focused enumeration and wheel datastructures. In: Hanrot, G., Morain, F., Thomé, E. (eds.) ANTS-IX. LNCS, vol. 6197, pp. 331–339. Springer, Heidelberg (2010)
8. Ireland, K., Rosen, M.: A Classical Introduction to Modern Number Theory, 2nd edn. Graduate Texts in Mathematics, vol. 84. Springer, Heidelberg (1990)
9. Williams, H.C.: Some properties of a special set of recurring sequences. Pacific Journal of Mathematics 77(1), 273–285 (1978)
10. Wooding, K.: The Sieve Problem in One- and Two-Dimensions. PhD thesis, The University of Calgary, Calgary, AB (April 2010), http://math.ucalgary.ca/~hwilliam/files/wooding10thesis.pdf
11. Bernstein, D.J.: Detecting perfect powers in essentially linear time. Mathematics of Computation 67, 1253–1283 (1998)
12. Cohen, H.: A Course in Computational Algebraic Number Theory, 4th edn. Springer, Heidelberg (1993)
13. Williams, H.C.: An m^3 public-key encryption scheme. In: Williams, H.C. (ed.) CRYPTO 1985. LNCS, vol. 218, pp. 358–368. Springer, Heidelberg (1986)
14. Hardy, G.H., Wright, E.M.: An Introduction to the Theory of Numbers, 5th edn. Oxford University Press, Oxford (1979)
15. Crandall, R., Pomerance, C.: Prime numbers: A computational Perspective, 2nd edn. Springer, New York (2005)
16. Lehmer, D.H.: The sieve problem for all-purpose computers. Mathematical Tables and Other Aids to Computation 7(41), 6–14 (1953)

Hyperbolic Tessellations Associated to Bianchi Groups

Dan Yasaki

Department of Mathematics and Statistics
University of North Carolina at Greensboro, Greensboro, NC 27412, USA
d_yasaki@uncg.edu

Abstract. Let $F/\mathbb{Q}$ be a number field. The space of positive definite binary Hermitian forms over F form an open cone in a real vector space. There is a natural decomposition of this cone into subcones. In the case of an imaginary quadratic field these subcones descend to hyperbolic space to give rise to tessellations of 3-dimensional hyperbolic space by ideal polytopes. We compute the structure of these polytopes for a range of imaginary quadratic fields.

1 Introduction

Let $F/\mathbb{Q}$ be a number field. The space of positive definite binary Hermitian forms over F form an open cone in a real vector space. There is a natural decomposition of this cone into polyhedral cones corresponding to the facets of the Voronoï polyhedron [1,11,13]. This has been computationally explored for real quadratic fields in [16,12] and the cyclotomic field $\mathbb{Q}(\zeta_5)$ in [23].

For F an imaginary quadratic field, the polyhedral cones give rise to ideal polytopes in $\mathbb{H}_3$, 3-dimensional hyperbolic space. In work of Cremona and his students [6,7,5,14,22], analogous polytopes have already been computed for class number one imaginary quadratic fields as well as a few fields with class number two and three using different methods. The structure of the polytopes was used to compute Hecke operators on modular forms for the Bianchi groups over those fields. These polytopes were used by Goncharov [10] in his study of Euler complexes on modular curves. The data of the polytope and stabilizer could also be used to give explicit presentations of $\mathrm{GL}_2(\mathcal{O})$ using results of Macbeath and Weil [15,21]. Swan [20] has computed presentations of these groups, though not with the polytopes constructed here, for imaginary quadratic fields $\mathbb{Q}(\sqrt{d})$ for

$$-d \in \{1, 2, 3, 5, 6, 7, 11, 15, 19\}.$$

Such explicit presentations have been used to compute cohomology of Bianchi groups of small discriminant with non-trivial coefficients in work of Berkove, Sengun, and Finis-Grunewald-Tirao [2,3,9,19].

We remark that there are other ways to obtain the fundamental polytope data. Riley [18] wrote the first computer implementation of Poincaré's Polyhedron Theorem, which works in the more general setting of geometrically finite Kleinian

G. Hanrot, F. Morain, and E. Thomé (Eds.): ANTS-IX 2010, LNCS 6197, pp. 385–396, 2010.

groups. He computed the fundamental polytopes for many Bianchi groups. From this data, he computed presentations for the Bianchi groups and calculated the rank of their abelianizations. Another method is to use reduction theory. An algorithm of Swan [20] has been very recently implemented by Rahm and Fuchs [17], who used it to compute the integral homology groups of all Bianchi groups which are over imaginary quadratic fields of class number less than three.

In this paper, we investigate the structure of these ideal polytopes for a large range of imaginary quadratic fields. Our approach and implementation works for general imaginary quadratic fields, but we restrict the range to ease the computation. We compute the ideal polytope classes for all imaginary quadratic fields of class number one and two, as well as some fields of higher class number with small discriminant. Specifically, we compute the ideal polytopes for the fields $\mathbb{Q}(\sqrt{d})$ for square-free d, where

$$-d \in \{1, \cdots, 100, 115, 123, 163, 187, 235, 267, 403, 427\}.$$

There is no theoretical obstruction to computing these tessellations for higher class number and higher discriminant.

The structure of the paper is as follows. We set the notation for the quadratic fields and Hermitian forms in Section 2. The implementation is described in Section 3. Finally, in Section 4, we summarize some of the data collected so far. Finally, we describe a general result of Macbeath on computing group presentations for groups of homeomorphisms, illustrating one possible use of this data. We use this technique to give an explicit presentation for $\mathrm{GL}_2(\mathbb{Q}(\sqrt{-14}))$ in Section 5.

2 Notation and Background

Let $F = \mathbb{Q}(\sqrt{d}) \subset \mathbb{C}$ be an imaginary quadratic number field. We always take $d < 0$ to be a square-free integer. Let $\mathcal{O} \subset F$ denote the ring of integers in F. Then $\mathcal{O}$ has a $\mathbb{Z}$-basis consisting of 1 and ω, where

$$\omega = \begin{cases} \frac{1+\sqrt{d}}{2} & \text{if } d \equiv 1 \bmod 4, \\ \sqrt{d} & \text{if } d \equiv 2, 3 \bmod 4. \end{cases}$$

Let $\bar{\cdot}$ denote complex conjugation, the nontrivial Galois automorphism of F.

Definition 1. *A* binary Hermitian form over F *is a map* $\phi : F^2 \to \mathbb{Q}$ *of the form*

$$\phi(x, y) = ax\bar{x} + bx\bar{y} + \bar{b}\bar{x}y + cy\bar{y},$$

where $a, c \in \mathbb{Q}$ *and* $b \in F$ *such that* ϕ *is positive definite.*

By choosing a $\mathbb{Q}$-basis for F, ϕ can be viewed as a quadratic form over $\mathbb{Q}$. In particular, it follows that $\phi(\mathcal{O}^2)$ is discrete in $\mathbb{Q}$.

Definition 2. *The* minimum of ϕ *is*

$$m(\phi) = \inf_{v \in \mathcal{O}^2 \setminus \{0\}} \phi(v).$$

A vector $v \in \mathcal{O}^2$ *is* minimal vector *for* ϕ *if* $\phi(v) = m(\phi)$. *The set of minimal vectors for* ϕ *is denoted* $M(\phi)$.

Definition 3. *A Hermitian form over* F *is* perfect *if it is uniquely determined by* $M(\phi)$ *and* $m(\phi)$.

3 Implementation

3.1 Cone of Hermitian Forms and Hyperbolic Space

The space of positive definite binary Hermitian forms over F form an open cone in a real vector space. There is a natural decomposition of this cone into polyhedral cones corresponding to the facets of the Voronoï polyhedron Π [11, 13, 1]. The top-dimensional cones of this decomposition correspond to perfect forms and descend to ideal polytopes in $\mathbb{H}_3$, 3-dimensional hyperbolic space. Details are given below.

Let $\mathbf{G}$ be the restriction of scalars $\mathbf{G} = \mathrm{Res}_{F/\mathbb{Q}}(\mathrm{GL}_2)$. Then the group of rational points $\mathbf{G}(\mathbb{Q}) = \mathrm{GL}_2(F)$, and the group of real points is $G = \mathbf{G}(\mathbb{R}) \simeq \mathrm{GL}_2(\mathbb{C})$. Let $\mathbb{H}_3$ be hyperbolic 3-space:

$$\mathbb{H}_3 = \{(z,t) \ : \ z \in \mathbb{C}, \quad t \in \mathbb{R}_{>0}\}.$$

Then G acts on $\mathbb{H}_3$ by

$$\begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix} \cdot (z,t) = (z^*, t^*), \quad \text{where}$$

$$z^* = \frac{(\alpha z + \beta)(\overline{\gamma z + \delta}) + (\alpha t)(\overline{\gamma t})}{|\gamma z + \delta|^2 + |\gamma|^2 t^2} \quad \text{and} \quad t^* = \frac{|\alpha\delta - \beta\gamma| t}{|\gamma z + \delta|^2 + |\gamma|^2 t^2}$$

Note that diagonal matrices act trivially on $\mathbb{H}_3$, and the stabilizer of the point $(i,1)$ is $U(2)$. Thus one gets an identification between $\mathbb{H}_3$ and the coset space $\mathrm{GL}_2(\mathbb{C})/(U(2) \cdot \mathbb{R}_{>0})$.

A binary Hermitian form can be identified with the 4-dimensional real vector space V of Hermitian 2×2 matrices. The group $\mathrm{GL}_2(\mathbb{C})$ acts on this space via

$$g \cdot A = g A g^*$$

and preserves the open cone $C \subset V$ of positive definite Hermitian matrices, and the stabilizer of I is $U(2)$. Thus one has identification $C \simeq \mathrm{GL}_2(\mathbb{C})/U(2)$. Modding out by homotheties, one gets

$$C/\mathbb{R}_{>0} \simeq \mathbb{H}_3. \tag{1}$$

3.2 Voronoï Decomposition

There is a map q from $\mathcal{O}^2$ to the closure $\bar{C}$ of $C \subset V$ given by $q(v) = vv^*$. The Voronoï polyhedron Π is the unbounded polytope gotten by taking the convex hull of $\{q(v) : v \in \mathcal{O}^2 \setminus 0\}$. Taking cones over the facets of Π, one gets a decomposition of C into polyhedral cones known as the *Voronoï decomposition* of C. By (11), this decomposition descends to a tessellation of $\mathbb{H}_3$ by ideal polytopes. Note that the group $\Gamma = \mathbf{G}(\mathbb{Z}) = \mathrm{GL}_2(\mathcal{O})$ acts on C and preserves this decomposition.

3.3 Perfect Forms

A perfect form ϕ is uniquely determined by its minimum $m(\phi)$ and set of minimal vectors $M(\phi)$. By scaling, we can assume $m(\phi) = 1$. Since each minimal vector defines a linear equation in V, and V is 4-dimensional, generically 4 minimal vectors will uniquely determine ϕ. Note that this does not imply that $\#M(\phi) = 4$. Indeed in many examples, one has $M(\phi) > 4$.

There is a bijection between perfect forms over F and the facets of Π. Let P be a facet of Π with vertices $\{w_1, \ldots, w_k\}$. Then there is a unique form $\phi_P \in C$ such that $m(\phi_P) = 1$ and

$$\{q(v) : v \in M(\phi_P)\} = \{w_1, \ldots, w_k\}.$$

There is an algorithm [11] that uses this bijection to compute the $\mathrm{GL}_2(\mathcal{O})$-equivalency classes of perfect forms. The algorithm uses linear algebra and convex geometry, but requires an initial input of a perfect form. To this end, we describe the method that we used to compute an initial perfect form.

For each field $F = \mathbb{Q}(\sqrt{d})$, we need only to find a single perfect form to begin the algorithm. Thus we limit our search to a particular family of quadratic forms. Specifically, let $S_0 \subset C$ be the subset of quadratic forms ϕ such that

$$\left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\} \subseteq M(\phi).$$

For $\phi \in S_0$, the Hermitian matrix A_ϕ associated to ϕ must have the form

$$A_\phi = \begin{bmatrix} 1 & \beta \\ \bar{\beta} & 1 \end{bmatrix}, \quad \text{where } \beta \in F \text{ with } \mathrm{Re}(\beta) = -\frac{1}{2} \text{ and } |\beta| < 1.$$

If $\phi \in S_0$ and ϕ has an additional minimal vector $\begin{bmatrix} a \\ b \end{bmatrix} \in \mathcal{O}^2$, then

$$\beta = -\frac{1}{2} + \left(\frac{1 - {a_1}^2 + {a_2}^2 d + a_1 b_1 - a_2 d b_2 - {b_1}^2 + {b_2}^2 d}{2\,d a_1 b_2 - 2\,d a_2 b_1} \right) \sqrt{d}, \tag{2}$$

where $a = a_1 + a_2\sqrt{d}$ and $b = b_1 + b_2\sqrt{d}$. Combined with (2), this implies

$$-\frac{\left(1 - {a_1}^2 + {a_2}^2 d + a_1 b_1 - a_2 d b_2 - {b_1}^2 + {b_2}^2 d\right)^2 d}{\left(2\,d a_1 b_2 - 2\,d a_2 b_1\right)^2} < \frac{3}{4}. \tag{3}$$

Reduction theory, specifically the existence of Siegel sets, ensures that the values $\mathrm{N}_{F/\mathbb{Q}}(a)$, $\mathrm{N}_{F/\mathbb{Q}}(b)$, and $\mathrm{N}_{F/\mathbb{Q}}(b-a)$ for a solution are bounded above by a constant depending upon d. Thus we implement a brute force search over $a, b \in \mathcal{O}$ beginning at 0 and moving out. When a vector $\begin{bmatrix} a \\ b \end{bmatrix}$ is found satisfying (3), we check that the corresponding form ϕ satisfies

$$\left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} a \\ b \end{bmatrix} \right\} \subseteq M(\phi).$$

This corresponds to a ideal polytope whose vertices contain $\{\infty, 0, 1, \frac{a}{b}\}$.

Once the initial form is found, we implement the algorithm of [11] to find all the perfect forms over F up to the action of $\mathrm{GL}_2(\mathcal{O})$ (and the corresponding structure of the Voronoï polyhedron) in Magma [4]. This descends, via (1), to give a tessellation of $\mathbb{H}_3$ by ideal polytopes.

4 Polytope Data

In this section we collect the results of the computations of the $\mathrm{GL}_2(\mathcal{O})$-conjugacy classes of the ideal Voronoï polytopes.

4.1 Example: $d = -14$

Let $F = \mathbb{Q}(\sqrt{-14})$. Then F has class number four and ring of integers $\mathcal{O} = \mathbb{Z}[\omega]$, where $\omega = \sqrt{-14}$. There are 9 $\mathrm{GL}_2(\mathcal{O})$-classes of polytopes which are of 3 combinatorial types. There are 3 triangular prisms with cuspidal vertices

$$P_1 = \left\{ \infty, 1, \frac{5+2\omega}{9}, \frac{2+\omega}{4}, \frac{4+2\omega}{9}, 0 \right\}$$

$$P_2 = \left\{ \frac{11+4\omega}{23}, 1, \frac{5+2\omega}{9}, \frac{4+2\omega}{9}, \frac{12+4\omega}{23}, 0 \right\}, \quad \text{and}$$

$$P_3 = \left\{ \frac{8+5\omega}{23}, \frac{2+\omega}{5}, \frac{1+\omega}{5}, \frac{2+\omega}{6}, \frac{3+2\omega}{10}, \frac{7+4\omega}{21} \right\},$$

and 5 tetrahedra with cuspidal vertices

$$T_1 = \left\{ \frac{11+4\omega}{23}, \frac{2+\omega}{5}, \frac{4+2\omega}{9}, 0 \right\},$$

$$T_2 = \left\{ 1, \frac{5+2\omega}{9}, \frac{3+\omega}{5}, \frac{12+4\omega}{23} \right\},$$

$$T_3 = \left\{ \frac{11+4\omega}{23}, \frac{2+\omega}{5}, \frac{2+\omega}{6}, 0 \right\},$$

$$T_4 = \left\{ \frac{8+5\omega}{23}, \frac{2+\omega}{5}, \frac{4+2\omega}{9}, 0 \right\}, \quad \text{and}$$

$$T_5 = \left\{ \frac{4+\omega}{6}, 1, \frac{3+\omega}{5}, \frac{12+4\omega}{23} \right\},$$

and a square pyramid with cuspidal vertices

$$S = \left\{ \frac{8+5\omega}{23}, \frac{2+\omega}{5}, \frac{1+\omega}{5}, \frac{2+\omega}{6}, 0 \right\}.$$

Given the cuspidal vertices, one can easily compute the stabilizers of each polytope. The stabilizers are all cyclic in this case. For each stabilizer, we compute a generator. The results are given in Table 1.

Table 1. Stabilizer groups of Voronoï ideal polytopes for $\mathbb{Q}(\sqrt{-14})$

Polytope	Stabilizer	Generator
P_1	C_6	$\begin{bmatrix} 1 & -1 \\ 1 & 0 \end{bmatrix}$
P_2	C_2	$\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$
P_3	C_4	$\begin{bmatrix} \omega+1 & -\omega+6 \\ 2 & -\omega-1 \end{bmatrix}$
T_1	C_2	$\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$
T_2	C_2	$\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$
T_3	C_2	$\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$
T_4	C_2	$\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$
T_5	C_2	$\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$
S	C_2	$\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$

4.2 Polytope Summary

We compute the Voronoï polytopes for all imaginary quadratic number fields $F = \mathbb{Q}(\sqrt{d})$ with class number one and two as well as higher class number for $d > -100$. Although there is no reason an arbitrary convex 3-dimensional polytope could not arise, in all of these cases only 8 combinatorial types show up. We give the names and F-vector ([#vertices, #edges, #faces]) for each in Table 2. We also note that the triangular dipyramid shows up in this range much less frequently than the other polytopes.

In Table 3, we give the number of $\mathrm{GL}_2(\mathcal{O})$-classes of each polytope type for F with class number one or two. In Table 4, we give the number of $\mathrm{GL}_2(\mathcal{O})$-classes of each polytope type for the remaining imaginary quadratic fields with $d > -100$.

Table 2. Combinatorial types of ideal polytopes that occur in this range

polytope	F-vector	picture
tetrahedron	$[4, 6, 4]$	
octahedron	$[6, 12, 8]$	
cuboctahedron	$[12, 24, 14]$	
triangular prism	$[6, 9, 5]$	
hexagonal cap	$[9, 15, 8]$	
square pyramid	$[5, 8, 5]$	
truncated tetrahedron	$[12, 18, 8]$	
triangular dipyramid	$[5, 9, 6]$	

5 Group Presentation

A general result of Macbeath [15] and analogous result of Weil [21] give a general method of computing group presentations for groups of homeomorphisms. For the convenience of the reader, we recall these results here and describe how the polytope data computed above can be used to compute explicit presentations of $\mathrm{GL}_2(\mathcal{O}_F)$.

Consider a connected space X acted upon by a group of homeomorphisms Γ. Let $U \subset X$ be an open set such that $\Gamma \cdot U = X$, and let $\Sigma \subset \Gamma$ denote the set

$$\Sigma = \{g \in \Gamma : g \cdot U \cap U \neq \emptyset\}.$$

Let $F(\Sigma)$ be the free group generated by Σ. For $g \in \Sigma$, let x_g denote the corresponding element of $F(\Sigma)$. Let $W \subset \Sigma \times \Sigma$ denote the set

$$W = \{(g, h) : U \cap g \cdot U \cap gh \cdot U \neq \emptyset\}.$$

Let $R \subset F(\Sigma)$ denote the subgroup generated by $x_g x_h x_{(gh)^{-1}}$ for $(g, h) \in W$. Suppose $\pi_0(X) = \pi_1(X) = \pi_0(U) = 1$. Then the subgroup R is a normal subgroup of $F(\Sigma)$ and $\Gamma \simeq F(\Sigma)/R$.

To apply this result to the polytope data computed above, choose $X = \mathbb{H}_3$. Fix representatives $P_1, \ldots, P_k$ of the $\mathrm{GL}_2(\mathcal{O})$ classes of polytopes such that $D = P_1 \cup \cdots \cup P_k$ is a connected set of polytopes meeting along facets. Let $U \subset \mathbb{H}_3$ be an open neighborhood of $D \cap \mathbb{H}_3$. We note that since the vertices D are at

Table 3. $\mathrm{GL}_2(\mathcal{O})$-classes of Voronoï ideal polytopes for class number one and two

h_F	d								
1	−1	0	1	0	0	0	0	0	0
1	−2	0	0	1	0	0	0	0	0
1	−3	1	0	0	0	0	0	0	0
1	−7	0	0	0	1	0	0	0	0
1	−11	0	0	0	0	0	0	1	0
1	−19	0	0	1	1	0	0	0	0
1	−43	0	0	0	2	1	0	1	0
1	−67	0	1	0	2	1	2	1	0
1	−163	11	0	1	8	2	3	0	0
2	−5	0	0	0	2	0	0	0	0
2	−6	0	0	0	0	1	0	1	0
2	−10	0	1	0	1	0	2	0	0
2	−13	1	0	0	3	1	1	0	0
2	−15	1	1	0	0	0	0	0	0
2	−22	5	0	1	4	0	2	0	0
2	−35	3	4	0	1	0	2	0	0
2	−37	10	0	0	8	1	8	0	0
2	−51	1	0	1	2	1	0	1	0
2	−58	47	0	0	7	2	6	0	0
2	−91	5	1	0	5	0	3	0	0
2	−115	3	1	0	5	2	4	0	0
2	−123	1	1	1	6	3	3	1	0
2	−187	18	1	1	4	1	9	1	0
2	−235	13	1	0	12	4	11	0	0
2	−267	24	1	1	13	5	10	1	0
2	−403	66	1	0	16	2	20	0	2
2	−427	65	2	0	19	4	24	0	0

Table 4. $\mathrm{GL}_2(\mathcal{O})$-classes of Voronoï ideal polytopes with $d > -100$

h_F	d								
3	−23	0	1	0	1	0	1	0	0
3	−31	0	0	0	3	0	1	0	0
3	−59	0	1	1	3	0	2	0	0
3	−83	6	0	0	2	2	1	1	0
4	−14	5	0	0	3	0	1	0	0
4	−17	5	0	0	2	1	3	1	0
4	−21	8	2	0	2	1	4	0	0
4	−30	6	0	0	6	4	4	0	0
4	−33	9	0	1	8	1	6	1	0
4	−34	20	0	0	3	1	6	1	0
4	−39	1	0	0	3	1	1	0	0
4	−46	32	1	0	5	0	9	0	0
4	−55	5	1	0	2	0	2	0	0
4	−57	33	1	0	10	3	14	2	0
4	−73	57	1	1	13	1	14	0	2
4	−78	69	1	0	11	4	18	0	0
4	−82	92	0	0	8	3	11	1	0
4	−85	56	0	0	17	0	28	0	0
4	−93	79	1	0	20	7	21	0	0
4	−97	95	0	1	19	3	19	0	0
5	−47	5	0	0	1	1	2	0	0
5	−79	9	0	0	5	0	4	0	0
6	−26	18	1	0	2	1	4	0	0
6	−29	15	0	0	6	0	6	0	0
6	−38	33	1	0	2	1	6	1	0
6	−53	45	0	0	7	2	13	0	0
6	−61	41	1	0	11	1	16	0	0
6	−87	6	0	0	6	2	3	0	0
7	−71	7	1	0	4	0	4	0	0
8	−41	31	0	1	9	0	8	0	0
8	−62	81	0	0	7	2	7	0	0
8	−65	69	2	0	9	0	19	0	0
8	−66	67	1	1	9	4	12	1	0
8	−69	51	2	0	15	2	21	0	0
8	−77	81	1	0	9	2	26	0	0
8	−94	125	1	0	10	2	17	0	0
8	−95	12	0	0	4	0	9	0	0
10	−74	105	1	0	9	1	12	0	0
10	−86	130	0	0	9	1	18	1	0
12	−89	136	0	0	14	1	21	1	0

infinity, the set U can be chosen so that if $g \in \Sigma$, then g takes an edge of D to another edge of D.

We remark that many redundant generators and relations are created when implementing this result, especially when the stabilizer groups of the polytopes are large. We can compensate for this using Magma's commands for simplifying finitely-presented groups. We illustrate the technique in the example below.

5.1 Example: $d = -14$

Theorem 1. *Let $F = \mathbb{Q}(\sqrt{-14})$ with ring of integers $\mathcal{O} = \mathbb{Z}[\omega]$, where $\omega = \sqrt{-14}$. Then the following is a presentation of $\mathrm{GL}_2(\mathcal{O})$:*

$$\mathrm{GL}_2(\mathcal{O}) = \langle g_1, \cdots, g_8 : R_1 = \cdots = R_{22} = 1 \rangle, \quad \textit{where}$$

$$\begin{aligned}
&R_1 = g_7^2, && R_2 = g_8^2, && R_3 = g_6^2, && R_4 = g_3^2,\\
&R_5 = g_4^2, && R_6 = g_2^2, && R_7 = g_5^4, && R_8 = (g_2 g_1^{-1})^2,\\
&R_9 = (g_4 g_1)^2, && R_{10} = g_5^{-1} g_1^{-3} g_5^{-1}, && R_{11} = (g_7 g_5^{-2})^2, && R_{12} = (g_8 g_5^{-2})^2,\\
&R_{13} = (g_6 g_5^{-2})^2, && R_{14} = (g_4 g_5^{-2})^2, && R_{15} = (g_3 g_5^{-2})^2, && R_{16} = (g_6 g_1^{-1} g_5^{-1})^2,
\end{aligned}$$

$$R_{17} = (g_3 g_5^{-1} g_3 g_1 g_2)^2, \quad R_{18} = (g_3 g_7 g_1 g_8 g_1^{-1})^2, \quad R_{19} = g_4 g_5 g_4 g_1^{-1} g_5 g_1,$$

$$\begin{aligned}
R_{20} &= g_8 g_5^{-1} g_7 g_5^{-1} g_3 g_1^{-1} g_3 g_7 g_3 g_7 g_1 g_8 g_3 g_5 g_7 g_5^{-1},\\
R_{21} &= g_1 g_5 g_7 g_5^{-1} g_3 g_1^{-1} g_3 g_7 g_1 g_5^{-1} g_7 g_5^{-1} g_3 g_1^{-1} g_3 g_7,\\
R_{22} &= g_6 g_5 g_7 g_5^{-1} g_3 g_1^{-1} g_3 g_7 g_1 g_6 g_1^{-1} g_7 g_3 g_1 g_3 g_5 g_7 g_5.
\end{aligned}$$

Proof. We choose X, U, and D as described above. In fact, one can choose D to be the polytopes given in Section 4.1. Then $F(\Sigma)/R$ is defined by 235 generators and 3416 relations. We can simplify this presentation in Magma to get the presentation of $\mathrm{GL}_2(\mathbb{Z}[\sqrt{-14}])$ above, with

$$\begin{aligned}
g_1 &= \begin{bmatrix} 1 & -1 \\ 1 & 0 \end{bmatrix}, & g_2 &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},\\
g_3 &= \begin{bmatrix} \omega+3 & -\omega+1 \\ 6 & -\omega-3 \end{bmatrix}, & g_4 &= \begin{bmatrix} 4\omega & -2\omega+13 \\ 2\omega+13 & -4\omega \end{bmatrix},\\
g_5 &= \begin{bmatrix} -2\omega-5 & 2\omega-3 \\ -10 & 2\omega+5 \end{bmatrix}, & g_6 &= \begin{bmatrix} -5\omega & 3\omega-15 \\ -3\omega-15 & 5\omega \end{bmatrix},\\
g_7 &= \begin{bmatrix} \omega+9 & -2\omega-1 \\ -2\omega+10 & -\omega-9 \end{bmatrix}, & g_8 &= \begin{bmatrix} -2\omega-13 & 4\omega+4 \\ \omega-14 & 2\omega+13 \end{bmatrix}.
\end{aligned}$$

The presentation given in the theorem has torsion elements as generators. In particular, $\mathrm{GL}_2(\mathcal{O})$ is generated by elements of order 2, 4, and 6. Since any torsion-free quotient must map these generators to the identity, one immediately gets the following corollary.

Corollary 1. $\mathrm{GL}_2(\mathbb{Z}[\sqrt{14}])$ *has no torsion-free quotients.*

One finds similar results for $F = \mathbb{Q}(\sqrt{d})$ for $d = -1$ and $d = -3$ in [8].

Acknowledgments. I thank the reviewers for their comments. I would like to thank John Cremona for helpful conversations at the beginning of this project, and Paul Gunnells for introducing me to these techniques. I thank Sebastian Pauli for his advice on the computation, Carlos Nicholas for his help with the polytopes, and Greg Bell for his help with the group presentations. Finally, I thank Steve Donnelly for helpful discussions and the Magma Group at the University of Sydney for their hospitality during a visit, in which part of this research was completed. This work was partially supported by the UNCG New Faculty grant.

References

1. Ash, A.: Deformation retracts with lowest possible dimension of arithmetic quotients of self-adjoint homogeneous cones. Math. Ann. 225(1), 69–76 (1977)
2. Berkove, E.: The mod-2 cohomology of the Bianchi groups. Trans. Amer. Math. Soc. 352(10), 4585–4602 (2000)
3. Berkove, E.: The integral cohomology of the Bianchi groups. Trans. Amer. Math. Soc. 358(3), 1033–1049 (2006) (electronic)
4. Bosma, W., Cannon, J., Playoust, C.: The Magma algebra system. I. The user language. J. Symbolic Comput. 24(3-4), 235–265 (1997); Computational algebra and number theory, London (1993)
5. Bygott, J.: Modular forms and modular symbols over imaginary quadratic fields, Ph.D. thesis, Exeter University (1998)
6. Cremona, J.E., Whitley, E.: Periods of cusp forms and elliptic curves over imaginary quadratic fields. Math. Comp. 62(205), 407–429 (1994)
7. Cremona, J.E.: Periods of cusp forms and elliptic curves over imaginary quadratic fields. In: Elliptic curves and related topics, CRM Proc. Lecture Notes, vol. 4, pp. 29–44. Amer. Math. Soc., Providence (1994)
8. Fine, B.: The HNN and generalized free product structure of certain linear groups. Bull. Amer. Math. Soc. 81, 413–416 (1975)
9. Finis, T., Grunewald, F., Tirao, P.: The cohomology of lattices in SL(2, $\mathbf{C}$). Experiment. Math. 19(1), 29–63 (2010)
10. Goncharov, A.B.: Euler complexes and geometry of modular varieties. Geom. Funct. Anal. 17(6), 1872–1914 (2008)
11. Gunnells, P.E.: Modular symbols for Q-rank one groups and Voronoĭ reduction. J. Number Theory 75(2), 198–219 (1999)
12. Gunnells, P.E., Yasaki, D.: Hecke operators and Hilbert modular forms. In: van der Poorten, A.J., Stein, A. (eds.) ANTS-VIII 2008. LNCS, vol. 5011, pp. 387–401. Springer, Heidelberg (2008)
13. Koecher, M.: Beiträge zu einer Reduktionstheorie in Positivitätsbereichen. I. Math. Ann. 141, 384–432 (1960)
14. Lingham, M.: Modular forms and elliptic curves over imaginary quadratic fields, Ph.D. thesis, University of Nottingham (2005)
15. Macbeath, A.M.: Groups of homeomorphisms of a simply connected space. Ann. of Math. 79(2), 473–488 (1964)
16. Ong, H.E.: Perfect quadratic forms over real-quadratic number fields. Geom. Dedicata 20(1), 51–77 (1986)
17. Rahm, A., Fuchs, M.: The integral homology of PSL_2 of imaginary quadratic integers with non-trivial class group, arXiv:0903.4517 (2009)

18. Riley, R.: Applications of a computer implementation of Poincaré's theorem on fundamental polyhedra. Math. Comp. 40(162), 607–632 (1983)
19. Şengün, M.H., Turkelli, S.: Weight reduction for mod$l$ Bianchi modular forms. J. Number Theory 129(8), 2010–2019 (2009)
20. Swan, R.G.: Generators and relations for certain special linear groups. Advances in Math. 6, 1–77 (1971)
21. Weil, A.: On discrete subgroups of Lie groups. Ann. of Math. 72(2), 369–384 (1960)
22. Whitley, E.: Modular symbols and elliptic curves over imaginary quadratic number fields, Ph.D. thesis, Exeter University (1990)
23. Yasaki, D.: Binary Hermitian forms over a cyclotomic field. J. Algebra 322, 4132–4142 (2009)

Author Index